UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE APPLIQUÉES

PAR
ALAIN GIRARD

EXPLORATION D'UN ALGORITHME GÉNÉTIQUE ET D'UN ARBRE DE DÉCISION À DES FINS DE CATÉGORISATION

AVRIL 2007

CE MÉMOIRE A ÉTÉ ÉVALUER
PAR UN JURY COMPOSÉ DE :

M. Ismail Biskri, Directeur du mémoire
Département de mathématiques et d'informatique

M. François Meunier, Évaluateur
Département de mathématiques et d'informatique

M. Mhamed Mesfioui, Évaluateur
Département de mathématiques et d'informatique

SOMMAIRE

Dans ce mémoire, nous vous présenterons deux algorithmes de classification de texte : les arbres de décision et les algorithmes génétiques. Ces deux algorithmes de classification produisent des règles de décision. Les règles de décision sont des outils pour analyser le comportement des données. Les règles de décision sont souvent utilisées pour regrouper les facteurs les plus importants dans une prise de décision.

Dans une classification de texte, le regroupement préalable des mots permet d'accélérer la rapidité des algorithmes de classification. Les regroupements de mots associent automatiquement les mots similaires dans le but de diminuer le nombre d'attributs à tester et favorisent aussi leurs appartenances à leurs classes respectives. Ils augmentent aussi leurs pertinences par rapport aux autres mots.

Les arbres de décision sont des outils qui démontrent le cheminement de la classification. Plus un attribut est important, plus il sera proche de la racine de l'arbre. Les arbres de décision ont comme objectif de produire des règles simples. Les arbres de décision représentent les attributs les plus représentatifs des classes dans le jeu d'apprentissage.

Les algorithmes génétiques sont des outils qui permettent quant à eux d'explorer un espace de recherche vaste, ils forment de nouvelles hypothèses en combinant les hypothèses actuelles. À chacune des générations d'hypothèses, une fonction d'évaluation favorise les hypothèses les plus intéressantes et élimine les hypothèses les moins performantes.

Les deux algorithmes de classification peuvent être considérés comme complémentaires, les arbres de décision sont une bonne référence pour créer une population de règles et les algorithmes génétiques permettent d'explorer en profondeur l'ensemble de l'espace de recherche plus facilement.

ABSTRACT

In this report, we will present you two algorithms of text classification, that are the decision trees and the genetic algorithms. These two classification algorithms of produce decision rules. The decision rules are tools to analyze the behavior of the data. The decision rules are often used to group the most important factors together in a decision-taking process.

In a text classification, the grouping of words allows the increase of the classification algorithms speed. The groups of words automatically associate the similar words with the goal of decreasing the number of attributes to be tested and favor their memberships in their respective classes. They also increase their aptnesses with regard to the other words.

The decision trees are tools which demonstrate the progress of the classification. The more an attribute is important, the more it will be close to the root of the decision tree. The decision trees have for objective to produce simple rules. The decision trees represent the most representative attributes of the classes in the training set.

The genetic algorithms are tools which allow investigating in a vast research space; they form new hypotheses by combining the current hypotheses. In each of the generation, the function of evaluation favors the most interesting hypotheses and eliminates the least successful hypotheses.

Both algorithms of classification can be considered as complementary, the decision trees are a good reference to create a population of rules and the genetic algorithms allow investigating the whole research in depth space more easily

REMERCIEMENT

En premier lieu, je veux remercier **mes parents et mes sœurs** pour leur encouragement et leur soutien pendant mes années d'étude.

À mon directeur de maîtrise, **Ismail Biskri**, pour m'avoir dirigé dans mon projet de maîtrise et de m'avoir accompagné durant le processus. Son appui, son implication et ses encouragements m'ont aidé à réaliser ma maîtrise.

Je remercie les professeurs du département de mathématiques et d'informatique de l'UQTR pour leurs enseignements de qualité.

J'aimerais remercier aussi mes amis **Yannick et Tony** qui m'ont aidé dans certains aspects en informatique.

Finalement, je voudrais dédier ce mémoire à mes neveux **Loïc** et **Noah Beauvais** et ma nièce **Gaelle Beauvais**.

TABLE DES MATIÈRES

Page

SOMMAIRE i
ABSTRACT ii
REMERCIEMENT iii
TABLE DES MATIÈRES iv
LISTE DES TABLEAUX vii
LISTE DES FIGURES ix
LISTE DES ABRÉVIATIONS xii

CHAPITRE 1 INTRODUCTION 13
1.1 Contexte 13
1.2 Problématique 14
1.3 Résolution du problème de classification textuelle 14
1.4 Organisation du document 15

CHAPITRE 2 ARBRE DE DÉCISION 17
2.1 Introduction sur les arbres de décision 18
2.1.1 Le but des algorithmes de construction d'arbre de décision 20
2.1.2 Construction d'un arbre de décision 20
2.2 Mesure de segmentation 21
2.2.1 Gain informationnel 21
2.2.2 Ratio de gain 22
2.2.3 Critère Gini 27
2.2.4 Khi-deux 36
2.3 Attribut discret et Attribut continu 41
2.4 Les erreurs de classification 43
2.4.1 L'erreur apparente 43
2.4.2 L'estimation de l'erreur réelle 44
2.5 Élagage 44
2.5.1 Pré élagage 45
2.5.2 Post élagage 45
2.5.3 Élagage sur l'erreur réduit 46
2.5.4 Coût de complexité 48
2.6 Importance du jeu d'apprentissage et de test 51
2.6.1 Préparation du jeu d'apprentissage et du jeu de Test 52
2.6.2 Cross-validation 52
2.7 Acquisition de règles de classification 52
2.8 Aspect de recherche 53
2.8.1 Données manquantes ou incohérentes 53
2.8.2 Arbre de décision évolutif 54
2.8.3 Arbre de décision 'Fuzzy' 54

CHAPITRE 3 ALGORITHME GÉNÉTIQUE 57
3.1 Introduction sur les algorithmes génétiques 58
3.1.1 Fonctionnement d'un AG 60
3.1.2 Algorithme d'un AG 61
3.2 Opérateurs génétiques 61
3.2.1 Codage de la structure génétique 62
3.2.2 Initialisation de la population 64
3.2.3 Sélection 64
3.2.4 Évaluation 67
3.2.5 Croisement (Crossover) 67
3.2.6 Mutation 70
3.2.7 Remplacement de la population 71
3.3 Les paramètres d'un AG 72
3.4 Simulation d'un SGA *(Simple Genetic Algorithm)* 72
3.5 Les limitations des algorithmes génétiques 74
3.6 Les algorithmes génétiques au service de la classification 74
3.6.1 Théorème des Schémas 75
3.6.2 Représentation de la codification de règle 75

CHAPITRE 4 IMPLÉMENTATION DES ARBRES DE DÉCISION ET DES ALGORITHMES GÉNÉTIQUES 78
4.1 Implémentation d'un arbre de décision 79
4.2 Structure interne d'un nœud d'un arbre de décision 79
4.2.1 Algorithme générale de construction d'un arbre de décision 80
4.3 Implémentation de l'élagage C4.5 82
4.4 Implémentation de l'élagage CART 83
4.5 Implémentation des règles de décision 83
4.5.1 Fonction de génération de règles de décision 85
4.5.2 L'extrapolation des règles de décision 86
4.5.3 Fonction d'évaluation de règles de décision 86
4.6 Implémentation d'un système de classification 86
4.6.1 Fonctionnement du système de classification 87
4.6.2 Paramètre d'une simulation 88
4.6.3 Structure d'un classificateur 91
4.6.4 Fonction d'encodage et de décodage des règles 92
4.6.5 Serveur de temps 93
4.6.6 Échantillonnage 93
4.6.7 Évaluateur 94
4.6.8 Rôle de l'algorithme génétique 95
4.6.9 Implémentation des opérateurs génétiques 95
4.6.10 Sélection et couplage des individus 95
4.6.11 Hybridation 96
4.6.12 Remplacement de la population 97
4.6.13 Décodage des classificateurs 98

CHAPITRE 5 PRÉPARATION DES DONNÉES 100
5.1 Introduction sur la préparation des données 101
5.2 CRISP-DM 101
5.2.1 Préparation des données 101
5.2.2 Préparation du jeu d'apprentissage 102
5.2.3 Préparation du jeu de tests 103
5.3 Qu'est-ce que la classification? 104
5.3.1 Classification supervisée 104
5.3.2 Classification non supervisée 104
5.4 Classification à partir d'un lexique 104
5.4.1 Établir les attributs 105
5.4.2 Compression du lexique avec les N-Grammes de caractères 107

CHAPITRE 6 EXPÉRIMENTATION ET RÉSULTATS 111
6.1 Fabrication du jeu de donnée 112
6.1.1 Information sur les classes 112
6.2 Observation sur les arbres de décision 115
6.2.1 Comprendre la classification avec les règles extraites d'un arbre de décision 117
6.2.3 Observation sur les règles de décision générées par CART 121
6.3 Observation avec un algorithme génétique 123
6.3.1 Paramètre de la simulation d'un algorithme génétique 124
6.3.2 Interprétation des résultats d'un algorithme génétique 139
6.4 Comparaison entre les arbres de décision et les algorithmes génétiques 140

CHAPITRE 7 CONCLUSION 143

Annexe A 145
Annexe B 162
Annexe C 173
Annexe D 186

Bibliographies 189

LISTE DES TABLEAUX

Page

Table 2-1: Exemple d'un jeu d'apprentissage. 23
Table 2-2: Calcul du gain informationnel pour le jeu d'apprentissage complet..... 24
Table 2-3: Jeu d'apprentissage avec la Condition = 'Nuage'..... 24
Table 2-4: Jeu d'apprentissage avec la Condition = 'Soleil'..... 25
Table 2-5: Calcul du gain informationnel pour le jeu d'apprentissage avec la Condition = 'Soleil'..... 25
Table 2-6: Jeu d'apprentissage avec la Condition = 'Nuage'..... 26
Table 2-7: Calcul du gain informationnel pour le jeu d'apprentissage avec la Condition = 'Pluie'..... 27
Table 2-8: Choix de répartition pour le nœud Condition. 28
Table 2-9: Choix de répartition pour le nœud Température. 29
Table 2-10 : Calcul de l'index Gini pour les autres valeurs pour déterminer la racine. 29
Table 2-11 : Jeu d'apprentissage avec la Condition ='Soleil' ou 'Pluie'..... 30
Table 2-12 : Choix de répartition pour le nœud Température. 30
Table 2-13 : Calcul de l'index Gini pour la branche Condition ='Soleil' ou 'Pluie'. 31
Table 2-14 : Jeu d'apprentissage avec Humidité = 'Élevée'. 31
Table 2-15 : Calcul de l'index Gini pour la branche Humidité = 'Élevée'. 32
Table 2-16 : Jeu d'apprentissage pour la Température = 'Chaud'..... 32
Table 2-17: Jeu d'apprentissage pour la Température = 'Doux'. 33
Table 2-18 : Calcul de l'index Gini pour la branche Température = 'Doux'. 33
Table 2-19 : Jeu d'apprentissage pour la Humidité = 'Normale'. 34
Table 2-20 : Calcul de l'index Gini pour la branche Humidité = 'Normale'. 34
Table 2-21 : Jeu d'apprentissage pour la branche Vent = 'Faible'. 35
Table 2-22 : Jeu d'apprentissage pour la branche Vent = 'Fort'. 35
Table 2-23: Calcul de l'index Gini pour la branche Vent = 'Fort'. 36
Table 2-24: Calcul du khi-deux pour chacun des attributs pour déterminer la racine38
Table 2-25 : Résultat du calcul du khi-deux pour déterminer la racine. 38
Table 2-26: Calcul du khi deux pour chacun des attributs pour la branche Soleil. ... 39
Table 2-27: Résultat du calcul du khi deux pour chacun des attributs pour le sous arbre de la branche Soleil. 39
Table 2-28: Calcul du khi deux pour chacun des attributs pour le sous arbre de la branche Pluie..... 40
Table 2-29: Résultat du calcul du khi deux pour chacun des attributs pour le sous arbre de la branche Pluie. 40
Table 2-30: Tableau des ensembles des regroupements possibles pour un attribut... 42
Table 2-31 : Attribut continu. 43
Table 2-32 : Jeu d'apprentissages parc automobile. 46
Table 3-1: Concept de base des AG. 58
Table 3-2: Comparaison entre le code entier, binaire et Gray. 63
Table 3-3: La population de l'exemple. 65
Table 3-4 : population initiale du SGA. 73
Table 3-5: Itération d'un SGA. 73

Table 3-6: Gène pour représenter un attribut. 76
Table 5-1 : Définition du jeu de données. 106
Table 6-1 : Jeux de données 112
Table 6-2: Nombre de mots présents dans les lexiques des classes. 114
Table 6-3 : Résultats des jeux selon l'algorithme C4.5 115
Table 6-4: Résultats des jeux selon l'algorithme CART 116
Table 6-5 : Résultats des arbres de décision C4.5 selon les jeux 118
Table 6-6: Résultats des arbres de décision CART selon les jeux 122
Table 6-7: Tableau des environnements 124
Table 7-1: Tableau des fréquences des règles acceptables des 100 premières itérations d'un AG 189
Table 7-2: Tableau des fréquences des règles acceptables des 100 dernières itérations d'un AG 190

LISTE DES FIGURES

Page

Figure 2.1: Schéma d'un arbre. 18
Figure 2.2: Schéma d'un arbre de décision. 19
Figure 2.3: Premier niveau de l'arbre C4.5. 24
Figure 2.4: Ajout du nœud terminal pour la valeur Nuage 25
Figure 2.5: Ajout du sous-arbre Humidité 26
Figure 2.6 : Arbre complet C4.5. 27
Figure 2.7 : Premier niveau de l'arbre CART. 30
Figure 2.8 : Ajout du nœud Humidité. 31
Figure 2.9 : Ajout du nœud Température lorsque l'humidité est élevée. 32
Figure 2.10 : Ajout du nœud Vent lorsque l'humidité est élevée. 33
Figure 2.11 : Ajout du nœud Vent lorsque l'humidité est normale. 35
Figure 2.12 : Arbre de décision complet CART. 36
Figure 2.13: Tableau du calcul 37
Figure 2.14 : Arbre de décision complet CHAID. 41
Figure 2.15: Arbre de décision avant élagage 47
Figure 2.16: Arbre de décision après élagage. 48
Figure 2.17: Arbre T_0. 49
Figure 2.18: Arbre T_1. 50
Figure 2.19. : Arbre T_2. 51
Figure 2.20 : Arbre T_3. 51
Figure 2.21 : Schéma d'un arbre flou. 55
Figure 3.1 : Fonctionnement d'un algorithme génétique. 60
Figure 3.2: Représentation de la roulette 66
Figure 3.3: Exemple de croisement. 68
Figure 3.4: Croisement à un point. 68
Figure 3.5: Croisement à deux-points. 69
Figure 3.6: Croisement à k-points où k = 4. 69
Figure 3.7 : Croisement uniforme. 70
Figure 3.8: Représentation d'une mutation aléatoire. 71
Figure 3.9: Représentation d'une mutation uniforme. 71
Figure 3.10: Représentation d'une roulette de sélection 73
Figure 4.1: Fenêtre du jeu d'apprentissages et du jeu de tests 81
Figure 4.2: Fenêtre de génération d'un arbre de décision. 82
Figure 4.3: Fenêtre de représentation des règles de décision. 85
Figure 4.4: Schéma du système de classification. 87
Figure 4.5: Jeu de références de l'évaluation 88
Figure 4.6: Paramètres nécessaires d'une simulation 89
Figure 4.7: Graphique d'une simulation 91
Figure 4.8: Fichiers de sortie. 98
Figure 4.9: Décodage des classificateurs 99
Figure 5.1: Initialisation de la liste de mots 106
Figure 5.2: Fabrication de la liste de similitude des mots. 107

Figure 6.1:Partie d'un arbre de décision C4.5. 119
Figure 6.2:Partie d'un arbre de décision CART. 122
Figure 6.3: Phénotype utilisé pour chacun des environnements. 124
Figure 6.4: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement A 126
Figure 6.5: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement B 126
Figure 6.6: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement C 127
Figure 6.7: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement D 128
Figure 6.8: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement E. 128
Figure 6.9: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement A 130
Figure 6.10: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement B 130
Figure 6.11: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement C 131
Figure 6.12: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement D 132
Figure 6.13: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement E. 132
Figure 6.14 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement A 133
Figure 6.15 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement B 134
Figure 6.16 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement C 135
Figure 6.17 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement D 135
Figure 6.18 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement E. 136
Figure 6.19 : Simulation de l'AG sur les fichiers combinés pour l'environnement A. 137
Figure 6.20: Simulation de l'AG sur les fichiers combinés pour l'environnement B. 138
Figure 6.21: Simulation de l'AG sur les fichiers combinés pour l'environnement C. 138
Figure 6.22: Simulation de l'AG sur les fichiers combinés pour l'environnement D. 139
Figure 6.23: Simulation de l'AG sur les fichiers combinés pour l'environnement E. 140
Figure 7.1: Arbre de décision C4.5 avec le jeu A 165
Figure 7.2: Arbre de décision CART avec le Jeu A 166
Figure 7.3: Arbre de décision C4.5 avec le jeu B 167
Figure 7.4: Arbre de décision CART avec le Jeu B 168

Figure 7.5: Arbre de décision C4.5 avec le Jeu C 169
Figure 7.6: Arbre de décision CART avec le Jeu C 170
Figure 7.7: Arbre de décision C4.5 avec le Jeu D 171
Figure 7.8: Arbre de décision CART avec le Jeu D 172
Figure 7.9: Arbre de décision C4.5 avec le Jeu E 173
Figure 7.10: Arbre de décision CART avec le Jeu E 174

LISTE DES ABRÉVIATIONS

SIGLE	SIGNIFICATION
AD	Arbre de décision
AG	Algorithme génétique
CART	Classification And Regresion Tree
SGA	Simple Genetic Algorithm
CRISP-DM	Cross Industries Standard Process for Data Mining

CHAPITRE 1 INTRODUCTION

Depuis le commencement de l'humanité, l'humain apprend sur le monde qui l'entoure, son intelligence lui a permis d'acquérir des connaissances. La mémoire de l'humain est très restreinte, à comparer à celle d'un ordinateur. Pour mieux se servir de ses connaissances, elle doit associer une classe à cette connaissance. L'humain se sert de son raisonnement pour associer une connaissance à une classe. Cette classification est le regroupement d'idées qui permet de distinguer un objet par rapport à un autre.

Le cerveau humain utilise un raisonnement pour mieux comprendre une connaissance. Pour établir un raisonnement, l'humain se sert généralement de ses expériences antérieures, son instinct et son sens moral.

La capacité d'un ordinateur d'apprendre est limitée, il ne sait pas vraiment faire la différence entre le bien et le mal et il n'a aucun instinct. Il faut qu'il se base sur les expériences qu'il effectue lors d'une classification. Pour cela, les chercheurs en intelligence artificielle se servent de techniques diverses pour associer des idées à une classe. Pour mieux classifier une situation particulière, l'intelligence artificielle s'inspire souvent de la nature pour mieux interpréter des connaissances. Plusieurs approches peuvent être utilisées par exemple : les arbres de décision, les algorithmes génétiques, le clustering ou encore les réseaux de neurones.

1.1 Contexte.

Un algorithme de classification doit explorer des données informationnelles structurées, semi-structurées ou non structurées afin de détecter leur similitude et leur affecter la même catégorie. Parmi les applications principales, on trouve l'analyse de forage et l'exploration de données. Ces domaines d'expertise apparus il y a moins d'une décade suscitent de plus en plus d'intérêts de la part des informaticiens, des mathématiciens et même des spécialistes en sciences humaines.

1.2 Problématique.

Depuis peu, la recherche s'intéresse aux algorithmes génétiques et aux arbres de décisions. Les algorithmes génétiques (AG) étant basés sur la théorie de l'évolution partent d'une collection d'hypothèses et génèrent à chaque itération de nouvelles hypothèses, qui contiendront forcément une hypothèse meilleure que la meilleure des hypothèses précédentes. L'utilisation des algorithmes génétiques présuppose en particulier la non-connaissance d'un savoir particulier du domaine traité. Les arbres de décision (AD) sont contrairement aux algorithmes génétiques dépendants de connaissances accumulées. Ils se révèlent être de très bonnes structures d'organisation des anciennes connaissances pour le contrôle des nouvelles connaissances. Ils représentent aussi divers événements à considérer dans l'analyse décisionnelle.

Les algorithmes génétiques et les arbres de décision peuvent apporter des résultats intéressants dans la classification. Ces deux types d'algorithmes de classification produisent des règles qui permettent d'établir les caractéristiques spécifiques des classes. Pour mieux comprendre les similarités et les différences entre les classes, ces règles de classification reflètent l'ensemble de connaissances en regroupant les caractéristiques similaires entre les objets des classes.

L'exploration des règles produites permet d'illustrer le raisonnement dans l'analyse d'une classification de textes. Cette analyse repose donc sur les mots associés aux classes. À partir des lexiques de classes, les algorithmes génétiques et les arbres de décision produisent des règles pour mettre en évidence les similarités ou les différences entre les classes.

1.3 Résolution du problème de classification textuelle.

Ce mémoire a pour objectif la comparaison entre deux formes d'algorithmes de forage de données, soit les arbres de décision et les algorithmes génétiques dans la classification de textes.

Les arbres de décision et les algorithmes génétiques peuvent travailler facilement avec les règles de classification, les algorithmes de construction d'arbres de décision produisent des règles avec le jeu d'apprentissage et les algorithmes génétiques effectuent ses opérateurs génétiques afin d'améliorer l'ensemble des règles.

Les règles reflètent les similitudes entre les différents textes qui appartiennent à une classe. Elle nous montre aussi les différences par rapport aux autres classes. Les règles sont composées essentiellement de conditions, ces conditions représentent les attributs selon leurs valeurs. Certains attributs ont plus d'impact sur la classification.

Dans la classification textuelle, les attributs sont représentés par les mots contenus dans les textes. Les lexiques des classes sont une bonne base de références pour trouver les mots associées aux classes. La liste de mots disponibles augmente donc rapidement. Nous avons remarqué que certains mots dans les lexiques ont des similarités orthographiques. Ces mots sont généralement similaires par rapport à l'idée qu'il véhicule. Pour augmenter la rapidité des algorithmes de classification et augmenter la pertinence des règles, il est intéressant de regrouper ces mots. L'impact de cette réduction du lexique permettra d'augmenter l'importance de ces mots dans la construction de l'arbre et favorisera la rapidité de traitement des opérateurs de l'algorithme génétique.

L'algorithme génétique crée les nouvelles règles à partir de sa population de règles précédente. La population initiale est donc importante, parce qu'elle influencera la performance de l'AG. Généralement, cette population est générée aléatoirement. Une population initiale de règles générées par des arbres serait probablement mieux appropriée comme population initiale. La comparaison entre les arbres de décision et les algorithmes génétiques reposent sur les règles de classification.

1.4 Organisation du document.

Les chapitres **2** et **3** présentent l'état de l'art des algorithmes de classification et les chapitres **4**, **5** et **6** représentent le travail effectué dans le cadre du mémoire.

Le **chapitre 2** représente l'état de l'art sur les arbres de décision. Nous traiterons de la conception d'un arbre de décision à l'extraction de règles de décision à l'aide de deux algorithmes de construction d'arbre, soit ***C4.5*** et ***CART***. Nous abordons des sujets tels que le traitement des valeurs à intervalle continue, le traitement des erreurs, l'élagage et la préparation du jeu d'apprentissage.

Le **chapitre 3** présente l'état sur les algorithmes génétiques. Nous aborderons des concepts fondamentaux sur les algorithmes génétiques. Nous vous expliquerons le fonctionnement d'un algorithme génétique et de ses opérateurs génétiques. Nous aborderons aussi la transformation des règles en classificateurs.

L'implémentation des algorithmes sera abordée au **chapitre 4**. Nous vous expliquerons les structures et les fonctions utilisées dans l'implémentation des arbres de décision et des algorithmes génétiques.

Le **chapitre 5** présente la phase de préparation des données. Dans ce chapitre, nous vous expliquerons la préparation des données utilisées dans ce projet.

Par la suite, le **chapitre 6** présente les résultats de l'analyse des tests effectués sur les jeux de données à partir des arbres de décision et les algorithmes génétiques.

Le **chapitre 7** sera présenté en guise de conclusion.

CHAPITRE 2 ARBRE DE DÉCISION

2.1 Introduction sur les arbres de décision.

Les arbres de décision sont la modélisation d'une classification. Ils apprennent à partir d'observations qu'on appelle des exemples. Un exemple est représenté par une série d'attributs et une classe associée, on doit connaître la classe parce que les arbres de décision travaillent sur la classification en mode supervisée [5]. Les arbres de décision sont un bon moyen d'illustrer le raisonnement pour distinguer les similitudes et les différences entre les attributs des exemples du jeu de données, ils sont souvent utilisés par les statisticiens pour illustrer le résultat d'une analyse.

Un arbre de décision est composé de nœuds en arborescence, le nœud à base de l'arbre est appelé la racine, chacun des nœuds sous la racine est soit une feuille ou un sous-arbre.

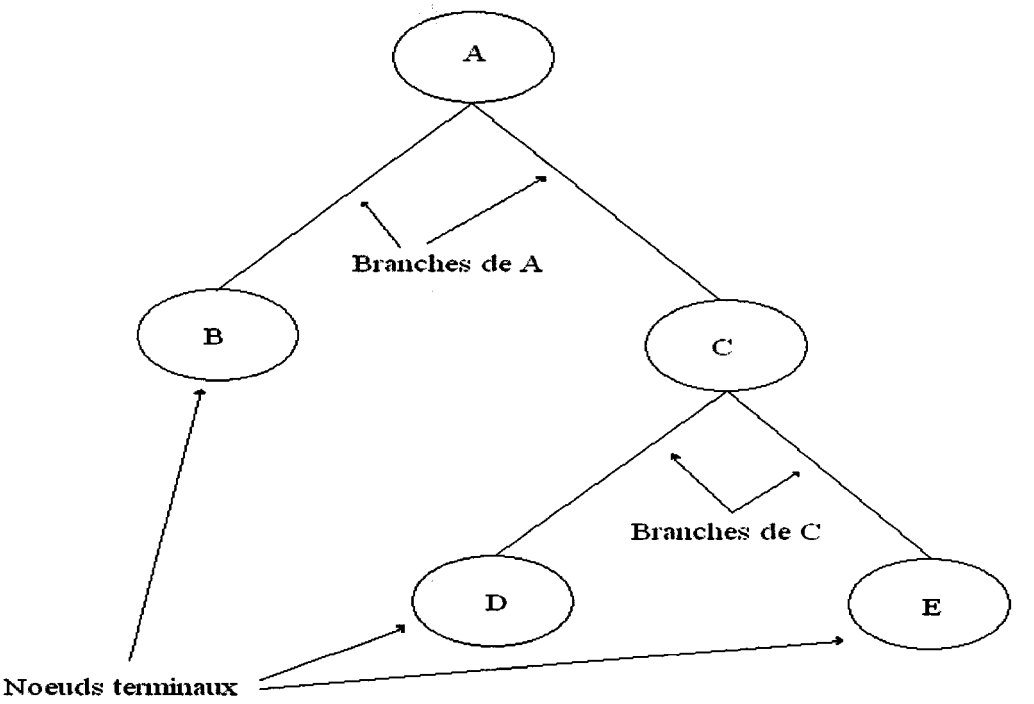

Figure 2.1: Schéma d'un arbre.

Dans la figure 2.1, les nœuds B, D et E sont des nœuds terminaux et le nœud C est un sous arbre du nœud A.

Une feuille est un nœud terminal qui représente le résultat d'une classification. La racine d'un sous arbre est étiquetée avec l'attribut qui a été choisi, les branches sont étiquetées avec les différentes valeurs que peut prendre l'attribut choisi pour le nœud.

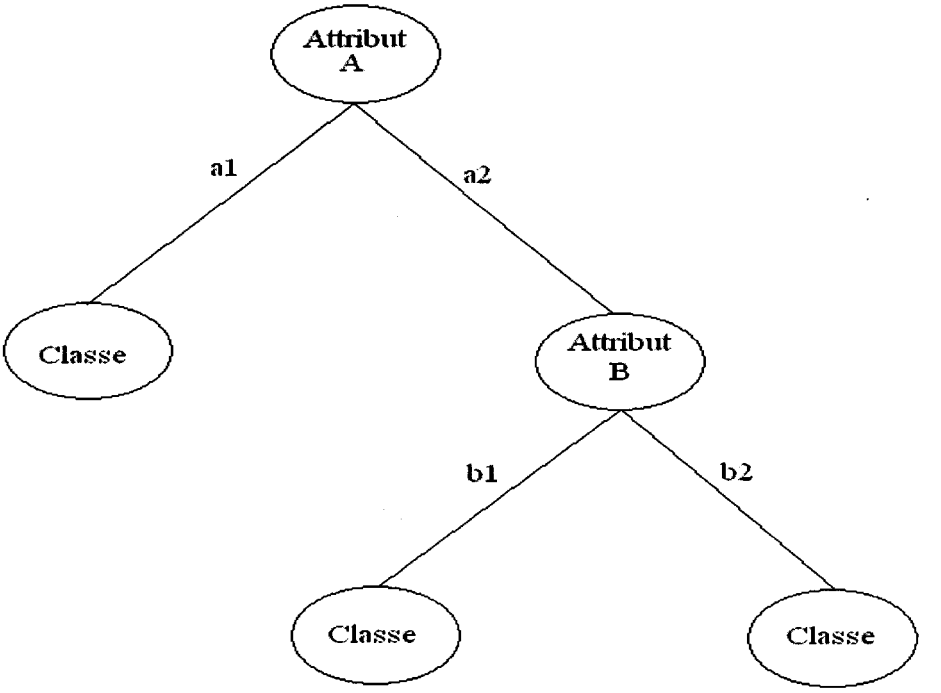

Figure 2.2: Schéma d'un arbre de décision.

Dans l'arbre de décision de la figure 2.2, les attributs A et B ont chacun deux valeurs distinctes, lorsque les exemples avec l'attribut A égal à a_1, ils correspondent à une seule classe. Dans le cas, où l'attribut A est égal à a_2, les exemples correspondent à deux classes différentes, on a besoin alors de prendre l'attribut B pour diviser les exemples dans leurs classes respectives.

Il existe plusieurs algorithmes de construction d'arbre, les plus populaires sont ID3, C4.5, CART et CHAID.

2.1.1 Le but des algorithmes de construction d'arbre de décision.

Les algorithmes de construction d'arbre de décision permettent de créer des arbres de décision avec une taille la plus petite que possible, et ce, de façon à créer des règles de décision simples. Plus un arbre de décision est grand, plus les règles sont complexes. Les algorithmes de construction d'arbres choisissent les attributs toujours par rapport aux classes.

2.1.2 Construction d'un arbre de décision.

Les arbres de décision sont construits à partir d'un jeu d'apprentissage, un jeu d'apprentissage est une matrice, où les lignes représentent les exemples et les colonnes représentent les caractéristiques des exemples, la dernière colonne est réservée aux classes associées aux exemples. L'algorithme de construction a aussi besoin d'un tableau d'index qui constitue la liste de référence des attributs à traiter.

L'algorithme de construction d'arbre de décision se divise en 3 étapes. La première étape consiste à vérifier si on doit faire un nœud terminal pour représenter les exemples du jeu d'apprentissage. Pour faire un nœud terminal, on doit respecter une des conditions suivantes : Tous les exemples du jeu d'apprentissage appartiennent à la même classe ou tous les attributs ont été utilisés pour les nœuds précédents. Cette étape permet d'arrêter l'expansion de la branche de l'arbre.

La deuxième et la troisième étape se produisent lorsqu'on ne respecte pas les critères de la première.

La deuxième consiste à trouver l'attribut pour représenter le nœud de l'arbre. Les algorithmes de construction d'arbre de décision utilisent une mesure de segmentation par rapport aux attributs à traiter. Nous allons voir en détail les différentes techniques plus tard.

La troisième étape consiste à éclater le jeu d'apprentissages pour créer les branches du nœud, chacune des branches du nœud prend une des différentes valeurs que

l'attribut du nœud peut prendre. Pour chacune des branches qu'on aura créées, il faut recommencer le processus en prenant les exemples correspondants à la branche.

2.2 Mesure de segmentation.

La mesure de segmentation est l'heuristique qui permet de choisir l'attribut qui permettra de répartir le mieux le jeu d'apprentissages. Cette mesure est souvent une mesure statistique. L'objectif principal est de construire des arbres de décision relativement simple. On recherche un arbre petit et simple plutôt qu'un arbre grand qui est complexe. Le choix des attributs à tester est une étape cruciale pour la construction d'un arbre. Pour cela, la mesure de segmentation doit évaluer toutes les possibilités de choix pour chacun des niveaux d'un arbre de décision.

2.2.1 Gain informationnel.

Le gain informationnel est une mesure de segmentation qui utilise l'entropie de Shannon. ID3 et C4.5 [1, 3, 5, 10, 13] utilisent le gain pour choisir l'attribut pour représenter le nœud. Il conserve seulement les informations absolument nécessaires pour classer un objet. À chaque fois, qu'on doit choisir un attribut pour partitionner l'ensemble d'exemples, il faut choisir celui dont l'entropie de classification est la plus petite. En général, le gain privilégie généralement les attributs ayant un grand nombre de valeurs [20].

Pour avoir un arbre de décision concis et suffisant, il ne faut pas seulement traiter les attributs séquentiellement. La richesse de cette mesure consiste à choisir judicieusement les attributs nécessaires comme des nœuds intermédiaires, pour arriver au le chemin le plus court qui correspond de plus au plus grand nombre d'exemples dans la même classe.

Le gain informationnel (voir l'équation 2.2) est la différence entre la répartition des classes par rapport au jeu d'apprentissage et la répartition des valeurs des attributs par rapport aux classes.

$$Info(S) = -\sum_{j=1}^{k} \frac{freq(C_j, S)}{|S|} * \log_2 \left(\frac{freq(C_j, S)}{|S|} \right) \quad (2.1)$$

$$Gain(X) = Info(T) - \sum_{i=1}^{nbTest} \frac{|T_i|}{|T|} * Info(T_i) \quad (2.2)$$

La fonction *freq* (C_j,S) trouve la fréquence des exemples qui correspondent à la classe C_j dans le jeu d'apprentissage S, |T| représente le nombre d'exemples à évaluer, *nbTest* est le nombre de valeurs pour l'attribut testé, $|T_i|$ est le nombre d'exemples qui correspond à la valeur i de l'attribut testé.

2.2.2 Ratio de gain.

C4.5 utilise une notion complémentaire au gain informationnel qu'on appelle le ratio de gain [3, 11]. Il est utilisé pour pondérer le gain qui favorise les attributs qui ont beaucoup de valeurs [20]. On calcule toujours le gain informationnel, cependant on calcule aussi la répartition des valeurs de l'attribut par rapport au jeu d'apprentissage. Ce facteur permet d'éviter de tomber dans le surapprentissage. Le Split Info représente l'information potentielle générée en partitionnant le jeu d'apprentissage T en n sous-ensembles, elle montre la proportion de l'information générée par l'éclatement par un attribut [3].

$$SplitInfo(X) = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} * \log_2 \left(\frac{|T_i|}{|T|} \right) \quad (2.3)$$

Le ratio de gain sélectionne le test de façon à optimiser le ratio, on prend toujours en compte du gain informationnel, mais on tient compte de la répartition des valeurs des attributs pour choisir l'attribut pour partitionner le jeu d'apprentissage.

$$GainRatio(X) = \frac{Gain(X)}{SplitInfo(X)} \quad (2.4)$$

Petite mise en situation.

Pour illustrer les différents comportements des algorithmes utilisant le gain informationnel et du ratio de gain. Voici un jeu d'apprentissage qui est souvent utilisé dans la littérature sur les arbres de décision pour expliquer le fonctionnement de la construction d'un arbre de décision. Ce jeu d'apprentissages décrit les conditions de météo pour savoir si les conditions sont idéales ou non pour aller jouer au golf.

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X1	*Soleil*	*Chaud*	*Élevée*	*Faible*	*Non*
X2	*Soleil*	*Chaud*	*Élevée*	*Fort*	*Non*
X3	*Nuage*	*Chaud*	*Élevée*	*Faible*	*Oui*
X4	*Pluie*	*Doux*	*Élevée*	*Faible*	*Oui*
X5	*Pluie*	*Froid*	*Normale*	*Faible*	*Oui*
X6	*Pluie*	*Froid*	*Normale*	*Fort*	*Non*
X7	*Nuage*	*Froid*	*Normale*	*Fort*	*Oui*
X8	*Soleil*	*Doux*	*Élevée*	*Faible*	*Non*
X9	*Soleil*	*Froid*	*Normale*	*Faible*	*Oui*
X10	*Pluie*	*Doux*	*Normale*	*Faible*	*Oui*
X11	*Soleil*	*Doux*	*Normale*	*Fort*	*Oui*
X12	*Nuage*	*Doux*	*Élevée*	*Fort*	*Oui*
X13	*Nuage*	*Chaud*	*Normale*	*Faible*	*Oui*
X14	*Pluie*	*Doux*	*Élevée*	*Fort*	*Non*

Table 2-1: Exemple d'un jeu d'apprentissage.

Dans le jeu d'apprentissage de la table 2-1, on a 14 exemples, chacun de ces exemples est composé de 4 attributs (Condition, Température, Humidité, Vent) et d'une classe.

L'algorithme commence par trouver quel attribut serait le meilleur choix pour répartir les exemples avec une classe en particulier. On doit considérer le nombre de valeurs pour chacun des attributs : on a 3 valeurs possibles pour l'attribut Condition, 3 pour l'attribut Température, 2 valeurs pour les attributs Vent et Humidité.
On commence par calculer la réparation du jeu d'apprentissage par rapport aux classes, dans l'exemple, on a 9 exemples pour la classe Oui et 5 exemples pour la classe Non. Ensuite, on regarde la réparation des valeurs pour chacun des attributs

par rapport aux classes. On calcule d'abord la répartition du jeu d'apprentissage par rapport aux classes.

$$-9/14 \times \log_2(9/14) - 5/14 \times \log_2(5/14) = 0{,}940$$

Attribut	*Gain de l'attribut*	*SplitInfo*	*Ratio*
Condition	0,247	1.577	0.157
Température	0.029	1.557	0.0186
Humidité	0.152	1	0.152
Vent	0.048	0.985	0.0487

Table 2-2: Calcul du gain informationnel pour le jeu d'apprentissage complet.

Le meilleur ratio est pour l'attribut **Condition** (voir la table 2-2), donc cet attribut est choisi pour constituer la racine de l'arbre. Pour chacune des valeurs de l'attribut, on aura une branche étiquetée pour cette valeur. Dans ce cas particulier, on aura 3 branches : Soleil, Nuage, Pluie (voir la figure 2.3).

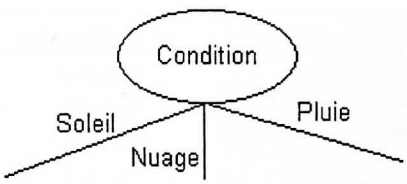

Figure 2.3: Premier niveau de l'arbre C4.5.

Il faut ensuite séparer le jeu d'apprentissage selon les valeurs de l'attribut choisi pour former des jeux d'apprentissage, on enlève aussi l'attribut choisi dans les attributs à traiter. L'algorithme de construction d'arbre est appelé avec chacun des jeux d'apprentissage et avec les attributs restants du niveau précédent.

Lorsque l'attribut Condition = 'Nuage ', le jeu d'apprentissage est représenté à la table 2-3.

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X3	*Nuage*	*Chaud*	*Élevée*	*Faible*	*Oui*
X7	*Nuage*	*Froid*	*Normale*	*Fort*	*Oui*
X12	*Nuage*	*Doux*	*Élevée*	*Fort*	*Oui*
X13	*Nuage*	*Chaud*	*Normale*	*Faible*	*Oui*

Table 2-3: Jeu d'apprentissage avec la Condition = 'Nuage'.

On remarque lorsque la Condition = 'Nuage ' que tous les exemples du jeu d'apprentissage (voir table 2-3) appartiennent à la même classe soit 'Oui'. On peut alors créer un nœud terminal associé à la classe 'Oui' (voir la figure 2.4).

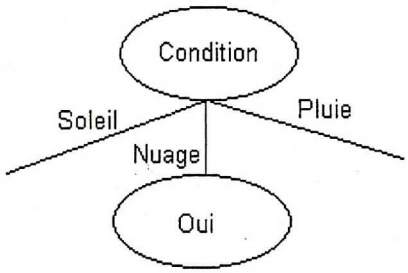

Figure 2.4: Ajout du nœud terminal pour la valeur Nuage.

Lorsque l'attribut Condition = 'Soleil ', le jeu d'apprentissages est représenté à la table 2-4.

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X1	*Soleil*	*Chaud*	*Élevée*	*Faible*	*Non*
X2	*Soleil*	*Chaud*	*Élevée*	*Fort*	*Non*
X8	*Soleil*	*Doux*	*Élevée*	*Faible*	*Non*
X9	*Soleil*	*Froid*	*Normale*	*Faible*	*Oui*
X11	*Soleil*	*Doux*	*Normale*	*Fort*	*Oui*

Table 2-4: Jeu d'apprentissage avec la Condition = 'Soleil'.

Si on examine attentivement la table 2.4, on remarque que les valeurs de l'attribut Humidité concordent avec les valeurs des classes, c'est-à-dire que pour une classe, il y a une seule valeur.

Attribut	*Gain de l'attribut*	*SplitInfo*	*Ratio*
Température	0.571	1.522	0.375
Humidité	0.971	0.971	1
Vent	0.020	0.981	0.002

Table 2-5: Calcul du gain informationnel pour le jeu d'apprentissage avec la Condition = 'Soleil'.

À partir de la table 2-5, on remarque que les attributs Température et Vent ne sont significatifs que par rapport à l'attribut Humidité, si on prenait un de ces attributs à la place de l'attribut Humidité, on risquerait de compromettre l'intégrité de l'arbre.

On remarque pour les deux valeurs des deux attributs que le ratio maximal est égal à 1. C'est-à-dire que les sous arbres auront chacun uniquement des nœuds terminaux. L'arbre final est un arbre parfait, c'est-à-dire que chacun des nœuds terminaux fait référence à plus qu'un exemple du jeu d'apprentissage et qu'il n'y a pas d'exemples mal classés par l'arbre (voir la figure 2.5).

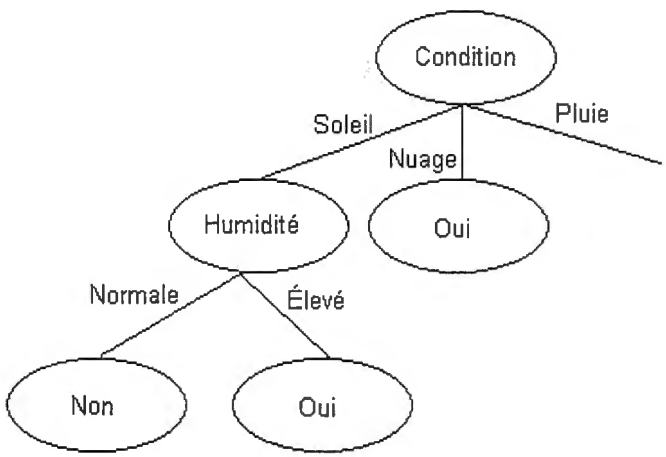

Figure 2.5: Ajout du sous-arbre Humidité

Lorsque l'attribut Condition = 'Pluie ', le jeu d'apprentissage est représenté à la table 2.6.

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X4	*Pluie*	*Doux*	*Élevée*	*Faible*	*Oui*
X5	*Pluie*	*Froid*	*Normale*	*Faible*	*Oui*
X6	*Pluie*	*Froid*	*Normale*	*Fort*	*Non*
X10	*Pluie*	*Doux*	*Normale*	*Faible*	*Oui*
X14	*Pluie*	*Doux*	*Élevée*	*Fort*	*Non*

Table 2-6: Jeu d'apprentissage avec la Condition = 'Nuage'.

Si on examine attentivement la table 2.6, on remarque que les valeurs de l'attribut Vent concordent avec les valeurs des classes, c'est-à-dire que pour une classe, il y a une seule valeur.

Attribut	*Gain de l'attribut*	*SplitInfo*	*Ratio*
Température	-0.947	0.971	-0.975
Humidité	-0.947	0.971	-0.975
Vent	0.971	0.971	1

Table 2-7: Calcul du gain informationnel pour le jeu d'apprentissage avec la Condition = 'Pluie'.

On remarque à la table 2-7 que les gains des attributs Température et Humidité sont négatifs, si on choisissait un de ces attributs, l'arbre serait plus profond sans apporter une distinction claire et précise. Le dernier ajout représente l'arbre complet est illustré à la figure 2.6.

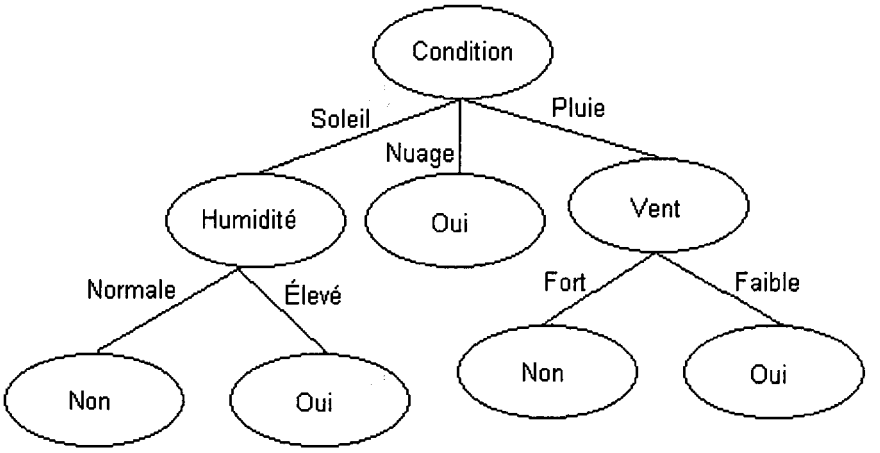

Figure 2.6 : Arbre complet C4.5.

2.2.3 Critère Gini.

Le critère Gini est la mesure de segmentation de l'algorithme CART (Classification and Regression Tree), cet algorithme construit des arbres binaires, c'est-à-dire que les nœuds non terminaux ont seulement 2 branches [4, 11]. Lorsqu'un attribut a plusieurs valeurs possibles, on doit faire des regroupements pour être en mesure de partitionner en deux.

Un bon critère d'éclatement doit prendre soin que l'éclatement soit fait à un nœud qui réduit le coût des erreurs de classification de l'arbre. L'index Gini est utile lorsque le problème comporte plusieurs classes.

$$Gini(S) = 1 - \sum_{j}^{x} freq(C_j, S)^2 \qquad (2.5)$$

$$CritèreGini(p) = Gini(p) - (P_{gauche} \times Gini(p1) + P_{droite} \times Gini(p2)) \qquad (2.6)$$

On évalue le critère Gini de l'attribut à la position p, P_{gauche} et P_{droite} sont les proportions des éléments dans le jeu d'apprentissage associé à p qui vont au nœud qui est associé à $p1$ et $p2$.

Nous allons vous expliquer plus en détail, l'algorithme CART avec le jeu de la table 2-1. Pour commencer, on calcule l'*index Gini* par rapport à l'objectif de classification

$$1 - ((9/14)^2 + (5/14)^2) = 0.459$$

Ensuite, on regarde le nombre de valeurs pour chacun des attributs : 3 valeurs possibles pour l'attribut **Condition**, 2 valeurs pour l'attribut Température, 2 valeurs pour l'attribut Humidité et 2 valeurs pour l'attribut Vent.

Pour déterminer le test pour l'attribut **Condition**, on doit évaluer les possibilités de regroupement de valeurs possible, on a 3 façons de séparer en deux les valeurs de cet attribut (voir la table 2-8).

Ensemble gauche	**Ensemble droit**	**Index Gini pour l'attribut Condition**
Soleil	Nuage ou Pluie	0.065
Nuage	Soleil ou Pluie	0.102**
Pluie	Soleil ou Nuage	0.002

** valeur qui maximise l'index Gini pour cet attribut

Table 2-8: Choix de répartition pour le nœud Condition.

Le choix du regroupement pour cet attribut a été déterminé selon celui qui maximisait l'index Gini. Dans ce cas, le regroupement choisi est **Nuage** pour l'ensemble gauche et **Soleil ou Pluie** pour l'ensemble droit.

Pour déterminer le test pour l'attribut **Température**, on doit évaluer les possibilités de regroupement de valeurs possible, on a 3 façons de séparer en deux les valeurs de cet attribut (voir la table 2-9).

Ensemble gauche	Ensemble droit	Index Gini
Chaud	Doux ou Froid	0.0163**
Doux	Chaud ou Froid	0.0008
Froid	Doux ou Chaud	0.0091

** valeur qui maximise l'index Gini pour cet attribut

Table 2-9: Choix de répartition pour le nœud Température.

Le choix du regroupement pour cet attribut a été déterminé selon celui qui maximisait l'index Gini. Dans ce cas, le regroupement choisi est **Chaud** pour l'ensemble gauche et **Doux ou Froid** pour l'ensemble droit.

Pour les autres attributs à la table 2-10, on a seulement 2 valeurs, donc on a seulement une façon d'éclater l'ensemble en deux pour chacun de ces attributs.

Attribut	Ensemble gauche	Ensemble droit	Calcul de l'index Gini
Humidité	Élevée	Normale	0.091
Vent	Faible	Fort	0.031

Table 2-10 : Calcul de l'index Gini pour les autres valeurs pour déterminer la racine.

On choisi comme racine de l'arbre l'attribut qui maximise l'index Gini et on rappelle récursivement l'algorithme pour la branche gauche avec les exemples qui correspondent avec les valeurs de l'ensemble gauche et pour la branche droite avec les exemples qui correspondent avec les valeurs de l'ensemble droit. L'attribut qui maximise l'index Gini est **Condition**.

Pour la branche gauche de l'attribut **Condition**, le jeu d'apprentissage est le même qu'à la table 2-3. On remarque que l'objectif est le même pour tous les exemples, on se retrouve donc dans le cas où le nœud est terminal (voir la figure 2.7).

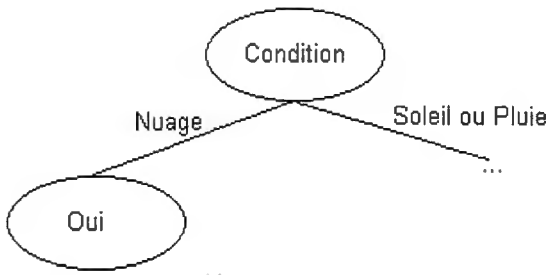

Figure 2.7 : Premier niveau de l'arbre CART.

Pour la branche droite, le jeu d'apprentissage est composé des exemples de la table 2-11.

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X1	*Soleil*	*Chaud*	*Élevé*	*Faible*	*Non*
X2	*Soleil*	*Chaud*	*Élevé*	*Fort*	*Non*
X4	*Pluie*	*Doux*	*Élevé*	*Faible*	*Oui*
X5	*Pluie*	*Froid*	*Normale*	*Faible*	*Oui*
X6	*Pluie*	*Froid*	*Normale*	*Fort*	*Non*
X8	*Soleil*	*Doux*	*Élevé*	*Faible*	*Non*
X9	*Soleil*	*Froid*	*Normale*	*Faible*	*Oui*
X10	*Pluie*	*Doux*	*Normale*	*Faible*	*Oui*
X11	*Soleil*	*Doux*	*Normale*	*Fort*	*Oui*
X14	*Pluie*	*Doux*	*Élevé*	*Fort*	*Non*

Table 2-11 : Jeu d'apprentissage avec la Condition ='Soleil' ou 'Pluie'.

Pour la branche droite de l'attribut **Condition**, on recommence le processus avec les attributs restants, soit **Température**, **Humidité** et **Vent** :

On calcule l'index Gini par rapport aux classes de la table 2-11

$$1 - ((5/10)^2 + (5/10)^2) = 0.5$$

Pour déterminer le test pour l'attribut **Température**, on doit évaluer les possibilités de regroupement de valeurs possible, on a 3 façons de séparer en deux les valeurs de cet attribut (voir la table 2-12).

Ensemble gauche	**Ensemble droit**	**Calcul de l'index Gini pour l'attribut Température**
Chaux	Doux ou Froid	0.125**
Doux	Chaud ou Froid	0.020
Froid	Doux ou Chaux	0.023

** valeur qui maximise l'index Gini pour cet attribut

Table 2-12 : Choix de répartition pour le nœud Température.

Le choix du regroupement pour cet attribut a été déterminé selon celui qui maximisait l'index Gini. Dans ce cas, le regroupement choisi est **Chaud** pour l'ensemble gauche et **Doux ou Froid** pour l'ensemble droit (voir la table 2-13).

Attribut	Ensemble gauche	Ensemble droit	Calcul de l'index Gini
Humidité	Élevée	Normale	0.180
Vent	Faible	Fort	0.083

Table 2-13 : Calcul de l'index Gini pour la branche Condition ='Soleil' ou 'Pluie'.

On choisi l'attribut **Humidité** comme nœud, et on rappelle récursivement l'algorithme avec **Humidité = élevée** (voir la table 2-14) pour la branche gauche du nœud et avec **Humidité = normale** (voir la table 2-19) pour la branche droite. Les valeurs de l'attribut Humidité ne permettent pas de faire un choix clair et précis, mais c'est le choix le plus significatif par rapport aux autres attributs.

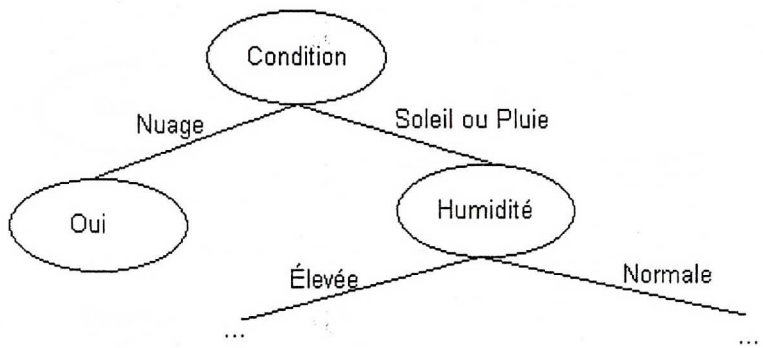

Figure 2.8 : Ajout du nœud Humidité.

Pour la branche gauche **Humidité = élevée**, les attributs restants sont **Température** et **Vent**

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X1	*Soleil*	*Chaud*	*Élevée*	*Faible*	*Non*
X2	*Soleil*	*Chaud*	*Élevée*	*Fort*	*Non*
X4	*Pluie*	*Doux*	*Élevée*	*Faible*	*Oui*
X8	*Soleil*	*Doux*	*Élevée*	*Faible*	*Non*
X14	*Pluie*	*Doux*	*Élevée*	*Fort*	*Non*

Table 2-14 : Jeu d'apprentissage avec Humidité = 'Élevée'.

On calcule l'index Gini par rapport aux classes de la table 2-14

$$1- ((4/5)^2 + (1/5)^2) = 0.32$$

Les deux attributs ont seulement 2 valeurs.

Attribut	Ensemble gauche	Ensemble droit	Calcul de l'index Gini
Température	Doux	Chaud	0.053
Vent	Faible	Fort	0.053

Table 2-15 : Calcul de l'index Gini pour la branche Humidité = 'Élevée'.

On choisi **Température** comme la racine du nœud, et on rappelle récursivement l'algorithme avec **Température = Chaud** pour la branche droite et **Température = Doux** pour la branche droite (voir la figure 2.9).

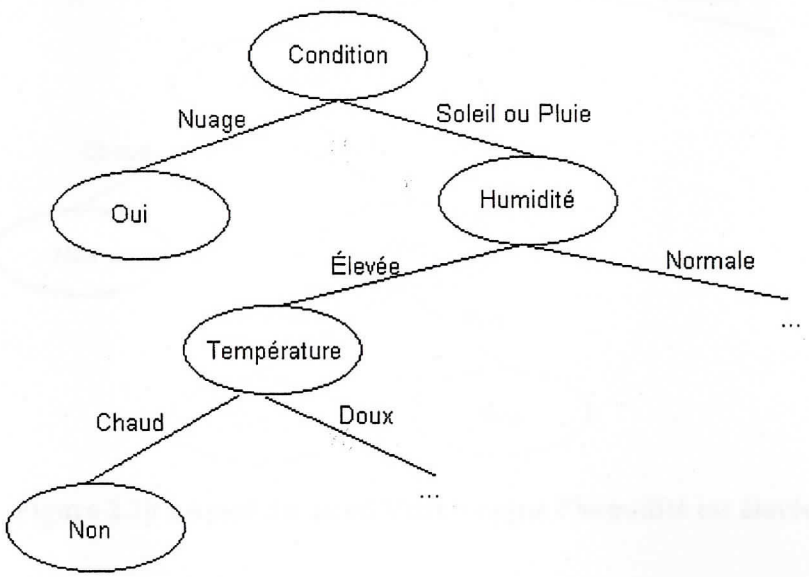

Figure 2.9 : Ajout du nœud Température lorsque l'humidité est élevée.

Pour la branche gauche **Température = Chaud**, tous les exemples (voir table 2-16) correspondent à la même classe, donc le nœud sera terminal avec le nom de la classe.

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X1	*Soleil*	*Chaud*	*Élevée*	*Faible*	*Non*
X2	*Soleil*	*Chaud*	*Élevée*	*Fort*	*Non*

Table 2-16 : Jeu d'apprentissage pour la Température = 'Chaud'.

Pour la branche droite **Température = Doux** (voir la table 2-17), il reste seulement l'attribut **Vent** (voir la figure 2.10).

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X4	*Pluie*	*Doux*	*Élevée*	*Faible*	*Oui*
X8	*Soleil*	*Doux*	*Élevée*	*Faible*	*Non*
X14	*Pluie*	*Doux*	*Élevée*	*Fort*	*Non*

Table 2-17: Jeu d'apprentissage pour la Température = 'Doux'.

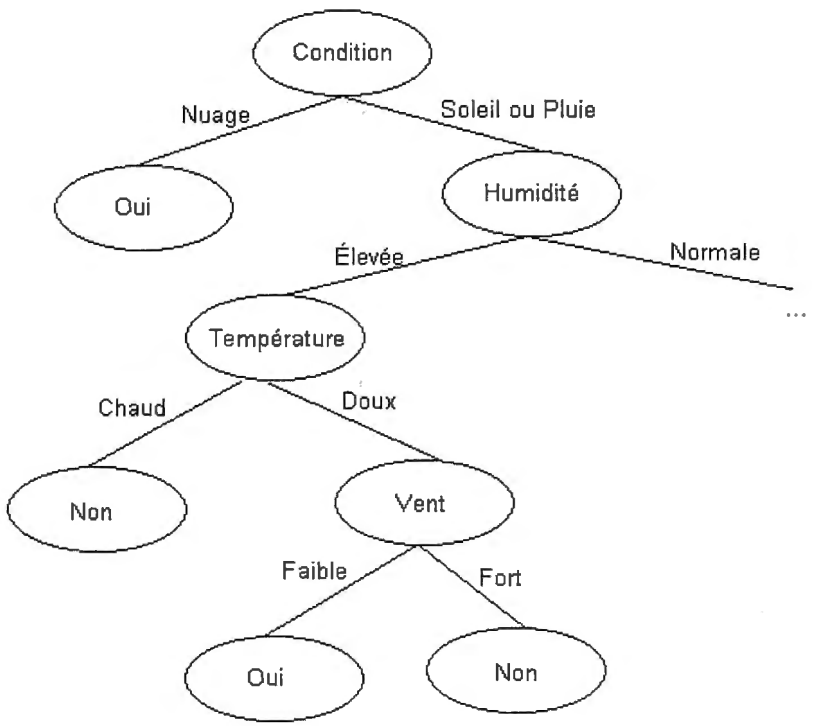

Figure 2.10 : Ajout du nœud Vent lorsque l'humidité est élevée.

On calcule l'index Gini par rapport aux classes de la table 2-17.

$$1-((2/3)^2+(1/3)^2)=0.44$$

L'attribut a seulement 2 valeurs

Attribut	Ensemble gauche	Ensemble droit	Calcul de l'index Gini
Vent	Faible	Fort	0.111

Table 2-18 : Calcul de l'index Gini pour la branche Température = 'Doux'.

Il reste seulement un attribut à traiter, donc on choisi l'attribut **Vent** comme racine du nœud et on rappelle récursivement l'algorithme avec les exemples correspondant à **Vent = Faible** pour la branche gauche et **Vent = Fort** pour la branche droite.

Pour les branches **Vent = Faible** et **Vent = Fort**, il ne reste plus d'attributs à traiter, donc on se retrouve dans un cas de nœud terminal. On regarde si tous les exemples appartiennent à la même classe. On prend la classe majoritaire comme nom pour chacun de nœud. Ensuite, on ajuste l'erreur en conséquence.

Pour la branche droite **Humidité = normale,** les attributs restants sont **Température** et **Vent** (voir la table 2-19).

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X5	*Pluie*	*Froid*	*Normale*	*Faible*	*Oui*
X6	*Pluie*	*Froid*	*Normale*	*Fort*	*Non*
X9	*Soleil*	*Froid*	*Normale*	*Faible*	*Oui*
X10	*Pluie*	*Doux*	*Normale*	*Faible*	*Oui*
X11	*Soleil*	*Doux*	*Normale*	*Fort*	*Oui*

Table 2-19 : Jeu d'apprentissage pour la Humidité = 'Normale'.

On calcule l'index Gini par rapport aux classes au jeu de la table 2-19.

$$1-((4/5)^2 + (1/5)^2) = 0.32$$

Les deux attributs ont seulement 2 valeurs (voir la table 2-20).

Attribut	**Ensemble gauche**	**Ensemble droit**	**Calcul de l'index Gini**
Température	Froid	Doux	0.053
Vent	Faible	Fort	0.119

Table 2-20 : Calcul de l'index Gini pour la branche Humidité = 'Normale'.

On choisi l'attribut **Vent** comme racine du nœud, et on rappelle récursivement l'algorithme avec les exemples correspondant **Vent = Faible** pour la branche gauche et **Vent = Fort** pour la branche droite (voir la figure 2.11).

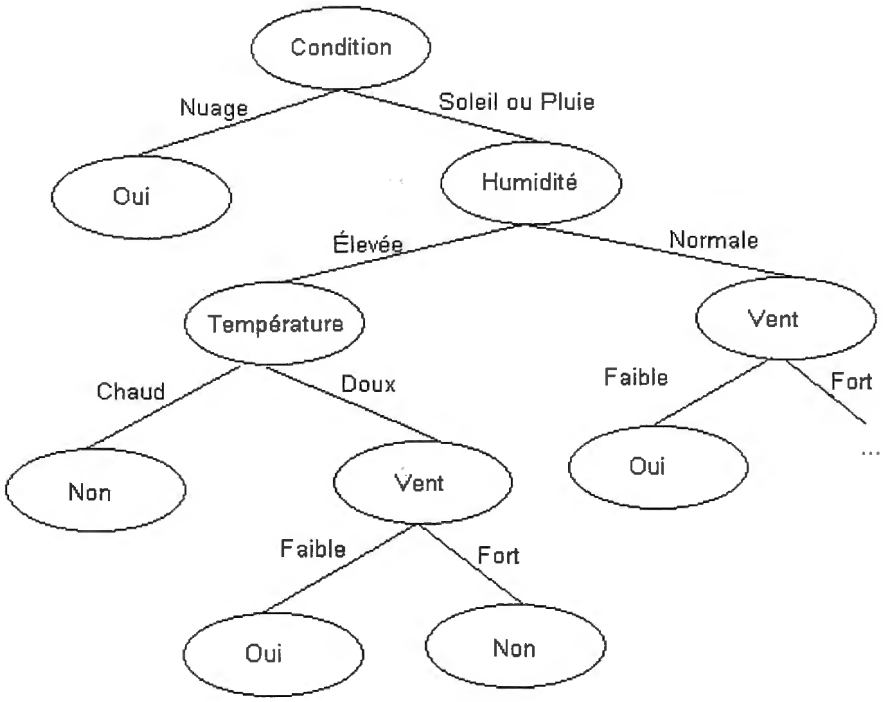

Figure 2.11 : Ajout du nœud Vent lorsque l'humidité est normale.

Pour la branche gauche **Vent = Faible**, tous les exemples correspondent à la même classe, donc le nœud sera terminal avec le nom de la classe (voir la table 2-21).

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X5	*Pluie*	*Froid*	*Normale*	*Faible*	*Oui*
X9	*Soleil*	*Froid*	*Normale*	*Faible*	*Oui*
X10	*Pluie*	*Doux*	*Normale*	*Faible*	*Oui*

Table 2-21 : Jeu d'apprentissage pour la branche Vent = 'Faible'.

Pour la branche droite **Vent = Fort**, il reste seulement l'attribut **Température** qui n'a pas été utilisé dans les niveaux supérieurs de l'arbre.

On calcule l'index Gini par rapport aux classes de la table 2-21

$$1-((1/2)^2 + (1/2)^2) = 0.5$$

Exemple	*Condition*	*Température*	*Humidité*	*Vent*	*Classe*
X6	*Pluie*	*Froid*	*Normale*	*Fort*	*Non*
X11	*Soleil*	*Doux*	*Normale*	*Fort*	*Oui*

Table 2-22 : Jeu d'apprentissage pour la branche Vent = 'Fort'.

L'attribut Température a seulement 2 valeurs selon la table 2-22.

Attribut	Ensemble gauche	Ensemble droit	Calcul de l'index Gini
Température	Froid	Doux	0.5

Table 2-23: Calcul de l'index Gini pour la branche Vent = 'Fort'.

Pour les branches **Température= Froid** et **Température = Doux**, il ne reste plus d'attributs à traiter, donc on se retrouve dans un cas de nœud terminal. On regarde si tous les exemples appartiennent à la même classe. On prend la classe majoritaire comme nom pour chacun des nœuds. Ensuite, on ajuste l'erreur en conséquence.

Voici l'arbre produit par l'algorithme de construction CART à la figure 2.12

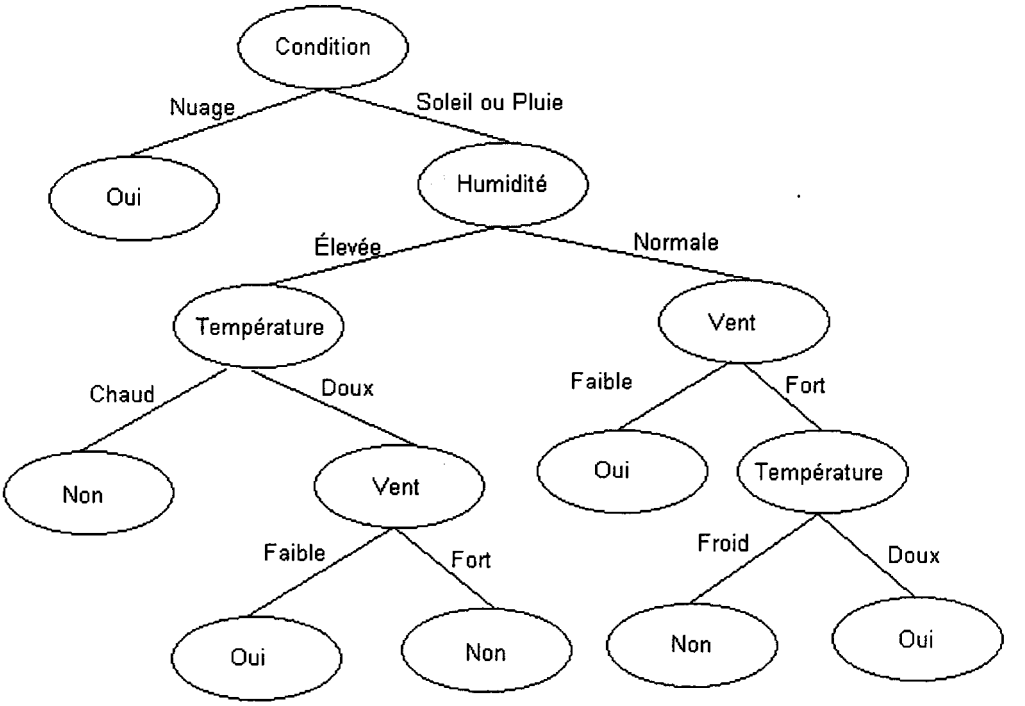

Figure 2.12 : Arbre de décision complet CART.

2.2.4 Khi deux.

Dans son tutoriel sur les arbres de décision, R. Rakotomalala[6] nous explique le fonctionnement de l'algorithme CHAID. CHAID est un algorithme de construction

d'arbre développé vers 1980 par Kass [35], il a été un des premiers algorithmes à être implanté dans des logiciels commerciaux. CHAID utilise le test du Khi-Deux. CHAID traite les attributs discrets avant les attributs continus.

$$\chi^2 = \sum_{k=1}^{K} \sum_{l=1}^{L} \frac{\left(n_{kl} - \frac{n_k * n_l}{n} \right)^2}{\frac{n_k * n_l}{n}} \qquad (2.7)$$

Pour chacun des attributs candidats, on construit un tableau de fréquence par classe selon les valeurs que cet attribut peut prendre (voir figure 2.13). Par exemple, si on a K classes et qu'un attribut peut prendre L valeurs.

Y/X	x_1	x_l	x_L	Σ
y_1		...		
y_k	...	n_{kl}	...	$n_{.l}$
y_K		...		
Σ		$n_{k.}$		n

Figure 2.13: Tableau du calcul.

Où n_{kl} représente le nombre d'éléments appartenant à la classe y_k pour la valeur x_l, $n_{k.}$ représente la sommation du nombre d'éléments de toutes les classes pour une certaine valeur de l'attribut, $n_{.l}$ représente la sommation du nombre d'éléments des valeurs possibles pour l'attribut calculé pour une certaine classe et n représente le nombre d'exemples dans le jeu d'apprentissage. On utilise ces valeurs dans l'équation 2.7.

On choisit comme nœud, l'attribut qui optimisera le test du khi deux. De plus, CHAID utilise une technique de pré élagage, c'est-à-dire qu'on arrête la croissance de l'arbre lorsque le test remplit une certaine condition.

Pour illustrer le comportement de l'algorithme, on se servira du jeu d'apprentissages de la table 2-1. Pour pouvoir calculer le test du Khi-Deux, il faut d'abord construire

un tableau pour chacun des attributs, les lignes représentent la répartition des classes et les colonnes représentent la répartition des valeurs de cet attribut, la dernière ligne et la dernière colonne représente la sommation.

Condition				
	Soleil	Nuage	Pluie	Σ
Oui	2	4	3	9
Non	3	0	2	5
Σ	5	4	5	14

A) Attribut Condition

Température				
	Chaux	Doux	Froid	Σ
Oui	2	4	3	9
Non	2	2	1	5
Σ	4	6	4	14

B) Attribut température

Humidité			
	Élevé	Normale	Σ
Oui	3	6	9
Non	4	1	5
Σ	7	7	14

C) Attribut Humidité

Vent			
	Faible	Fort	Σ
Oui	6	3	9
Non	2	3	5
Σ	8	6	14

D) Attribut Vent

Table 2-24: Calcul du khi-deux pour chacun des attributs pour déterminer la racine

À partir des tableaux des attributs de la table 2-24, on calculera le test du khi deux pour chacun des attributs.

Condition	3,54666667
Température	2,17037037
Humidité	2,8
Vent	0,93333333

Table 2-25 : Résultat du calcul du khi-deux pour déterminer la racine.

L'attribut Condition sera choisi comme nœud. L'attribut Condition a la valeur la plus élevée à la table 2-25. On répartira le jeu d'apprentissage pour chacune des valeurs. Quand il prend la valeur de nuage (voir table 2-2), les exemples appartiennent à la même classe, on aura donc un nœud terminal pour cette branche, pour les autres valeurs, il faut recommencer le processus avec les jeux correspondants à leurs branches respectives.

Pour la branche où la condition égale soleil, le jeu d'apprentissage correspond à la table 2-4.

	Température			
	Chaud	Doux	Froid	Σ
Oui	0	1	1	2
Non	2	1	0	3
Σ	2	2	1	5

A) Attributs Température

	Humidité		
	Élevé	Normale	Σ
Oui	0	2	2
Non	3	0	3
Σ	3	2	5

B) Attribut Humidité

	Vent		
	Fort	Faible	Σ
Oui	1	1	2
Non	1	2	3
Σ	2	3	5

C) Attribut Vent

Table 2-26: Calcul du khi deux pour chacun des attributs pour la branche Soleil.

On calcule le test du khi deux pour les attributs restants de la branche Condition = 'Soleil' (voir la table 2-26), soit pour la température, l'humidité et le vent.

Température	2,91666667
Humidité	5
Vent	0,13888889

Table 2-27: Résultat du calcul du khi deux pour chacun des attributs pour le sous arbre de la branche Soleil.

L'attribut Humidité sera choisi comme nœud (voir la table 2-27), les deux branches auront chacune un nœud terminal parce que chacune des valeurs est représentative d'une classe particulière.

Pour la branche où la condition égale pluie, le jeu d'apprentissage correspond à la table 2-5.

Température			
	Doux	Froid	Σ
Oui	2	1	3
Non	1	1	2
Σ	3	2	5

A) Attribut Température

Humidité			
	Élevé	Normale	Σ
Oui	1	2	3
Non	1	1	2
Σ	2	3	5

B) Attribut humidité

Vent			
	Fort	Faible	Σ
Oui	0	3	3
Non	2	0	2
Σ	2	3	5

C) Attribut Vent

Table 2-28: Calcul du khi deux pour chacun des attributs pour le sous arbre de la branche Pluie.

On calcule le test du khi deux pour les attributs restants de la branche Condition = 'Pluie' (voir la table 2-28), soit pour la température, l'humidité et le vent.

Température	0,13888889
Humidité	0,13888889
Vent	5

Table 2-29: Résultat du calcul du khi deux pour chacun des attributs pour le sous arbre de la branche Pluie.

L'attribut Vent répartit le jeu d'apprentissage en deux classes distinctes (voir la table 2-29), chacune des branches du nœud sera alors associée à un nœud terminal représentant une classe.

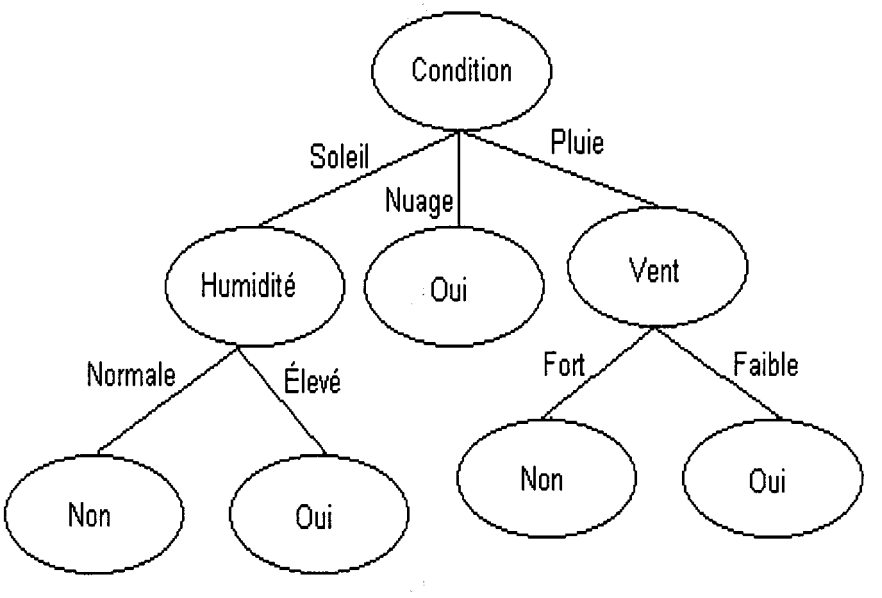

Figure 2.14 : Arbre de décision complet CHAID.

2.3 Attribut discret et Attribut continu.

Les attributs discrets sont des attributs qui ont un nombre limité de valeurs possible, C4.5 traite chacune des possibilités indépendamment, si un attribut choisi comme un nœud de l'arbre a *n* valeurs, le nœud aura *n* branches; CART produit des nœuds avec seulement deux branches, si un attribut a *n* valeurs, on a 2^{n-1}-1 possibilités de faire des regroupements de valeurs pour effectuer les tests [4], l'algorithme calcule l'index *GINI* de chacune de ces possibilités, on choisi le test qui maximisera l'index *GINI*.

Exemple : Un attribut quelconque a une possibilité de 4 valeurs différentes, soit x1, x2, x3 ou x4. Si on veut évaluer toutes les possibilités de regroupement binaire de valeurs pour cet attribut, on doit trouver les combinaisons de valeurs (voir la figure 2-30).

Valeur	Valeur
x1	x2, x3, x4
x2	x1, x3, x4
x3	x1, x2, x4
x4	x1, x2, x3
x1, x2	x3, x4
x1, x3	x2, x4
x1, x4	x2, x3

Table 2-30: Tableau des ensembles des regroupements possibles pour un attribut.

Les attributs continus sont des valeurs numériques contenues généralement dans un certain intervalle. L'approche la plus utilisée est de trier le jeu d'apprentissage selon l'attribut, de séparer les exemples par classes et ensuite de trouver un seuil ($d>\theta$ et $d\leq\theta$) qui convient le mieux pour éclater le plus équitablement le jeu d'apprentissage [1, 2, 3, 6, 11]. Un seuil est déterminé avec le point milieu entre 2 classes différentes.

$$\frac{v_1 + v_2}{2} \qquad (2.8)$$

Dans l'équation 2.8, v_1 représente la borne supérieure limite entre deux classes et v_2 représente la borne inférieure de la deuxième classe. En général, lorsqu'on analyse un attribut numérique par rapport aux classes, il y a plusieurs seuils de segmentation, on doit choisir le seuil qui favorise la répartition du jeu d'apprentissages.

Exemple : On a un jeu d'apprentissages qui contient un attribut qui est continu. Voici les valeurs de cet attribut et la classe qui lui est associée.

Valeur	Classe
70	Classe 1
72	Classe 2
72	Classe 1
75	Classe 1
80	Classe 1
83	Classe 2
85	Classe 1

Table 2-31 : Attribut continu.

Pour cet exemple de la table 2-31, on a 4 seuils possibles, soient 71, 72, 81.5 ou 84. On doit évaluer la mesure de segmentation Gini en utilisant chacun des seuils et on garde le meilleur résultat de classification.

2.4 Les erreurs de classification.

Pour valider une classification, il est important de prendre en considération les erreurs de classification, elles sont la conséquence de la création d'une feuille qui contient plus qu'une classe. C'est-à-dire que lors de l'appel de l'algorithme, le jeu d'apprentissage avait des exemples qui appartenaient à au moins deux classes différentes et que l'ensemble de tests à effectuer était vide. Pour calculer les erreurs de classification d'un arbre, il faut partir des feuilles à la racine (de bas jusqu'en haut de l'arbre).

2.4.1 L'erreur apparente.

L'erreur apparente d'une feuille représente le nombre d'exemples mal classifiés par cette feuille, celui de l'arbre est la somme des erreurs de toutes les feuilles. On calcule le taux d'erreur de classification avec la formule suivante :

$$Taux = \frac{nbErreurs}{nbExemple} \qquad (2.9)$$

2.4.2 L'estimation de l'erreur réelle.

Cette estimation est surtout utilisée pour l'élagage de l'arbre, on utilise une approche ascendante. Avec le paramètre de confiance *CF*. Pour chacune des feuilles de l'arbre, notons N le nombre d'exemples qu'une feuille couvre et E le nombre d'erreurs de classification qu'elle induit dans l'échantillon. Soit p, la probabilité pour qu'un nouvel exemple soit mal classé par cette feuille. La valeur de p est trouvée à l'aide de la fonction suivante, où p est une valeur entre 0 et 1:

$$U_{CF}(N,E) = P\left(\sum_{i=0}^{E} \binom{N}{i} p^i (1-p)^{N-i} >= CF\right) \qquad (2.10)$$

Par la suite, on remonte jusqu'à la racine, en calculant l'estimation de l'erreur réelle de cet arbre en faisant une somme pondérée des estimations des erreurs réelles de ses fils.

Pour calculer l'erreur d'un sous-arbre, on doit prendre en compte l'erreur de chacun des nœuds rattachés au sous-arbre. Par exemple, si un sous-arbre a n fils, soit A_1,..., A_n, si le nombre d'exemples couverts par chacun de ces nœuds est respectivement de N_1, N_2,..., N_n, et si les erreurs réelles estimées pour chacun de ces fils sont e_1,..., e_n.

$$\frac{\sum_{i=1}^{n} ei * Ni}{\sum_{i=1}^{n} Ni} \qquad (2.11)$$

2.5 Élagage.

Pour éviter le sur apprentissage, c'est-à-dire qu'on a deux nœuds qui contiennent un seul élément, on enlève les sous-arbres les moins significatifs de l'arbre afin de les remplacer par des feuilles avec un seuil d'erreur acceptable [3, 11], de cette façon, on généralise les règles extraites de l'arbre [12].

La phase d'élagage consiste à enlever les feuilles les moins significatives de l'arbre. Cette phase est exécutée après la construction de l'arbre. On considère que la racine de l'arbre comprend au moins deux feuilles. Tant qu'il existe un sous arbre que l'on peut remplacer par une feuille, sans faire croître l'estimation de l'erreur réelle, alors on élague ce sous arbre. On doit savoir si les nœuds fils sont des feuilles ou des sous-arbres. Si tous les nœuds sont des feuilles, on remplace la racine du sous-arbre par une feuille, si l'erreur de la nouvelle feuille est plus petite que celle de l'ancien sous-arbre.

Pour créer la feuille de remplacement, on crée la liste des possibilités d'éléments des feuilles du sous-arbre et on prend la valeur la plus fréquente pour le nom de l'étiquette et le(s) autre(s) dans le tableau Etiquette des autres possibilités. On remonte ensuite jusqu'à la racine jusqu'à ce qu'on ne puisse plus remplacer un sous-arbre par une feuille [20].

2.5.1 Pré élagage.

Le pré élagage se produit pendant la construction de l'arbre, il agit comme un critère d'arrêt dans l'expansion de l'arbre. Il consiste à fixer une condition d'arrêt pour arrêter la construction [6,8]. Cette condition limite l'expansion d'une branche, c'est-à-dire que les attributs restants ne permettent plus de diviser les exemples. Si on n'effectuait pas cet arrêt exceptionnel, la branche croîtrait en prenant les attributs restants. Les exemples restants au niveau du nœud que le seuil atteint seraient les mêmes que celle au niveau du nœud terminal. Dans les algorithmes C4.5 et CART, lorsque la mesure de segmentation est égale à zéro ou à l'infini, on doit créer un nœud terminal. Il vaut mieux arrêter la phase d'expansion de la branche que de continuer sans avoir d'apport significatif dans la classification.

2.5.2 Post élagage.

Le post élagage est la méthode la plus utilisée dans la plupart des algorithmes, elle s'effectue une fois que l'algorithme d'expansion est terminé [3, 6,11]. C4.5 utilise

une estimation de l'erreur réduite de l'arbre [3,11], cette estimation est produite à partir de l'erreur apparente de l'arbre.

2.5.3 Élagage sur l'erreur réduite réelle.

C4.5 utilise une estimation de l'erreur pour l'élagage de l'arbre. On utilise une approche ascendante. Avec le paramètre de confiance *CF*. Pour chacune des feuilles de l'arbre, notons N le nombre d'exemples qu'une feuille couvre et E le nombre d'erreurs de classification qu'elle induit dans l'échantillon. Soit p, la probabilité pour qu'un nouvel exemple soit mal classé par cette feuille. La valeur de p est trouvée à l'aide de la fonction suivante, où p est une valeur entre 0 et 1 :

Par la suite, on remonte jusqu'à la racine, en calculant l'estimation de l'erreur réelle (voir l'équation 2.10) de cet arbre en faisant une somme pondérée des estimations des erreurs réelles de ses fils.

Exemple : Soit le jeu d'apprentissages de la table 2-32 :

Exemple	*Couleur*	*Durée*	*Modèle*	*Garanti*
X1	*Rouge*	*2 mois*	*A*	*1 an*
X2	*Bleu*	*3 mois*	*C*	*3 ans*
X3	*Rouge*	*1 mois*	*B*	*2 ans*
X4	*Rouge*	*3 mois*	*C*	*3 ans*
X5	*Bleu*	*1 mois*	*A*	*1 an*
X6	*Rouge*	*2 mois*	*B*	*1 an*
X7	*Rouge*	*1 mois*	*B*	*2 ans*
X8	*Bleu*	*3 mois*	*C*	*3 ans*
X9	*Rouge*	*2 mois*	*C*	*2 ans*
X10	*Bleu*	*1 mois*	*B*	*1 an*
X11	*Bleu*	*1 mois*	*A*	*1 an*
X12	*Rouge*	*3 mois*	*C*	*2 ans*

Table 2-32 : Jeu d'apprentissages parc automobile.

On obtient l'arbre de la figure 2.15 en utilisant l'approche C4.5 :

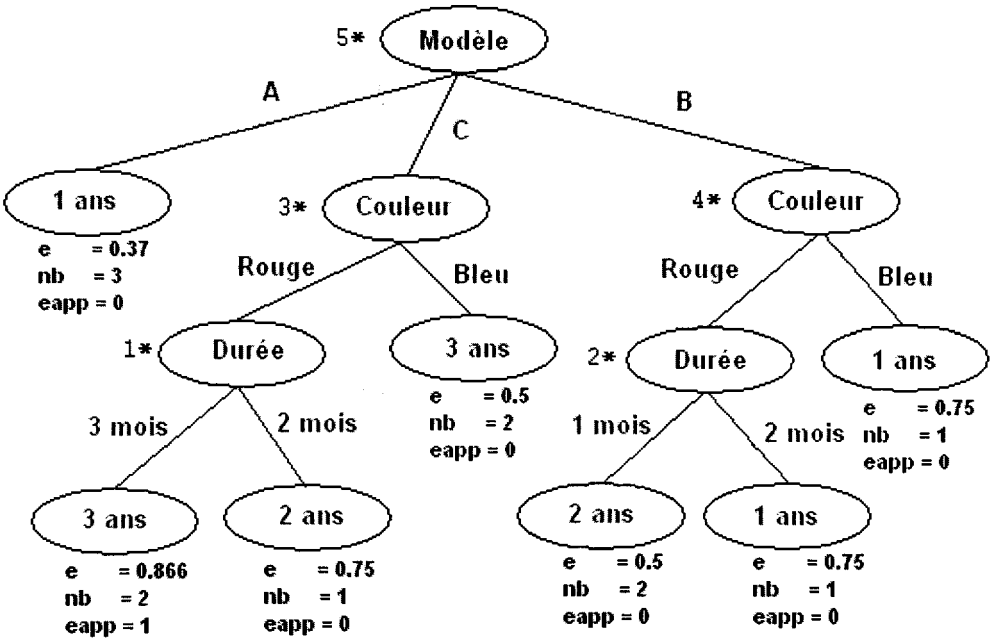

Figure 2.15: Arbre de décision avant élagage.

Si on remplace *1** par une feuille, la nouvelle erreur sera de 0.673, cette valeur est plus petite que l'erreur courante du sous-arbre *1 **, donc on remplace le sous-arbre *1** par une feuille avec la valeur la plus fréquente. On diminuera alors l'estimation de l'erreur totale de l'arbre.

Si on remplace *2** par une feuille, la nouvelle erreur sera 0.696, cette valeur est plus grande que l'erreur courante du sous-arbre *2**, donc on ne remplace pas ce sous-arbre (voir la figure 2.16).

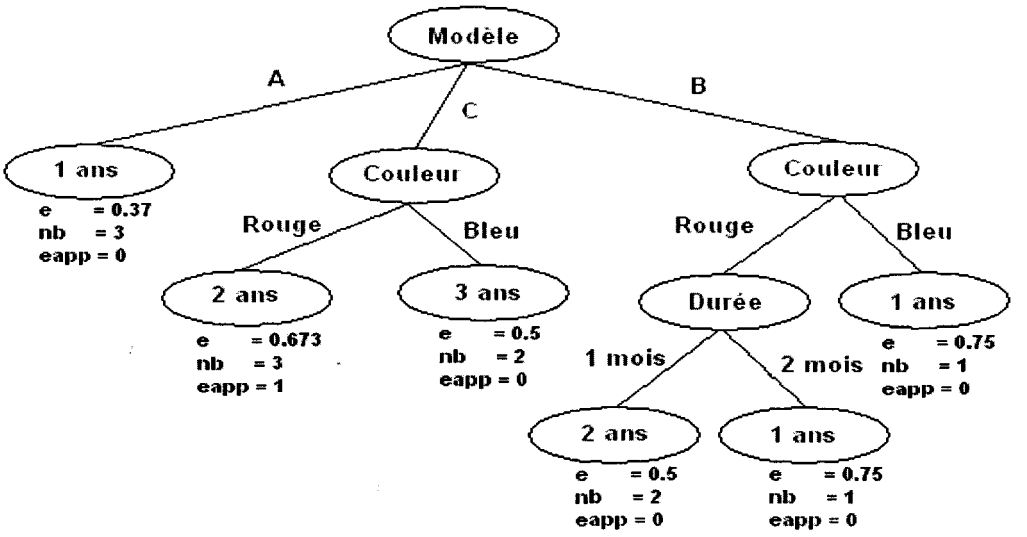

Figure 2.16: Arbre de décision après élagage.

2.5.4 Coût de complexité

CART produit une série d'arbres, soit T_0, T_1, T_2,..., T_k, où T_0 est l'arbre obtenu à la fin de la phase d'expansion de l'arbre et T_k représente seulement la racine de l'arbre, les autres arbres sont représentés par l'élagage successif de l'arbre T_1 jusqu'à ce que l'arbre soit simplement une feuille [4]. Pour cela, CART utilise le coût de complexité [4,11] à partir de l'erreur produite par le jeu d'apprentissage, contrairement à C4.5, CART utilise un jeu de tests ou aussi appelé jeu d'élagage afin de choisir l'arbre qui minimisera le taux d'erreur.

Pour passer de T_k à T_{k+1}, on doit élaguer un où plusieurs nœuds. Pour cela, on établi une liste de nœuds potentiels à élaguer, les nœuds choisis doivent être obligatoirement un sous arbre, dans la figure 4, les nœuds potentiels sont *n1*, *n2*, *n3*, *n4*, *n5*, et *n6*. On mesure le critère pour choisir le nœud qu'on va élaguer.

(2.11)

$$Critère\ (T_k,d) = \frac{MC\ (d,k) - MCT(d,k)}{N(k) \times (Nt(d,k) - 1)}$$

Où *MC (d,k)* est le nombre d'exemples mal classés du jeu d'apprentissage par le nœud d de l'arbre T_k quand on fait l'hypothèse qu'il a été transformé en feuille, *MCT(d,k)* est le nombre d'exemples mal classés par les feuilles du nœud T_k situé sous le nœud d, *N(k)* représente le nombre de feuilles de T_k, Nt (d,k) représente le nombre de feuilles du sous arbre de T_k situé sous le nœud *d.*

Exemple : Voici l'arbre produit par l'algorithme CART avec la table 2-1. Les nœuds non terminaux ont été numérotés de n1 à n6 (voir la figure 2.17).

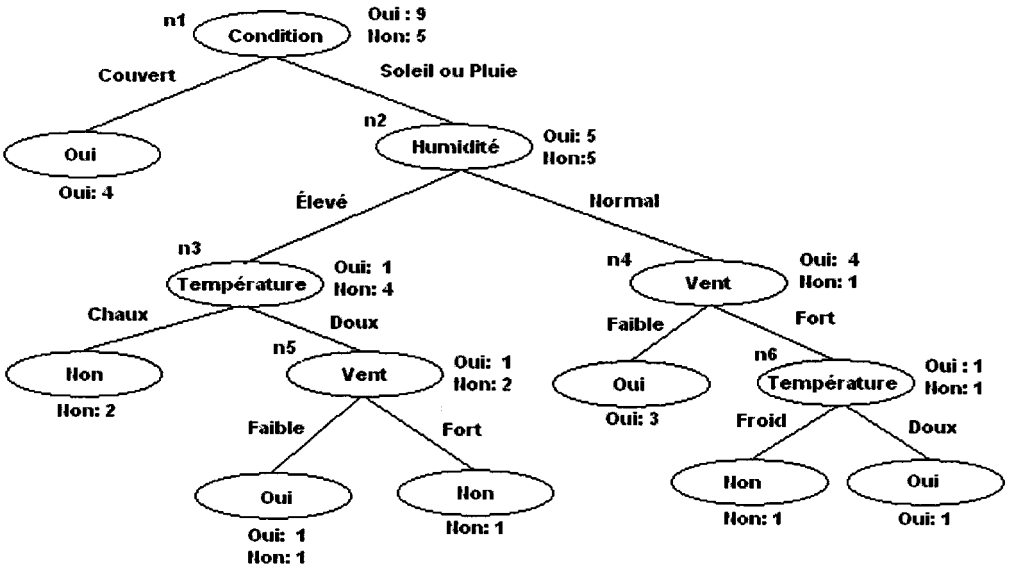

Figure 2.17: Arbre T_0.

À partir de l'arbre de la figure 2.17, on cherche à trouver le nœud qu'on va élaguer pour passer de l'arbre originaire T_0 à T_1. Il faut donc mesurer le critère d'élagage et de choisir le nœud qui minimisera la fonction.

$$0.0095238 = \quad Critère(T_0,n1) = \frac{5-1}{7\,x\,(7-1)}$$

$$0.112857 = \quad Critère((T_0, n2) = \frac{5-1}{7\,x\,(6-1)}$$

$$0 = \quad Critère(T_0,n3) = \frac{1-1}{7\,x\,(3-1)}$$

$$0.007142 = \quad Critère(T_{max}, n4) = \frac{1-0}{7 \times (3-1)}$$

$$0 = \quad Critère(T_{max}, n5) = \frac{1-1}{7 \times (2-1)}$$

$$0.14\,28714 = \quad Critère(T_{max}, n6) = \frac{1-0}{7 \times (2-1)}$$

Le nœud n3 est remplacé par une feuille, la classe attribuée à cette feuille est la classe qui représente la plus fréquente, soit non (voir la figure 2.18).

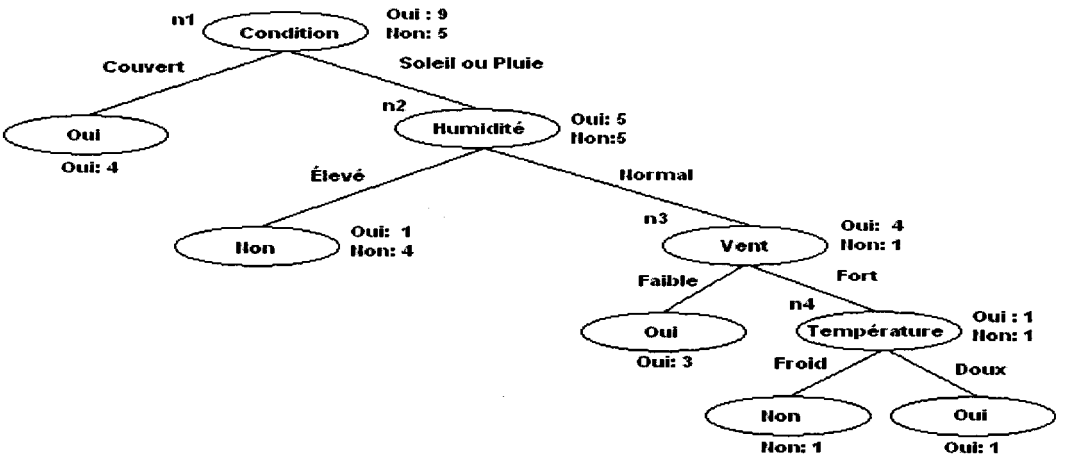

Figure 2.18: Arbre T_1.

$$0.2 = \quad Critère(T_1, n1) = \frac{5 - 1}{5 \times (5 - 1)}$$

$$0.266666 = \quad Critère(T_1, n2) = \frac{5 - 1}{5 \times (4-1)}$$

$$0.006666 = \quad Critère(T_1, n3) = \frac{1-0}{5 \times (3-1)}$$

$$0.1 = \quad Critère(T_1, n4) = \frac{1-0}{5 \times (2-1)}$$

On choisit d'élaguer le nœud n3 pour passer de T_1 à T_2. La feuille sera représentée par oui (voir la figure 2.19).

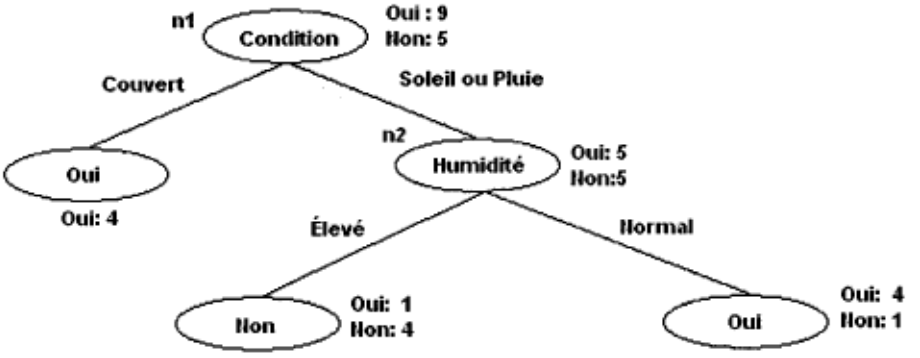

Figure 2.19. : Arbre T_2.

On recommence le processus avec l'arbre T_2.

$$0.5 = \quad Critère(T_2,n1) = \frac{5-2}{3 \times (3-1)}$$

$$1 = \quad Critère(T_2,n2) = \frac{5-2}{3 \times (2-1)}$$

On choisit d'élaguer le nœud n1 pour passer de T_2 à T_3 (voir la figure 2.20)

Figure 2.20 : Arbre T_3.

Le processus d'élagage se termine lorsque la racine de l'arbre est une feuille. Ensuite, on choisit avec le jeu de tests, l'arbre qui convient le mieux à la classification.

2.6 Importance du jeu d'apprentissages et de tests

Le jeu d'apprentissage est le jeu avec lequel on construit l'arbre. Il est construit à partir des données d'entrée. Le jeu de tests est le jeu qui permet de valider la classification. Il a les mêmes caractéristiques que le jeu d'apprentissage, en général,

les exemples contenus dans le jeu de test sont différents du jeu d'apprentissage pour voir la validité des règles de décision produites par l'arbre.

2.6.1 Préparation du jeu d'apprentissages et du jeu de tests

La répartition du jeu de données entre le jeu d'apprentissages et le jeu de tests est une chose très importante à considérer. Dans le cas contraire, on aura un jeu débalancé. Chawla présente des méthodes pour construire un jeu de données et éviter d'avoir un jeu débalancé [14]. Il doit avoir une bonne répartition entre les classes dans le jeu d'apprentissages et le jeu de tests, le but d'un arbre de décision est de classifier des cas non vus pendant la construction de l'arbre; par exemple : une classe qui se retrouve uniquement dans le jeu de tests, c'est-à-dire que l'arbre ne connait pas cette classe, donc cette classe ne sera pas reflétée dans les règles.

2.6.2 Validation croisée (Cross-validation)

Pour effectuer une validation croisée, on doit séparer en *n* sous-ensembles [2, 3, 4, 5], par exemple dans la validation à 10 sous-ensembles, on sépare aléatoirement le jeu de données en 10 sous-ensembles, la représentation des classes doit être la même que dans le jeu de données, on en choisi 9 sous-ensembles pour faire le jeu d'apprentissage et l'autre sous-ensemble pour le jeu de test. On construit un arbre de décision à partir du jeu d'apprentissages et on teste les règles avec le jeu de tests. Ensuite, on recommence le processus jusqu'à chacun des sous-ensembles soit testé. À la fin du processus, on aura dix ensembles de règles de décision. Pour avoir une approche plus générale, on peut combiner les règles de façon à diminuer les conflits entre les règles [12]. La validation croisée est une autre technique pour éviter le sur apprentissage [5].

2.7 Acquisition de règles de classification

Un arbre de décision est un bon outil pour représenter des connaissances. Il représente une connaissance experte en utilisant les nœuds pour les attributs, les branches pour les valeurs des attributs et les feuilles pour les classes [10]. Une règle

représente une série de conditions à respecter les caractéristiques d'une classe en particulier. Chacune des feuilles de l'arbre représente une règle de classification.

Les règles de décision sont l'interprétation directe d'un arbre de décision, une règle de décision représente la lecture d'une branche de l'arbre de la racine à un nœud terminal de l'arbre. Un arbre contient plusieurs règles. Une règle de décision est constituée d'une ou plusieurs conditions et d'une classe avec laquelle elle est associée. Une condition est composée d'un attribut à tester, d'un signe et de valeurs de l'attribut qui répond à la condition. Les valeurs d'une condition sont séparées par une clause **OU** et les conditions d'une règle sont séparées par une clause **AND**. Voici un exemple d'une règle de décision :

IF Condition 1 = Val 1 OU Val 2 AND ... AND Condition N = Val X THEN Classe X

Pour évaluer les règles, on se sert du jeu de tests. Pour qu'un exemple corresponde à une règle particulière, il doit respecter toutes les conditions de cette règle, s'il respecte toutes les conditions et que la classe de l'exemple correspond à la classe de la règle, on incrémente le nombre d'exemples qui correspond à cette règle. Par contre, si un exemple respecte toutes les conditions d'une règle et que la classe de l'exemple est différente du développement de la règle, on incrémente l'erreur de cette règle.

2.8 Aspect de recherche

Malgré que les méthodes existantes pour la construction d'arbre de décision, il y a encore des recherches sur les arbres de décision, certaines se concentrent sur les données inconnues dans le jeu d'apprentissages [15], d'autres essaient de faire évoluer génétiquement l'arbre [7, 16] ou choisit une segmentation floue [18, 19].

2.8.1 Données manquantes ou incohérentes

Les données manquantes ou incohérentes peuvent avoir des conséquences désastreuses sur la classification. Certains algorithmes de construction d'arbre

essaient d'évaluer les coûts des données dans le calcul de segmentation [15]. C4.5 introduit un facteur de correction pour incorporer les données manquantes dans le calcul du ratio de gain [2, 3, 11]. Les données manquantes sont généralement représentées pas des valeurs nulles. Les données incohérentes peuvent être provoquées par des fautes d'orthographe ou encore plus qu'un nom pour désigner une valeur pour un attribut quelconque. Une autre manière de traiter les valeurs manquantes est simplement de les traiter comme une des possibilités de valeurs pour l'attribut. Les données manquantes sont un des principaux échecs dans une classification.

2.8.2 Arbre de décision évolutif

Les arbres évolutifs sont le croisement entre les arbres de décision et les algorithmes génétiques, ils ont adapté les opérateurs génétiques (croisement et mutation) pour fonctionner avec des arbres [16]. Les algorithmes génétiques sont capables de résoudre des problèmes réels de façon optimale [17]. La combinaison des algorithmes génétiques et les arbres de décision peuvent être bénéfiques, elle permettra de produire des règles qui seront plus optimales sans qu'elles soient trop spécifiques.

L'opérateur de mutation choisit un nœud au hasard et remplace ce nœud par une nouvelle valeur aléatoire et l'opérateur de croisement choisit aléatoirement deux nœuds et échange les sous-arbres des nœuds [16].

2.8.3 Arbre de décision 'flou'

Les arbres 'Fuzzy' apportent des résultats significatifs pour la classification linguistique [17]. Un arbre 'flou' est construit avec les mêmes méthodes que les arbres conventionnels, à la différence qu'au lieu de choisir uniquement le meilleur attribut, la méthode sélectionne d'autres attributs alternatifs qui sont proches du meilleur choix. Avec les autres possibilités, on peut construire des arbres parallèles (voir la figure 2.21).

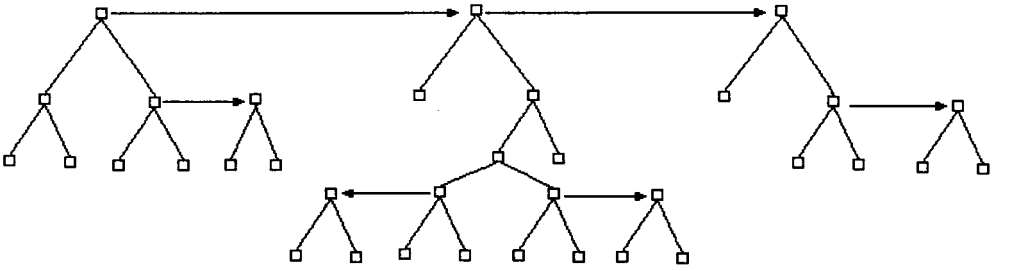

Figure 2.21 : Schéma d'un arbre flou.

Il n'y a pas seulement au niveau du choix de l'attribut à considérer, on peut essayer de trouver des similitudes au niveau des valeurs, par exemple, le mot 'nation' est contenu dans le mot 'nationalisme'. les méthodes 'floues ' recherchent entre autres les similitudes dans les chaînes de caractères, de façon à regrouper ses valeurs pour former une seule valeur. Dans la classification de texte, le nombre de mots est souvent très élevé, il est donc nécessaire de regrouper certains mots similaires afin d'augmenter la qualité des règles que les arbres produiront.

Les arbres de décision 'flous' sont nés de la fusion des arbres conventionnels de décision et des ensembles 'flous '.Cette amélioration au pouvoir de représentation des arbres de décision ajoute des notions de logique floue, menant à une meilleure robustesse et à des applications dans des contextes incertains ou incomplets. Les arbres de décision floue assument que tous les domaines des attributs ou les variables linguistiques ont été prédéfinis dans des termes flous [17].

La phase d'induction consiste à suivre les étapes suivantes [18].

La fuzzification du jeu d'apprentissage : Le jeu d'apprentissage est traité avec des fonctions utilisant la logique floue pour regrouper les attributs ayant des caractéristiques similaires.

La génération de l'arbre de décision Fuzzy : Les arbres de décision Fuzzy sont construits avec les mêmes algorithmes de constructions conventionnelles, plusieurs

des algorithmes ont été adaptés pour prendre en considération les ensembles Fuzzy. Fuzzy ID3 utilise le gain informationnel en utilisant la logique floue [18,19].

L'extraction des règles Fuzzy de l'arbre de décision Fuzzy : L'extraction des règles Fuzzy s'effectue de la même manière qu'avec les règles de décision conventionnelle,

Application des règles Fuzzy pour la classification : Pour appliquer les règles des décisions Fuzzy, le jeu de test doit aussi subir les mêmes transformations que le jeu d'apprentissage, de cette façon, le résultat de l'évaluation des règles qu'on a générées sera plus conforme avec la réalité.

CHAPITRE 3 ALGORITHME GÉNÉTIQUE

3.1 Introduction sur les algorithmes génétiques (AG)

Les AG sont des algorithmes basés sur le mécanisme de sélection naturelle et de processus génétiques, ces algorithmes, basés sur la théorie de l'évolution de Darwin [27], sont apparus vers 1975, ils ont été développés par John Holland de l'université du Michigan. Le principe de base consiste à simuler le processus d'évolution naturel dans un environnement quelconque plus ou moins hostile. Ils combinent les structures les plus appropriées qui survivent à l'évaluation, pour former un algorithme de recherche avec certains facteurs d'intuition. À partir de deux structures, il combine l'information des structures pour créer une nouvelle génération. Dans chacune des générations, un ensemble nouveau de chaînes est créé en utilisant les différentes caractéristiques de l'ancienne population.

Les AG utilisent un vocabulaire similaire à celui de la génétique (voir la table 3-1) [21].

Concept dans l'évolution naturelle	Concept dans les AG
Chromosome	Chaîne, individu, solution, hypothèse
Gène	Caractéristique ou détecteur
Locus	Position dans une chaîne
Allèle	Valeur de la caractéristique
Génotype	Structure
Phénotype	Ensemble de paramètres, solution alternative, une structure décodée

Table 3-1: Concept de base des AG.

Les AG sont devenus des outils importants dans l'apprentissage automatique, le forage de données et pour l'optimisation de fonctions non linéaires. Pour résoudre une tâche d'apprentissage, une tâche de modélisation ou une tâche d'optimisation, un AG maintient une population de chromosomes, et modifie la population selon les opérateurs génétiques, en recherchant une solution la plus proche de la solution optimale pour une tâche donnée. Un individu dans la population est représenté généralement par une chaîne de caractères. Chaque caractère de la chaîne représente une caractéristique quelconque.

Les AG font partie de la famille des algorithmes évolutifs, c'est-à-dire des algorithmes probabilistes qui maintiennent une population d'individus qui vont évoluer de génération en génération pour obtenir un résultat se rapprochant de la solution optimale. Les meilleurs individus d'une génération vont créer une nouvelle génération plus adaptée au problème. Bien qu'utilisant le hasard, les AG ne sont pas purement aléatoires, car on suit des lois probabilistes.

Dans un premier temps, chacune des solutions est évaluée. Cette évaluation permet de juger de la pertinence des solutions par rapport au problème considéré. Ceci conduit à éliminer les solutions jugées inutiles ou très mauvaises, on va donc mettre à l'écart les individus les plus faibles pour favoriser les plus performants.

Un algorithme génétique peut être appliqué à la production d'une variété d'objets, tant qu'il est possible d'obtenir une note représentant la justesse de la solution.

Le mécanisme d'encouragement du plus fort permet alors l'apparition du prédicateur reclassant le mieux les données. Ce type de construction de prédicateur fait partie des techniques de forage de données et porte le nom d'« induction ».

Avant d'appliquer un algorithme génétique à un problème, il faut définir [5]:

La fonction de performance : C'est la fonction avec laquelle on évalue un individu par rapport à la population. Cette fonction est directement reliée au problème.

La représentation des individus : C'est la signification de chacun des gènes d'un individu.

La façon de sélectionner des individus : C'est la manière de sélectionner des individus pour la reproduction. Est-ce qu'on applique la loi du plus fort où on donne une petite chance aux individus qui sont faibles

La façon de reproduction des individus : C'est l'échange des gènes entre les couples d'individus sélectionnés.

3.1.1 Fonctionnement d'un AG

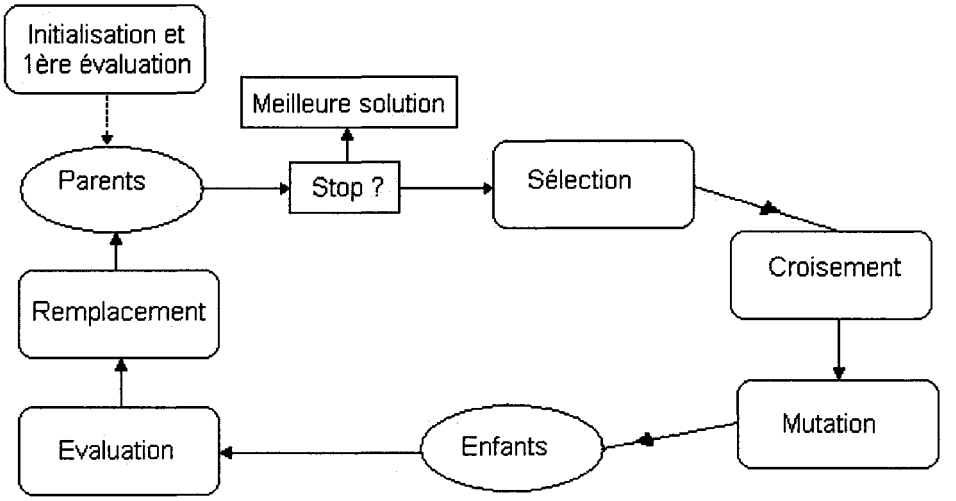

Figure 3.1 : Fonctionnement d'un algorithme génétique.

Le fonctionnement d'un AG est assez simple, il consiste à simuler le processus de sélection naturelle, plusieurs auteurs font beaucoup d'analogies avec la biologie pour expliquer le fonctionnement des AG. La figure 3.1 décrit les principales étapes d'un algorithme génétique.

La première étape consiste à initialiser la population en encodant dans une structure d'un chromosome.

Avec la fonction de performance, on évalue chaque membre de la population pour déterminer sa performance au sein de la population.

La nouvelle génération est créée à partir des individus sélectionnés de la génération précédente et ayant croisé entre eux et muter les gènes, afin de créer de nouveaux individus. Ensuite, on réapplique le processus sur la nouvelle population. L'algorithme s'arrête quand le meilleur individu d'une génération a dépassé un certain

stade ou alors quand un certain nombre d'individus qui sont identiques à la solution locale optimale.

3.1.2 Algorithme d'un AG

1) Génère une population de k hypothèses $P = \{s_1, ..., s_k\}$;
2) Calcul la Fitness(s_i) de chaque membre de l'ensemble P ;
3) Tant que l'AG ne satisfait pas un critère d'arrêt,
a) Sélection de k_1 pair de membres de la population;
b) Croiser chaque couple pour produire deux nouveaux individus dans la population E;
c) Pour chaque membre de l'ensemble E, selon la probabilité de mutation, appliquer la mutation sur un gène d'un chromosome s_i ;
d) Choisir k survivants dans la population P et E, pour remplacer l'ensemble P;
e) Calcul la Fitness(s_i) de chaque membre de l'ensemble P .

On peut définir la fonction Fitness(s_i) selon le problème qu'on a résoudre. La fonction décode les chromosomes pour les évaluer par rapport aux autres chromosomes de la population. Par exemple, dans le problème du commis-voyageur qui doit parcourir x villes, où chacune des villes doit être visitée une seule fois, la fonction de fitness est optimise la route du commis-voyageur en optimisant la plus coute distance entre les villes.

3.2 Opérateurs génétiques

Les opérateurs génétiques permettent de manipuler la population, ils ne travaillent pas seulement sur un individu, mais sur l'ensemble de la population. Les opérateurs génétiques jouent un rôle important dans la réussite d'un AG. Les principaux sont l'opérateur de sélection, l'opérateur de croisement et l'opérateur de mutation. Chacun de ces opérateurs agit selon certains critères qui leur sont propres. Ces opérateurs effectuent un processus qui modifie la nature de l'individu, ce changement a pour but de trouver une meilleure combinaison des individus et ainsi trouver une solution optimale.

3.2.1 Codage de la structure génétique

Pour initialiser la population d'individus, il faut d'abord coder l'ensemble des chromosomes qui représente la population. Un chromosome représente une solution d'un problème quelconque. La codification d'un chromosome est généralement binaire, c'est-à-dire qu'il est représenté par une chaîne de 0 ou de 1 (Ex. : 10001101). Le chromosome est composé d'un ou de plusieurs gènes. Chaque gène représente une caractéristique. L'efficacité de l'algorithme génétique va donc dépendre du choix du codage d'un chromosome. Il existe trois principaux types de codage :

Codage binaire

C'est le type de codage le plus utilisé. Le chromosome est représenté par une suite de bits en codage binaire, par exemple une suite de longueur n, $A = \{a_1, a_2, \ldots, a_n\}$ $\forall i \in [1, n]$, où $a_i \in V = \{0,1\}$

Cependant, le codage binaire comporte quelques problèmes, il peut être difficile d'adapter ce codage à certains problèmes, la résolution de l'algorithme peut être coûteuse en temps ou le croisement et la mutation peuvent être inadaptés parce qu'il crée un individu qui n'appartient pas à l'espace de recherche.

Codage réel

Dans ce type de codage, le gène est représenté par un nombre réel, pour certains types de problème d'optimisation, on a besoin d'aller chercher une certaine précision numérique sur un certain intervalle ou encore avoir des valeurs négatives.

Le codage binaire associe à chaque variable réelle une sous-chaîne de symboles binaires dont la longueur dépend de la précision souhaitée. Une alternative à ce codage est de considérer que chaque variable réelle correspond à un et un seul symbole du génotype : il s'agit du "codage réel " d'un génotype. La taille de

l'alphabet des symboles est alors identique au nombre de valeurs possibles pour chaque variable, nombre qui peut être très grand. Chaque gène est considéré comme équivalent au nombre entier, ou réel, qu'il représente.

Codage de Gray

Dans un codage binaire, on utilise souvent la "distance de Hamming " comme mesure de la différence entre deux éléments de la population, cette mesure compte les différences de bits de même rang de ces deux séquences. Et c'est là que le codage binaire commence à montrer ses limites. En effet, deux éléments voisins en termes de distance de Hamming ne codent pas nécessairement deux éléments proches dans l'espace de recherche. Cet inconvénient peut être évité en utilisant un "codage de Gray" : le codage de Gray est un codage qui a comme propriété : Entre un élément n et un élément n + 1, donc voisin dans l'espace de recherche, un seul bit diffère (voir la table 3-2) [21].

Entier	Binaire	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111

Table 3-2: Comparaison entre le code entier, binaire et Gray.

3.2.2 Initialisation de la population

Cette opération consiste à coder les chaînes de bits représentant les caractéristiques de chacune des instances de la population initiale. On peut considérer chacune des instances comme une solution potentielle à un problème. L'initialisation de la population est généralement aléatoire, mais on peut se servir de solution existante pour initialiser la population.

3.2.3 Sélection

Cette opération consiste à sélectionner les instances de la population qui se reproduiront pour créer la nouvelle génération. Comme son nom l'indique, la sélection vise à sélectionner des individus à partir d'une population actuelle. C'est un des plus importants opérateurs puisqu'il est responsable de la survie, de la reproduction et de la mort de chacun des individus de la population. Généralement, la probabilité de survie d'un individu est reliée directement à sa performance au sein de la population.

Qualité d'une bonne méthode de sélection

Une bonne méthode de sélection permet qu'une grande partie de la population active puisse survivre à la prochaine génération. À chaque génération, quelques individus qui sont parmi les plus performants sont sélectionnés pour créer de nouveaux individus. Même si les individus faibles ont une chance faible d'être sélectionnée, ils ont une chance de participer à la production de nouveaux individus. Sinon, on risque de perdre un certain bagage génétique qui pourrait être utile comme un phénomène d'adaptation. La population doit être composée d'un nombre limité d'individus.

Il existe diverses techniques de sélection :

Tirage de roulette

Cette technique consiste à donner une chance proportionnelle à la performance d'un individu, chaque individu possède une case sur une roulette donc la taille est reliée à son adaptation. On lance une boule sur la roulette et l'individu qui possède la case où la roulette est tombée est sélectionné. C'est une méthode assez classique, mais elle possède une variance forte, un individu mauvais a la même chance d'être sélectionné qu'un autre individu plus fort, c'est le hasard qui détermine la case de la roulette. Si on est vraiment malchanceux, on peut sélectionner des individus avec des facteurs de performance faible [21, 26, 28]. Dans ce problème, le facteur de fitness est trouvé en transformant l'individu de la base 2 à la base 10. Pour calculer les chances de sélection de chacun des individus de la table 3-3, on transforme le facteur du fitness en pourcentage par rapport à la somme de tous les individus.

Exemple : On a une population qui contient les éléments suivants :

Individu	Fitness	Chance de sélection	Intervalle associé
0001111	15	37,5%	0 à 0.375
0001100	12	30%	0.3751 à 0.675
0000110	6	15%	0.6751 à 0.825
0000111	7	17.5%	0.8251 à 1

Table 3-3: La population de l'exemple.

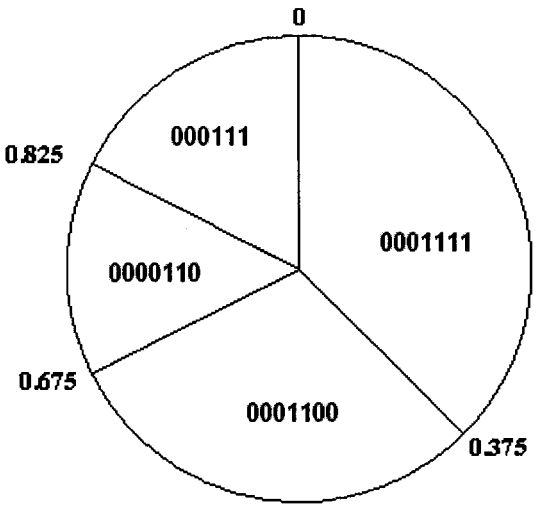

Figure 3.2: Représentation de la roulette.

Pour effectuer une sélection d'un individu, on tire aléatoirement un nombre réel entre 0 et 1, le chromosome, donc l'intervalle associé correspondant, sera choisi pour faire partie d'un couple pour la phase de reproduction. La table 3-3 représente la roulette de la figure 3.2 utilisée pour la sélection des individus pour le couplage.

Sélection par tournois

Cette technique consiste à partir de *n* individus de choisir le meilleur individu qui pourra participer à un tournoi, on organise autant de tournois qu'il y a d'individus à sélectionner. Le nombre n permet de donner plus ou moins de chance aux individus moins forts. Avec un nombre élevé de participants, un individu faible est presque sûr de perdre lors du tournoi [26, 28, 29].

k-Élitisme :

Cette technique consiste à trier l'ensemble de la population selon leur facteur de performance et de sélectionner les *k* premiers. Les individus faibles n'ont vraiment aucune chance d'être sélectionnés. La méthode consiste à appliquer la loi du fort [21, 22, 26, 28]. Cependant, cette méthode n'est pas la meilleure méthode, parce qu'elle

élimine une certaine diversité de gènes qui se trouve uniquement dans un individu faible.

3.2.4 Évaluation

Cette opération consiste à évaluer les instances pour voir quelle est leurs performances, une certaine solution peut avoir par rapport aux autres. Ce facteur est appelé un facteur de performance, ce facteur permet de déterminer un certain classement des solutions caractérisées dans la population. On définit au préalable la fonction d'adaptation qui associe à chaque phénotype une valeur dite d'adaptation, c'est la note de l'individu, plus elle est élevée, plus la solution donnée par ce phénotype est optimale.

Cette fonction décode le chromosome pour évaluer la valeur d'un individu, elle associe un coût à chaque chromosome de la population. Il faut distinguer entre la fonction objective et la fonction d'adaptation. Dans certains cas, elles peuvent être identiques, mais en général la fonction d'adaptation dépend de la fonction objective, laquelle dépend de la nature du problème. La fonction d'adaptation peut être à objectif unique ou à objectif multiple.

3.2.5 Croisement (Crossover)

Cette opération consiste à croiser deux structures parents pour former deux nouvelles structures différentes. À partir de deux individus, on obtient deux nouveaux individus qui héritent de certaines caractéristiques de leurs parents. Le croisement sélectionne des gènes parmi deux individus appelés parents. À partir de ces gènes sont générés les enfants. La probabilité d'hybridation représente la fréquence à laquelle les hybridations sont appliquées.

Le croisement, qui symbolise la reproduction sexuée (toujours par métaphore du mécanisme de sélection naturelle), est une des étapes majeures de l'AG. C'est l'instrument majeur des innovations dans l'algorithme. Il peut être effectué de

plusieurs manières, mais la plus courante croise les chaînes de caractères de deux individus parents pour former des chaînes de caractère enfants.

Cet opérateur permet la création de deux nouveaux individus. Cependant, un individu sélectionné ne subira pas nécessairement l'action de croisement, l'action ne s'effectue seulement avec une certaine probabilité. Plus cette probabilité est élevée et plus la population subira des changements. Le croisement est la clef de la puissance des AG. Il est directement relié à la capacité d'une population d'individu d'explorer un espace de recherche et de combiner les meilleurs résultats.

La première étape du croisement est de déterminer un point de coupure, à partir du point, on sépare les gènes des chromosomes pour les échanger avec l'autre chromosome, on obtiendra deux nouveaux chromosomes. La figure 3.3, nous montre le fonctionnement du croisement des couples d'individus.

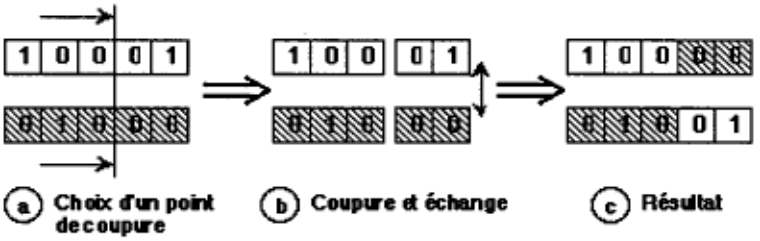

Figure 3.3: Exemple de croisement.

Il existe plusieurs méthodes de croisement :

Croisement à un point :

À partir d'un point de coupure, choisi au hasard, sur deux génotypes, on échange les parties situées après le point de coupure choisi pour permettre de créer de nouveaux génotypes [29]. La figure 3.4 illustre le cheminement du croisement à un point de coupure.

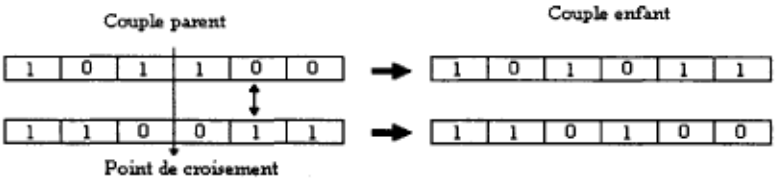

Figure 3.4: Croisement à un point.

Croisement à deux-points :

Selon le hasard, on choisit deux points de croisement et on échange les parties entre les deux-points [29]. La figure 3.5 illustre le cheminement du croisement à deux points de coupure.

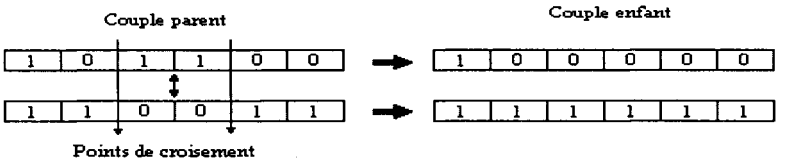

Figure 3.5: Croisement à deux-points.

Croisement à k points :

Cette méthode de croisement consiste en une généralisation à k points de coupure de la méthode de croisement à deux-points. La figure 3.6 illustre le cheminement du croisement à k points de coupure, dans cet exemple k = 4.

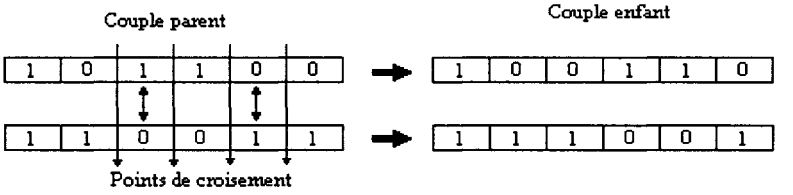

Figure 3.6: Croisement à k-points où k = 4.

Croisement uniforme :

Le croisement uniforme peut être vu comme un croisement multipoint avec un nombre de points de coupure qui n'est pas connu a priori. On utilise un masque binaire de la même longueur que les génotypes qui est généré aléatoirement pour chaque couple. On applique un masque sur les deux individus, ce masque doit être de même taille que le génotype. Par la suite, on applique les opérations de base, on échange des parties du génotype. Un "0" à la nième position du masque laisse inchangé les symboles à la nième position des deux génotypes, un "1" déclenche un

échange des symboles correspondants. Le masque est généré aléatoirement. La figure 3.7 illustre le cheminement du croisement uniforme.

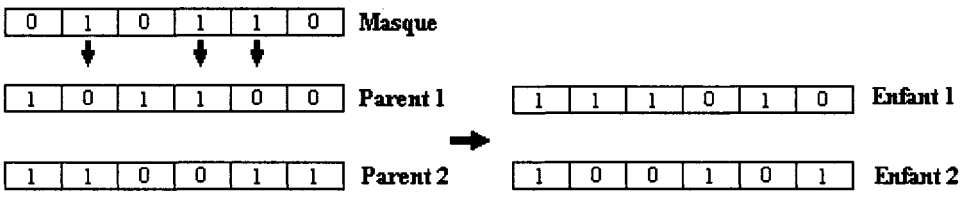

Figure 3.7 : Croisement uniforme.

3.2.6 Mutation

Cette opération consiste à changer, selon une certaine probabilité qui est généralement très petite, un ou plusieurs bits d'un gène. Pour introduire une certaine diversité dans les solutions, entre autres ça évite que les nouvelles solutions convergent vers la même structure. Le but est d'éviter à l'AG de converger vers une valeur extrême locale de la fonction d'évaluation et de permettre de créer des éléments originaux. Si l'individu créé est faible, il est éliminé.

Comme pour le croisement, la mutation vise à modifier de façon aléatoire une partie de la population. Ici, le principe est de choisir une valeur de remplacement aléatoire pour l'un des gènes des individus de la population concernés. La probabilité de mutation représente la fréquence à laquelle les gènes d'un chromosome sont mutés. Si la mutation est appliquée, une partie du chromosome est changée, sinon l'individu nouveau est inséré dans la population sans aucun changement. Même si la mutation permet d'éviter de converger vers un extrême local, si elle est trop fréquente, l'AG est orienté vers une recherche aléatoire de la bonne solution. Il existe diverses méthodes pour la mutation.

Mutation aléatoire

Dans le cas du codage binaire, c'est une technique populaire, si un gène qui doit muter est égal à 1, la valeur du gène est inversée à 0, et vice-versa. La figure 3.8 illustre le fonctionnement de la mutation aléatoire.

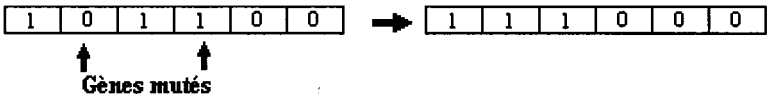

Figure 3.8: Représentation d'une mutation aléatoire.

Mutation uniforme

La mutation uniforme consiste à appliquer un masque sur un individu à l'aide d'une variable aléatoire de distribution uniforme sur l'espace de recherche. La mutation d'un gène quelconque est effectuée lorsque le gène du masque est égal à 1. La figure 3.9 illustre le fonctionnement de la mutation uniforme.

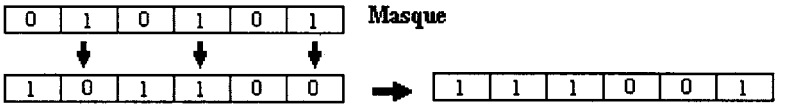

Figure 3.9: Représentation d'une mutation uniforme.

3.2.7 Remplacement de la population

Cette opération consiste à incorporer les nouvelles solutions dans la population, chacune des nouvelles solutions est évaluée. Comme dans le processus de l'évolution, les plus forts sont appelés à survivre et les plus faibles sont appelés à mourir. Les solutions qui ont un plus petit facteur de fitness sont appelées à disparaître pour introduire les nouvelles solutions qui sont généralement meilleures.

3.3 Les paramètres d'un AG

Les opérateurs d'un AG sont guidés par un certain nombre de paramètres fixés à l'avance. La valeur de ces paramètres a une grande influence sur la réussite ou non d'un AG. Les principaux sont [26] :

La taille de la population

Si la taille est très élevée, le temps de calcul peut s'avérer très important, par contre s'il est trop petit, l'algorithme peut converger rapidement vers une mauvaise solution.

Le nombre de générations

Le nombre de générations est un paramètre qui permet d'éviter de s'enliser dans une convergence partielle et permet d'arrêter l'AG lorsque le nombre maximal est atteint. Il est préférable qu'il soit assez grand afin de mieux visualiser la convergence de la solution.

La probabilité de croisement

La probabilité dépend généralement de la fonction de performance. Plus elle est élevée, plus la population subira des changements importants. En général, cette probabilité est comprise entre 0.5 et 0.9.

La probabilité de mutation

Cette probabilité est généralement faible puisqu'un taux élevé risquerait de conduire l'AG vers une solution non optimale.

3.4 Simulation d'un SGA *(Simple Genetic Algorithm)*

Goldberg [21], nous explique le fonctionnement d'un algorithme génétique à l'aide d'un SGA, la population est constituée de 4 chaînes qui représentent un chiffre en représentation binaire et la fonction de performance est représentée par $f(x) = x^2$.

No	Chaîne	Performance	% du Total
1	"01101"	169	14,4
2	"11000"	576	49,2
3	"01000"	64	5,5
4	"10011"	361	30,9
Total		1170	100

Table 3-4 : population initiale du SGA.

À partir de la population initiale de la table 3-4, on sélectionne des couples d'individus (chaînes) pour se reproduire à l'aide d'une roulette. Chaque case de la roulette représente un individu de la population selon leurs probabilités de sélection.

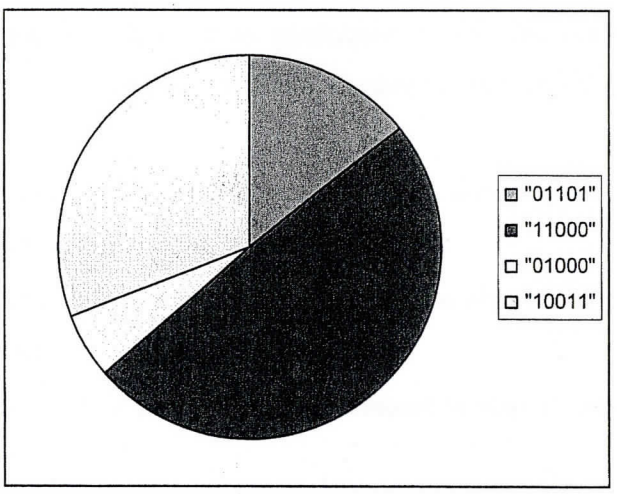

Figure 3.10: Représentation d'une roulette de sélection

No	Valeur de x	f(x)	Compagnon	Chaîne avec le point de croisement	Nouvelle Pop	nouveau x	Nouveau facteur
1	13	169	2	"0110|1"	"01100"	12	144
2	24	576	1	"1100|0"	"11001"	25	625
3	8	64	4	"11|000"	"11011"	27	729
4	19	361	3	"10|000"	"10001"	17	289
	Somme	1170					1787
	Moyenne	293					446
	Max	576					729

Table 3-5: Itération d'un SGA.

Après avoir sélectionné les couples au hasard, on a jumelé le numéro 1 avec le numéro 2 et le numéro 3 avec le numéro 4.

Pour le croisement, le hasard a déterminé le point de croisement pour chacun des couples, le point de croisement pour le couple (1, 2) est à la position 4 et pour le (3, 4) est à la position 2. La mutation s'est effectuée seulement sur le $5^{ième}$ gène du chromosome 4 de la nouvelle population. La nouvelle population remplace complètement l'ancienne. Le processus recommence jusqu'à qu'un critère d'arrêt soit atteint, soit le nombre de générations maximums a été atteint ou soit la solution optimale a été trouvée.

3.5 Les limitations des algorithmes génétiques

Malgré le fait que les algorithmes génétiques soient des outils efficaces pour l'optimisation des solutions d'un problème, ils sont limités quand même sur plusieurs aspects.

- La nécessité d'une fonction d'encodage et de décodage pour évaluation de la performance des individus.
- Les AG sont difficiles à programmer, chaque algorithme est très dépendant du problème.
- Les AG ont besoin de beaucoup de ressources pour effectuer des calculs en parallèle.
- Les paramètres sont très importants pour la performance des AG, trop souvent ils sont déterminés selon l'expérience de l'utilisateur.
- Il y a des limites pratiques sur le nombre illimité d'itérations.
- Il y a une limite sur la taille hypothétique d'une population.
- Il n'y a pas de garantie quant à l'obtention de la solution optimale au problème posé en un temps fini
- L'utilisation de ces algorithmes est souvent coûteuse en temps de calculs

3.6 Les algorithmes génétiques au service de la classification

Les algorithmes génétiques utilisent des classificateurs (classifier) comme individus, un classificateur est une règle encodée avec le théorème des schémas.

3.6.1 Théorème des Schémas

Un schéma est un modèle de similarité décrivant un sous-ensemble de chaînes avec des similarités à certaines positions dans la chaîne. Dans un codage binaire, l'alphabet du codage est composé par les exemples 0 ou 1, un schéma introduit un nouveau caractère dans l'alphabet : '#'. Ce caractère spécial veut dire qu'on ne se préoccupe pas de ce caractère (don't care symbol). Le schéma représente toutes les chaînes qui possèdent des caractères équivalents à la même position dans la chaîne, les positions, où apparaissent le symbole '#' ne sont pas prises en compte dans la comparaison [21].

Par exemple, supposons qu'on a un ensemble de chaînes de 10 caractères, le schéma suivant {#111100100} représente deux chaînes, soit {(1111100100), (0111100100)} et un autre schéma {#1#11001100} représente quatre chaînes, soit {(11111001100), (01111001100), (11011001100), (01011001100)}, le schéma 1111100100 représente seulement une chaîne et le schéma (##########) représente toutes les chaînes de l'ensemble de la population.

Un schéma correspond à 2^r chaînes, où r correspond au nombre de symboles '#'. Une chaîne de longueur m correspond à 2^m schémas. Les schémas permettent de créer des sous-ensembles de recherche. Il est plus facile de caractériser un individu dans la population [21, 22, 23].

3.6.2 Représentation de la codification de règle

Les AG ont besoin pour fonctionner d'une population d'individus, chaque individu représente une règle quelconque. Pour que l'AG soit capable d'effectuer ses opérateurs génétiques (la sélection, le croisement et la mutation), on doit encoder la règle. Celle-ci peut avoir une ou plusieurs conditions. Chaque règle correspondra à un chromosome et les conditions de la règle correspondront aux gènes du

chromosome. Généralement, le codage binaire qui est utilisé pour la codification, même si quelques fois la représentation binaire n'est pas suffisante [22].

La longueur d'un gène en particulier est déterminée selon les différentes représentations qu'une condition peut prendre, dans le cas d'un attribut binaire, on a besoin d'un seul caractère pour représenter le gène dans le chromosome. Cependant pour représenter un attribut continu ou encore un attribut ayant plus de 2 valeurs, la longueur du gène peut varier. Pour représenter un attribut ayant *n* valeurs, on a besoin d'un gène de longueur *n*. Chacun des bits du gène représentera une des valeurs de l'attribut. Par exemple, on veut encoder un gène pour représenter l'attribut Couleur, l'attribut peut prendre comme valeur : rouge, orange, bleu, jaune et vert. La table 3-6 illustre les gènes représentés par les couleurs.

Couleur	**Gène**
Rouge	10000
Orange	01000
Bleu	00100
Jaune	00010
Vert	00001

Table 3-6: Gène pour représenter un attribut.

Si on veut encoder une condition où la couleur est soit rouge, soit jaune ou encore non bleu, le gène pour représenter cette condition sera 1#01#.

Pour représenter un attribut continu, on doit être en mesure d'encoder tous les intervalles entre la borne minimale et la borne maximale des valeurs, pour cela, on utilise un radius [26]. Supposons qu'on a un ensemble d'entiers X = {1, ..., n}, où le radius est (n-1)/2, la chaîne du gène sera représentée par $b_1,\ldots, b_n$, où $b_j = 1$ si x appartient à l'intervalle [j – radius, j + radius]. Par exemple, on a n = 5, voici les intervalles à respecter pour chacune des positions du gène :

{[-1, 3], [0, 4], [1, 5], [2, 6], [3, 7]}

On représente chacun des entiers individuellement.

$1 \Rightarrow 11100$ $2 \Rightarrow 11110$ $3 \Rightarrow 11111$ $4 \Rightarrow 01111$ $5 \Rightarrow 00111$

Maintenant, on est en mesure d'encoder tous les intervalles imaginables.

$[1,2] \Rightarrow 111\#0$ $[1,3] \Rightarrow 111\#\#$ $[1,4] \Rightarrow \#11\#\#$ $[1,5] \Rightarrow \#\#1\#\#$

$[2,3] \Rightarrow 1111\#$ $[2,4] \Rightarrow \#111\#$ $[2,5] \Rightarrow \#\#11\#$ $[3,4] \Rightarrow \#\#111$

$[3,5] \Rightarrow \#\#111$ $[4,5] \Rightarrow 0\#111$

CHAPITRE 4 IMPLÉMENTATION DES ARBRES DE DÉCISION ET DES ALGORITHMES GÉNÉTIQUES

4.1 Implémentation d'un arbre de décision

L'implémentation d'un arbre de décision se fait selon le paradigme "diviser pour régner », c'est-à-dire qu'on utilise un phénomène de récurrence pour construire l'arbre ou pour aller chercher des informations dans l'arbre.

4.2 Structure interne d'un nœud d'un arbre de décision

Un nœud est soit un nœud terminal ou un sous arbre, un nœud terminal est un nœud avec une décision ou une classe. Un sous arbre est un nœud qui possède des descendants. Un nœud contient nécessairement les informations suivantes :

Une étiquette : C'est le nom du nœud dans l'arbre, il représente soit un attribut, si le nœud a des descendants ou une classe, si le nœud est terminal.
La colonne référence au jeu d'apprentissages : Cette valeur est utilisée pour indiquer aux règles, la position de l'attribut dans le jeu d'apprentissages pour l'évaluation.
Le tableau des branches : C'est le tableau qui contient le nom des branches. Dans un arbre de décision, les branches représentent les valeurs des attributs choisis pour le nœud.
Le tableau d'enfants : Ce tableau contient d'autres nœuds de niveau inférieur.

Les informations suivantes sont complémentaires :
La classe majoritaire (CART) : C'est la classe la plus représentative dans le jeu d'apprentissages. Lors de l'élagage du nœud, cette information servira à remplacer le nom de la feuille.
Le nombre d'exemples classifiés par ce nœud : C'est le nombre d'exemples dans le jeu d'apprentissages qui sont classifiés au niveau du nœud.
Le nombre d'erreurs de classification : C'est le nombre d'exemples qui n'appartient pas à la classe majoritaire. On l'appelle aussi l'erreur apparente de l'arbre.

L'estimation du taux d'erreur réel (C4.5) : Cette erreur est calculée en utilisant l'erreur apparente, elle est utilisée dans l'élagage de l'arbre, c'est la valeur qui détermine si le nœud sera élagué ou non.

Le tableau des classes : C'est l'ensemble des classes qui font parti du jeu d'apprentissage au niveau du nœud.

Le tableau des cardinalités des classes : Ce tableau contient les cardinalités de chacune des classes.

Jeu d'apprentissages et tableau des attributs non utilisés : Le jeu d'apprentissages au niveau du nœud contient les exemples classés par le nœud et le tableau des attributs non utilisés contient les indexes restant du processus de construction de l'arbre, on garde ces informations pour pouvoir aller rechercher de l'information sur les similarités entre les exemples classés au niveau de ce nœud.

4.2.1 Algorithme général de construction d'un arbre de décision

L'algorithme de construction a besoin d'un jeu d'apprentissages, des attributs à traiter et de l'objectif de classification. Le jeu d'apprentissages est une matrice (voir figure 4.1 fenêtre A), où chacune des lignes est représentée par un exemple et les colonnes sont représentées par les attributs, les attributs à traiter (voir figure 4.2 fenêtre B) sont représentés dans un tableau d'index, où chacune des colonnes représente la colonne de l'attribut dans le jeu d'apprentissages et l'objectif de classification (voir figure 4.2 fenêtre C) représente la colonne de la classe dans le jeu d'apprentissages.

Étape 1 : Vérifier si on est en présence d'un nœud terminal. Lorsque le nœud est une feuille, on arrête l'expansion de la branche rattachée au nœud.

On est en présence d'un nœud terminal si on respecte une des conditions suivantes :

Tous les exemples appartiennent à la même classe.

Tous les attributs ont été exploités.

On a atteint un seuil critique dans le calcul pour le critère d'éclatement du jeu d'apprentissages.

Étape 2 : Calcul du critère d'éclatement pour tous les attributs à traiter.

Pour C4.5, on utilise soit le gain informationnel ou le ratio de gain. Le nombre de branches est associé au nombre de valeurs que l'attribut peut prendre.

Pour CART, on utilise le critère d'index Gini. Avec le critère Gini, on peut seulement avoir 2 branches par nœud. Si un attribut a n valeurs, alors il y a 2^{n-1}-1 possibilités de regroupement. Il faut donc calculer l'index Gini pour chacun des regroupements possibles des valeurs de cet attribut.

Étape 3 : Choix de l'attribut pour représenter le nœud.
Étape 4 : Trouver les valeurs de l'attribut associé au nœud.
Étape 5 : On enlève l'index de l'attribut choisi dans le tableau des attributs à traiter et pour chacune des valeurs, on associe une branche au nœud. On répète les étapes 1 à 5 pour chacune des branches avec les exemples du jeu correspondants à la branche.

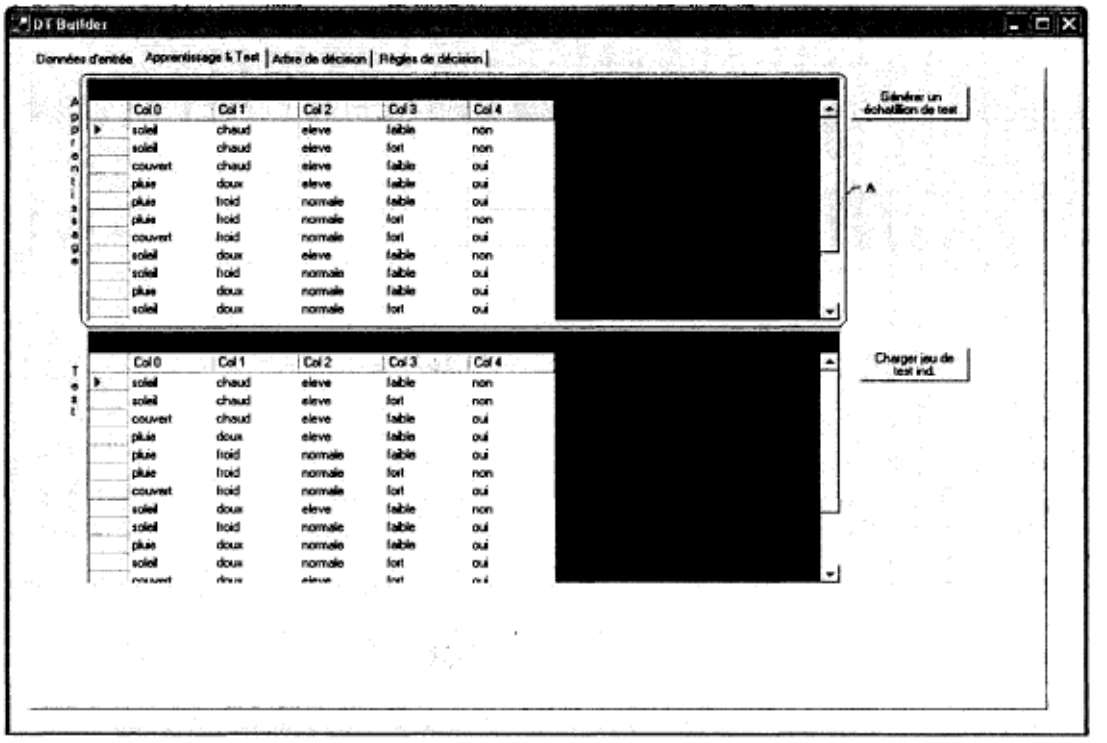

Figure 4.1: Fenêtre du jeu d'apprentissages et du jeu de tests.

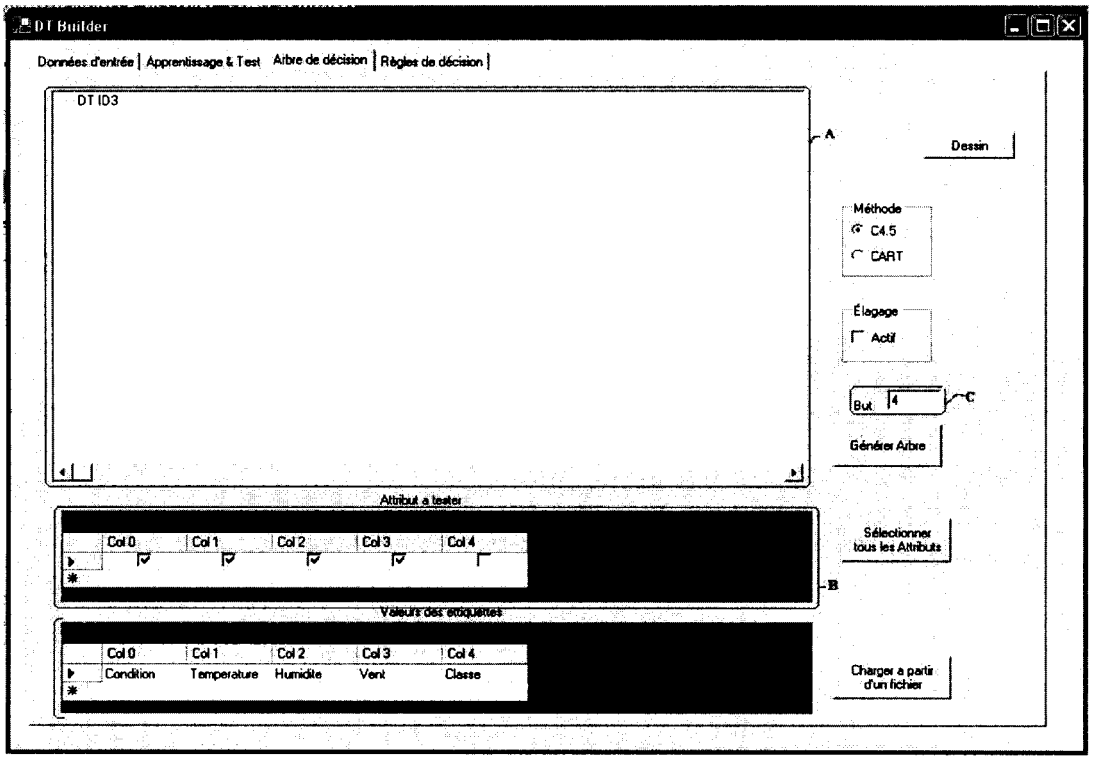

Figure 4.2: Fenêtre de génération d'un arbre de décision.

4.3 Implémentation de l'élagage C4.5

L'élagage de l'arbre s'effectue selon un certain coefficient de confiance *CF*, plus ce coefficient est élevé, plus l'arbre se dirigera vers le sur apprentissage, c'est-à-dire que la majorité des nœuds terminaux ne classent seulement qu'un seul exemple par nœud. Dans le cas opposé, si le *CF* est petit, l'arbre peut être représenté seulement par la racine, c'est-à-dire la classe majoritaire dans le jeu d'apprentissages. Par défaut, ce coefficient est fixé à 25%.

Tant qu'il existe un sous-arbre que l'on peut remplacer par une feuille sans faire croître l'erreur réelle estimée alors, on doit élaguer ce sous-arbre. On doit savoir si les nœuds fils sont des feuilles ou des sous arbres, si tous les nœuds descendants du sous arbre sont des feuilles. On crée une feuille qui pourrait remplacer le sous-arbre, on regarde la classe majoritaire pour déterminer la classe qui sera affectée au nœud. On ajuste en conséquence, le nombre d'exemples classé et le nombre d'erreurs

apparentes, on calcule à partir de ces valeurs, l'erreur réelle de ce nœud. On remplace le sous-arbre par le nouveau nœud l'erreur du nouveau nœud est plus petite que celle de l'ancien sous arbre.

4.4 Implémentation de l'élagage CART

Pour effectuer la phase d'élagage de *CART*, on doit effectuer l'élagage successif de l'arbre-source, on aura une suite d'arbres de T_0 à T_{max}, T_0 est l'arbre-source et T_{max} est représenté seulement par la racine de l'arbre. Pour passer de T_i à T_{i+1}, on doit enlever le sous-arbre le moins significatif de l'arbre pour le remplacer par une feuille. Pour cela, on regarde le coût de complexité. Chacun des sous arbres contenant des nœuds terminaux est évalué avec un jeu de tests, ensuite on transforme le sous arbre qui minimisera l'erreur de classification en un nœud terminal.

Pour le choix de l'arbre de décision, on utilise un jeu de tests indépendant du jeu d'apprentissages et on choisit l'arbre qui minimisera l'erreur de classification.

4.5 Implémentation des règles de décision

Les règles de décision sont la principale sortie des arbres de décision, chacune des règles de décision générées est l'expression d'un chemin entre la racine et le nœud terminal. Pour chacun des nœuds terminaux, il y a une règle de décision. Une règle de décision est une expression booléenne composée (voir figure 4.3) :

Une classe associée à cette règle : C'est la classe pour laquelle la règle est vraie.
Le nombre de classifications réussies : C'est le nombre d'instances dans le jeu de tests qu'ils remplissent et qui ont la bonne classe.
Le nombre d'erreurs à la règle : C'est le nombre d'instances dans le jeu de test qui respectent les conditions de la règle et qui appartiennent à une autre classe.
Index restant de la construction de l'arbre : C'est un tableau contenant l'index des attributs n'ayant pas servi à la construction de l'arbre, lors de l'extrapolation de la règle, ce tableau servira à explorer les attributs non utilisés.

Les exemples du jeu d'apprentissages classés par cette règle : Ce sont les exemples du jeu d'apprentissages qui ont contribué à la construction de l'arbre, on garde les exemples pour mieux explorer les différents attributs que l'algorithme de construction a oubliés lors de son exécution.
Une série de conditions : C'est un tableau de conditions, pour qu'une règle soit évaluée, toutes les conditions doivent être vraies.

Chacune des conditions est un test à effectuer sur le jeu de test, voici les informations contenues dans une condition :
Type du test : Il s'agit de savoir si le test est effectué sur des attributs discrets ou continus.
Attribut testé : C'est le nom de l'attribut qui sera testé lors de l'évaluation de cette condition.
Colonne de référence : C'est l'index de la donnée dans le jeu de tests.
Symbole du test : C'est le symbole utilisé pour le test, pour un attribut discret, le symbole sera '=', par contre, si le type du test est continu, alors le symbole sera '<=' ou ' >'.
Valeur associée : C'est la valeur qui correspond à la véracité de la condition.

Ces informations permettront d'évaluer la condition, une condition est soit vraie ou fausse.

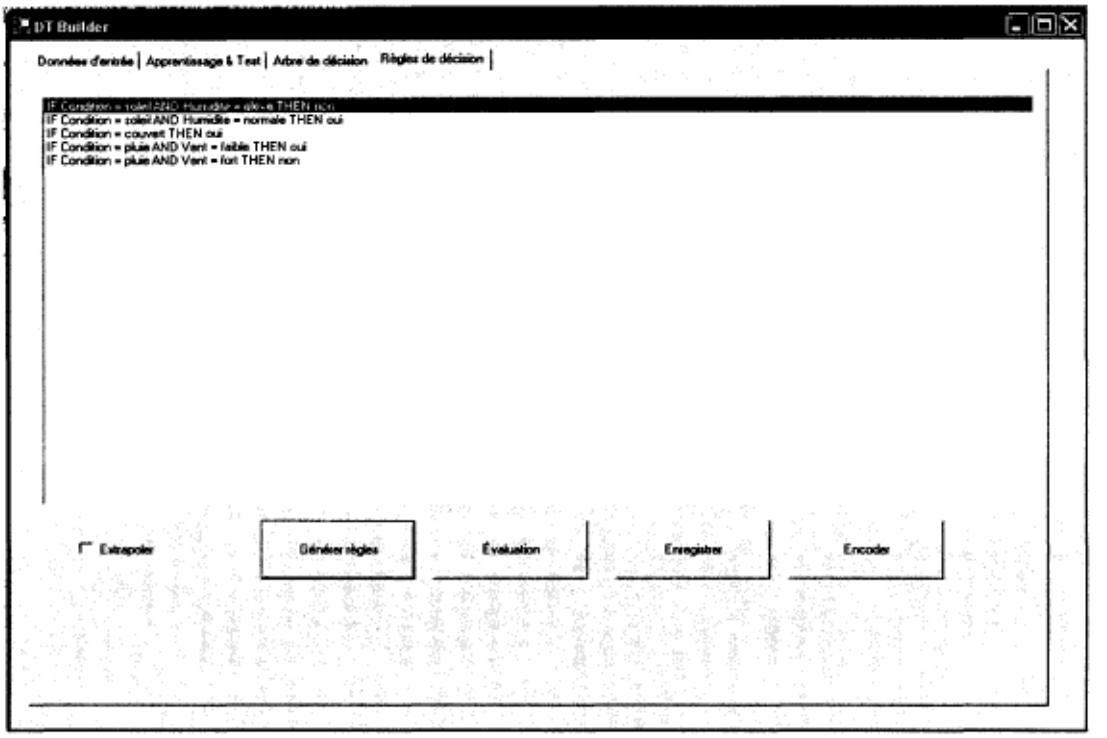

Figure 4.3: Fenêtre de représentation des règles de décision.

La figure 4.3 contient les règles de décision extraites à partir d'un arbre. Elle contient les règles suivantes :

IF Condition = soleil AND Humidité = élevé THEN Non

IF Condition = soleil AND Humidité = normale THEN Oui

IF Condition = couvert THEN Oui

IF Condition = pluie AND Vent = faible THEN Oui

IF Condition = pluie AND Vent = fort THEN Non

Ces règles sont l'interprétation directe de chacune des branches de l'arbre de décision.

4.5.1 Fonction de génération de règles de décision

Pour chacun des nœuds terminaux de l'arbre de décision, on aura une règle de décision, les nœuds terminaux représentent les classes associées aux règles de décision. On génère les règles de décision avec un arbre de décision. On remonte

chacune des branches de l'arbre de décision jusqu'aux nœuds terminaux de gauche à droite. Pour chacun des nœuds d'une branche entre la racine et le nœud terminal de la branche, on aura une condition. Le nom de l'attribut du nœud sera l'attribut de test de la condition, le symbole du test est déterminé par la nature de l'attribut, la valeur associée à la branche liée au nœud terminal associé à la règle.

4.5.2 L'extrapolation des règles de décision

L'extrapolation d'une règle de décision est l'exploration des données complémentaires à la règle. Elle recherche dans les exemples, qui ont permis la construction de la branche, les similitudes des exemples pour chacun des attributs non traités. L'extrapolation des règles donne vraiment des résultats significatifs lorsqu'il y a au moins deux exemples qui ont contribué à cette règle. Dans le cas contraire, on risque d'avoir l'exemple comme règle. On ajoute les conditions pour les attributs qui ont seulement une valeur associée à cet attribut, de cette façon, on ajoute une information complémentaire à la règle.

4.5.3 Fonction d'évaluation de règles de décision

Pour l'évaluation, nous avons besoin de définir un jeu de tests, ce jeu peut être indépendant du jeu d'apprentissages. On évalue chacune des règles de décision avec chacun des exemples.

Pour chacune des règles de décision, on évalue chacune des conditions, lorsque toutes les conditions de la règle de décision sont vraies, on regarde si la classe de l'exemple correspond à classe de la règle de décision. Si la règle est vraie, on incrémente le nombre de classification, sinon on incrémente l'erreur.

4.6 Implémentation d'un système de classification

D. Goldberg [21] implémente un système de classification pour représenter l'apprentissage d'un multiplexeur à 11 bits. Notre système de classification est une

version adaptée pour classifier des segments de textes avec un algorithme génétique (voir figure 4.4). Un système de classification est composé de plusieurs composants interagissant les uns avec les autres afin d'optimiser les classificateurs.

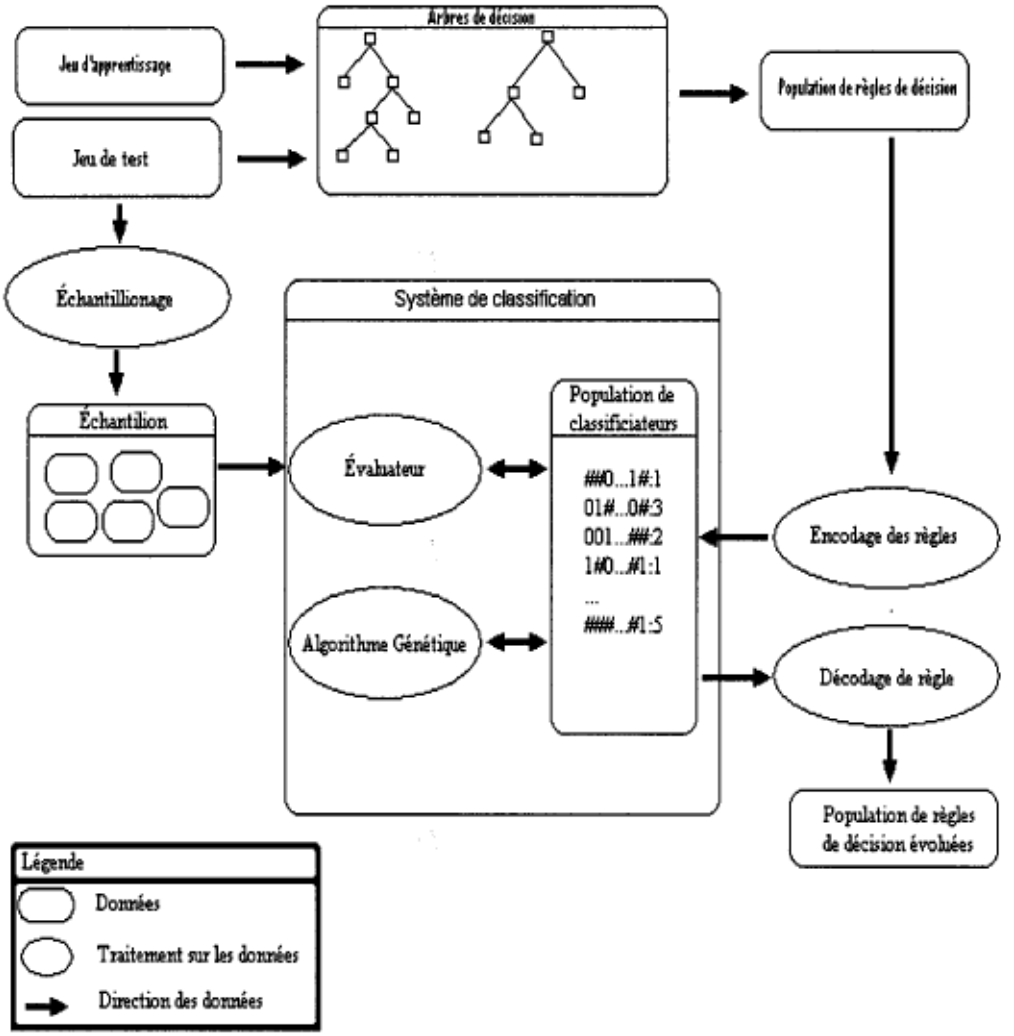

Figure 4.4: Schéma du système de classification.

4.6.1 Fonctionnement du système de classification

Un système de classification est une simulation d'une classification [21]. La population initiale de classificateurs est encodée à partir des règles produites par des arbres de décision. On échantillonne aussi le jeu de tests. On transmet les échantillons à l'évaluateur. Une fois qu'un échantillon a été évalué avec les

classificateurs, l'algorithme génétique est appelé pour produire une nouvelle population à partir de la population actuelle.

Lorsque la simulation se termine, on décode les classificateurs pour obtenir de nouvelles règles.

4.6.2 Paramètre d'une simulation

Pour initier une simulation, on a besoin d'un jeu de tests (voir figure 4.5) et les paramètres de la simulation (voir figure 4.6). Ces paramètres sont reliés au fonctionnement de l'algorithme génétique et contiens l'information sur les classificateurs.

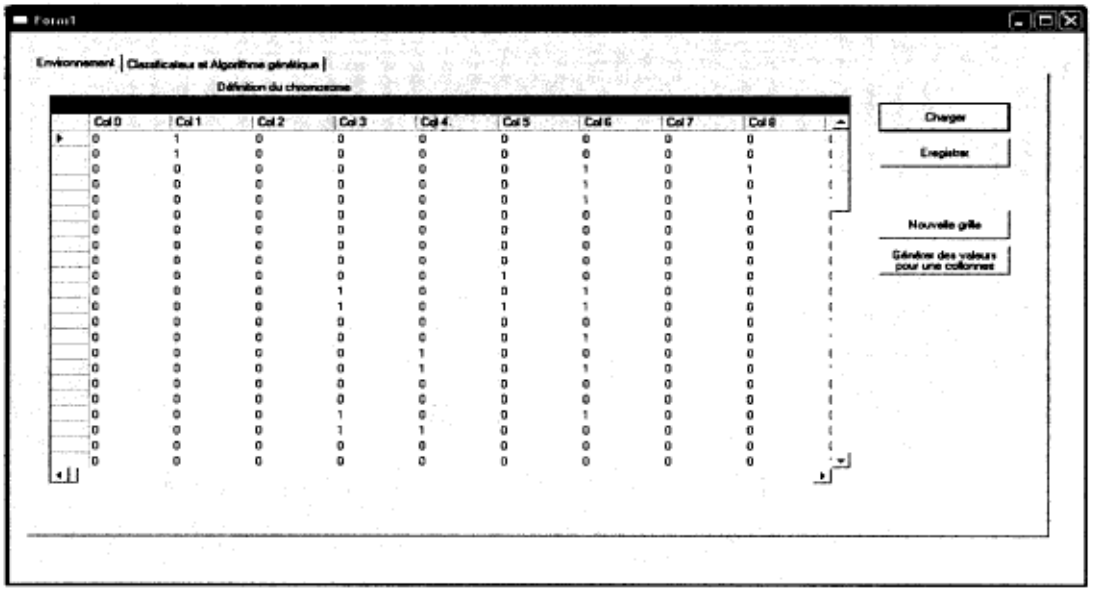

Figure 4.5: Jeu de références de l'évaluation

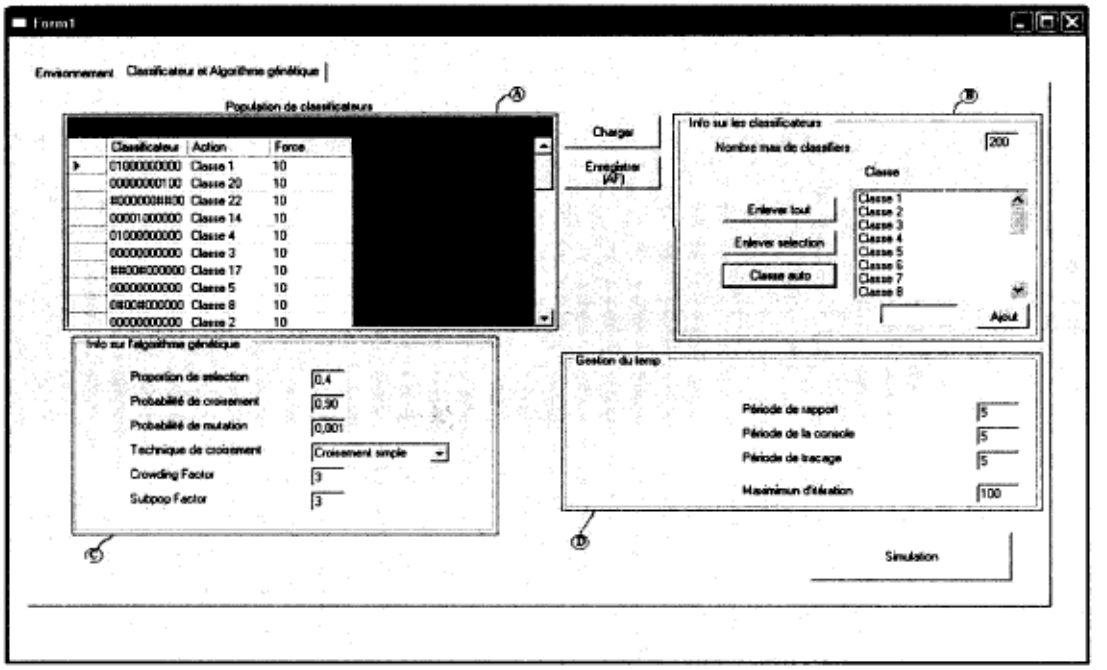

Figure 4.6: Paramètres nécessaires d'une simulation.

Jeu de tests : Le jeu de tests représente l'ensemble de références des instances pour évaluer les classificateurs (Figure 4.5).

Classificateurs : Les classificateurs sont des règles de décision qui sont encodées selon le théorème des schémas (Figure 4.6 fenêtre A).

Paramètre de la population (Figure 4.6 fenêtre B)

Nombre maximal d'individus dans la population : C'est le nombre maximal d'individus de la population.
Les classes : C'est les classes qui sont représentées dans les classificateurs

Paramètre de l'algorithme génétique (Figure 4.6 fenêtre C)

Proportion de la sélection : C'est la proportion d'individus sélectionnés pour reproduire de nouveaux classificateurs à chacune des itérations de l'algorithme génétique.

Probabilité de croisement : C'est la probabilité qu'un couple sélectionné se reproduise.

Probabilité de mutation : C'est la probabilité qu'un gène d'un nouvel individu change pour une autre valeur.

Facteur de repeuplement : C'est le nombre d'essais pour trouver l'individu à remplacer dans la phase de repeuplement.

Facteur de sous-peuplement : C'est le nombre d'individus choisi à chacun des essais pour déterminer le pire.

Paramètre de la gestion du temps (Figure 4.6 fenêtre D)

Période de rapport : Ce nombre représente le nombre d'itérations avant de créer un fichier de classificateur. Exemple : si on 5, on fera un fichier de classificateurs à toutes les 5 itérations.

Période de la console : Ce nombre représente le nombre d'itérations avant de prendre des statistiques sur les classificateurs et le processus d'évolution de l'algorithme génétique. Exemple : si on a 5, on fera un fichier contenant les itérations qui sont des multiples de 5

Périodes de traçage : Ce nombre représente le nombre d'itérations entre la prise de point pour tracer le graphique de la simulation (voir figure 4.7). Ce graphique représente la force de l'individu le plus performant de l'itération et la force moyenne de la population. Exemple : si on a 5, à toutes les 5 itérations, on prendra la force maximale et la force

Maximum d'itération : C'est le nombre d'itérations nécessaires pour compléter la simulation.

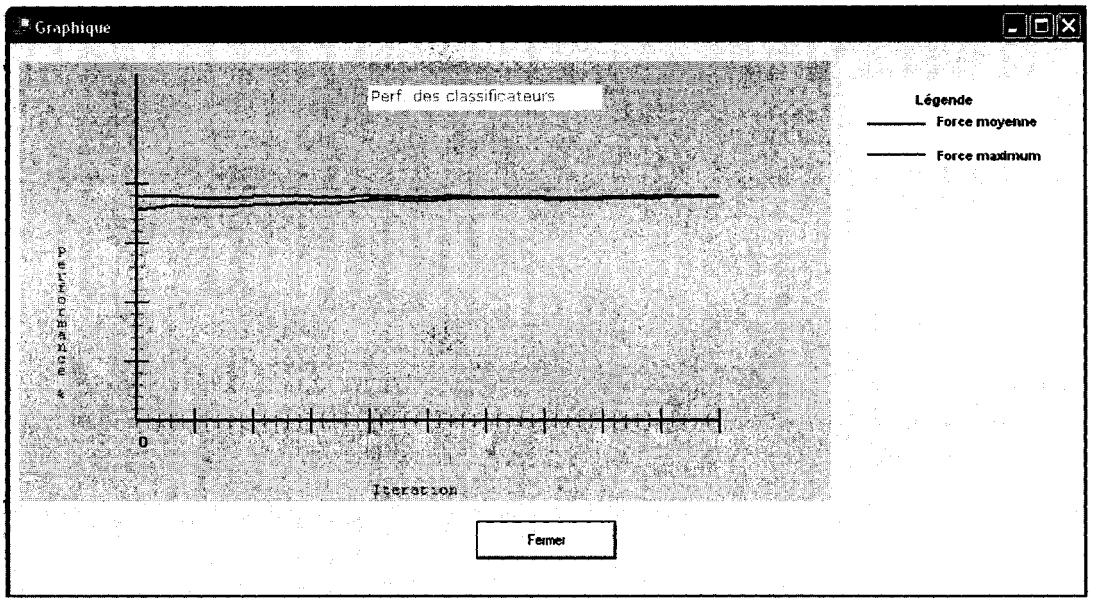

Figure 4.7: Graphique d'une simulation.

Le graphique de la figure 4.7 représente le résultat de l'algorithme génétique au fil du temps. L'algorithme prend des statistiques sur la population de classificateurs à intervalle régulier. Ce graphique est composé de deux courbes. La première courbe représente la force moyenne de la population (en bas). La deuxième courbe représente la force maximale pour chacune des itérations de l'AG. La force moyenne est la moyenne de la performance des individus et la force maximale représente la performance du meilleur individu. La performance des individus est évaluée selon la fonction d'évaluation (voir 4.6.7).

4.6.3 Structure d'un classificateur

Un classificateur est une règle encodée de façon à interagir avec l'algorithme génétique. La structure est composée de:

Un tableau d'entiers : C'est le chromosome du classificateur, il représente le phénotype du lexique de mot. Chacune des positions représente une condition de la règle.

Une classe associée : C'est la classe associée à la règle.

Force : Cette valeur est l'indicateur de performance de la règle, il est le résultat de la fonction d'évaluation des règles. L'indicateur de performance représente la somme de la cohésion interne et de la cohésion externe de l'individu par rapport au jeu d'évaluations (voir 4.6.7).

4.6.4 Fonction d'encodage et de décodage des règles

Une règle de décision a besoin d'être encodée dans un classificateur pour qu'un algorithme génétique soit capable d'analyser la règle. Les gènes du classificateur sont composés par une série de mots contenus dans un lexique de mot. Chacun des gènes représente un mot unique ou un sous série de mots. Pour traduire la règle en classificateur, on examine les conditions de la règle. Pour chacune des conditions où le mot doit être présent, la position du mot dans le classificateur sera 1; dans le cas où le mot doit être absent, la position du mot dans le classificateur sera alors 0; pour les autres mots qui n'appartiennent pas à la règle, tous les gènes définissants les positions des mots restants seront représenter par # (don't care symbol).

Par exemple : On a un lexique composé de 5 mots et la règle de décision suivante :

If (mot=1 AND mot2=0 AND mot5=1) THEN Classe 1.

Si le phénotype du classificateur est défini comme mot1|mot2|mot3|mot4|mot5:Classe, le classificateur sera 10##1 :Classe1

Lorsque la simulation est terminée, on doit faire le processus inverse afin d'obtenir les règles. Pour traduire un classificateur en règle, on regarde les gènes qui sont représentés par 0 ou 1. Pour chacun de ces gènes, on affectera une condition dans la nouvelle règle produite.

4.6.5 Serveur de temps

Comme un système de classification est un système itératif, le serveur de temps gère les itérations de façon à coordonner les classificateurs et l'algorithme génétique dans le temps. Il est aussi responsable de prendre des statistiques sur les classificateurs, sur le couplage de l'algorithme génétique et de créer des fichiers de classificateur à des intervalles de temps fixe.

4.6.6 Échantillonnage

L'échantillonnage consiste à séparer les instances du jeu de tests en un échantillon composé d'instances représentant chacune des classes. On choisit un nombre aléatoire entre le nombre de classes distinctes dans le jeu de tests et le nombre d'exemples contenu dans le jeu de tests, ce nombre représente le nombre d'exemples choisis pour composer l'échantillon. L'échantillon doit être représentatif des classes du jeu de tests, c'est lui qui servira de base de comparaison des règles avec la fonction d'adaptation.

Pour que chacune des classes soit représentée, on regroupe dans un tableau l'index des exemples appartenant à la classe. On effectue les opérations suivantes jusqu'à ce qu'on choisisse tous les exemples pour représenter le jeu de tests avec la fonction d'adaptation.

On choisit aléatoirement une classe *C* dans un tableau des classes disponibles.
On choisit aléatoirement un index d'un exemple dans le jeu de tests dans le tableau des exemples appartenant à la classe *C*.
On enlève la classe *C* du tableau des classes disponibles.
On enlève l'index de l'exemple du jeu de tests dans le tableau des exemples appartenant à la classe *C*.
Si le tableau des classes disponibles est vide, c'est-à-dire qu'on a choisi un exemple de chacune des classes du jeu de tests. On remet les classes présentes dans le jeu de tests dans le tableau des classes disponibles.

4.6.7 Évaluateur

L'évaluateur vérifie si les classificateurs sont acceptables par rapport aux instances d'un échantillon. . C'est la fonction d'adaptation qui fixe la force des classificateurs. On évalue chacun des exemples de l'échantillon avec tous les classificateurs.

Fonction d'adaptation

Rialle [34], nous présente une fonction d'adaptation appliquée à l'analyse terminologique. La fonction d'évaluation d'un classificateur est basée sur le coefficient de Jaccard. Le coefficient de Jaccard est calculé pour un couple (X, Y), X représente un exemple de l'ensemble de l'échantillon d'évaluation et Y représente le classificateur évalué.

$$\text{Aff (X, Y)} = \frac{\text{(Nombre de gènes communs + Nombre de \# * 0.5)}}{\text{Nombre total de gènes du chromosome}} \quad (4.1)$$

Pour les exemples appartenant à la même classe i que le classificateur, on calcule la cohésion interne notée CI. Ce coefficient est la somme pondérée des affinités entre les exemples de l'échantillon et un classificateur.

$$CI = \frac{1}{N(i)} \times \sum_{j \in CCi} \text{Aff(exemple j, classificateur)} \quad (4.2)$$

N(i) représente le nombre d'exemples donc la classe du classificateur correspond à la classe de l'exemple, CCi est l'ensemble des exemples qui appartiennent à la même classe du classificateur et j représente un exemple de l'ensemble CCi.

Pour les exemples n'appartenant pas à la même classe que celle du classificateur, on calcule la cohésion externe notée CE.

$$CE = \frac{1}{NC(i)} \times \sum_{j \in CCe} \text{Aff(exemple j, classificateur)} \quad (4.3)$$

NC(i) représente le nombre d'exemples dont la classe du classificateur est différente à la classe de l'exemple, CCe est l'ensemble des exemples qui n'appartiennent pas à la même classe du classificateur et j représente un exemple de l'ensemble CCe.

c) La force d'un classificateur est la somme de CI et CE

4.6.8 Rôle de l'algorithme génétique

L'algorithme génétique a pour but d'introduire de nouvelles règles dans la population. Il permet surtout d'explorer l'ensemble de recherche de règles significatives. De cette façon, on pourra voir l'adaptation des règles de classification par rapport aux échantillons d'instances à vérifier.

Un algorithme génétique est un processus itératif de simulation, c'est-à-dire qu'à chacune des itérations, la population est évaluée avec la fonction de performance et elle subit des changements avec les opérateurs génétiques.

4.6.9 Implémentation des opérateurs génétiques

Les opérateurs génétiques sont les composants de l'algorithme génétique, ils simulent les comportements de reproduction pour essayer de trouver les meilleures combinaisons de conditions dans l'espace de recherche.

4.6.10 Sélection et couplage des individus

La sélection est le premier opérateur génétique à être appelée, elle sélectionne des couples d'individus pour que l'algorithme génétique produise de nouveaux individus. Le nombre de couples sélectionnés dépend du paramètre de la *proportion d'individus sélectionnés.* Un classificateur ne peut pas être choisi plus qu'une fois par itération de l'algorithme génétique.

On simule une roulette, où chacune des cases est représentée par un classificateur. La case est représentée numériquement par un intervalle de chiffre fixé en fonction de la force du classificateur. On tire un nombre aléatoirement et on choisit le classificateur de la case qui est reliée à ce nombre.

On utilise un tableau de couplage pour stocker l'information sur les individus sélectionnés pour la reproduction.

4.6.11 Hybridation

La phase d'hybridation est la phase de reproduction des chromosomes d'un algorithme génétique. Elle combine la mutation et le croisement.

Mutation

La mutation fait partie du processus de reproduction de la règle, elle est nécessaire pour éviter que les règles convergent vers une optimisation locale. La mutation effectue une modification sur un ou plusieurs gènes du classificateur en fonction de la *probabilité de mutation.* La valeur de ce paramètre est généralement peu élevée.

Croisement

Le croisement des individus fait aussi parti du processus de reproduction des règles. Il alterne des gènes entre les individus pour chacun des couples sélectionnés dans la reproduction de nouveaux individus. Pour chacun des couples sélectionnés, on aura deux nouveaux individus. Le croisement est effectué selon une *probabilité de croisement*, cette probabilité est généralement élevée.

Croisement à un point

Le croisement à un point consiste à séparer les chromosomes parent en deux à partir d'un point de division choisi aléatoirement. On échange les gènes jusqu'au point de division pour créer de nouveaux individus.

Croisement à k-points

Le croisement à k-points sépare les chromosomes en k parties, où k est le nombre de points choisi aléatoirement pour séparer les gènes. On choisi k points de division, chacun des points est choisi aléatoirement parmi les positions des gènes dans un chromosome, on trie en ordre ascendant tous les points choisis. On crée les nouveaux individus en alternant les parties impaires des chromosomes.

4.6.12 Remplacement de la population

Le remplacement de la population consiste à introduire les nouveaux individus issus de la reproduction des couples dans la population active, pour ce faire, les nouveaux individus remplacent les individus sélectionnés pour mourir, cependant un individu qui est sélectionné pour être remplacé peut réussir à survivre, si le maximum d'individus de la population n'est pas atteint. Comme dans la vraie vie, si l'espace vital est réduit, le risque d'épidémie risque d'augmenter et plus de personnes risquent de mourir plus rapidement. C'est un mécanisme d'autorégulation de la population. On tente d'éliminer les règles les moins performantes dans le but de favoriser les règles optimales.

La fonction de peuplement choisit un nombre d'individus sélectionnés pour mourir, ce nombre est un paramètre de l'algorithme génétique qu'on appelle le facteur de peuplement. Chacun des individus désignés pour mourir est choisi aléatoirement. Parmi les individus, on choisit le pire individu, ensuite on le remplace par le nouvel individu

4.6.13 Décodage des classificateurs

Le système produit des fichiers de classificateurs pour faire connaître la population de classificateurs à un intervalle régulier (Figure 4.8). Le but du décodage est de transformer les classificateurs en règles de classification. Pour décoder les classificateurs, on associe chacun des gènes du classificateur à une condition, à l'exception des gènes qui sont représentés par le symbole '#'.

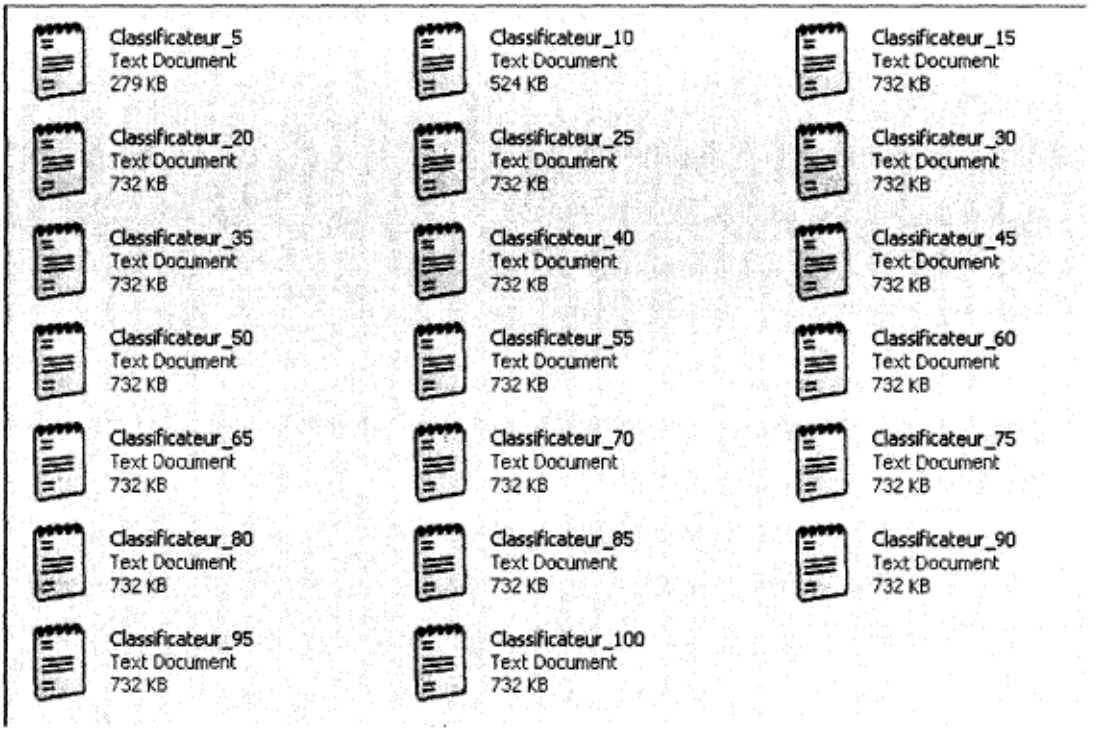

Figure 4.8: Fichiers de sortie.

La figure 4.8 représente l'évolution de la population à travers le temps. Chacun des fichiers contient la population de classificateurs pour une certaine itération de l'AG. Chacun des classificateurs de ces fichiers est représenté par une règle encodée, une classe et sa performance pendant l'itération du fichier.

Pour mieux interpréter les classificateurs, on a besoin de transformer les branches d'un arbre en règles (figure 4.9), de cette façon on pourra les évaluer et voir comment elles ont évolué à travers l'algorithme génétique.

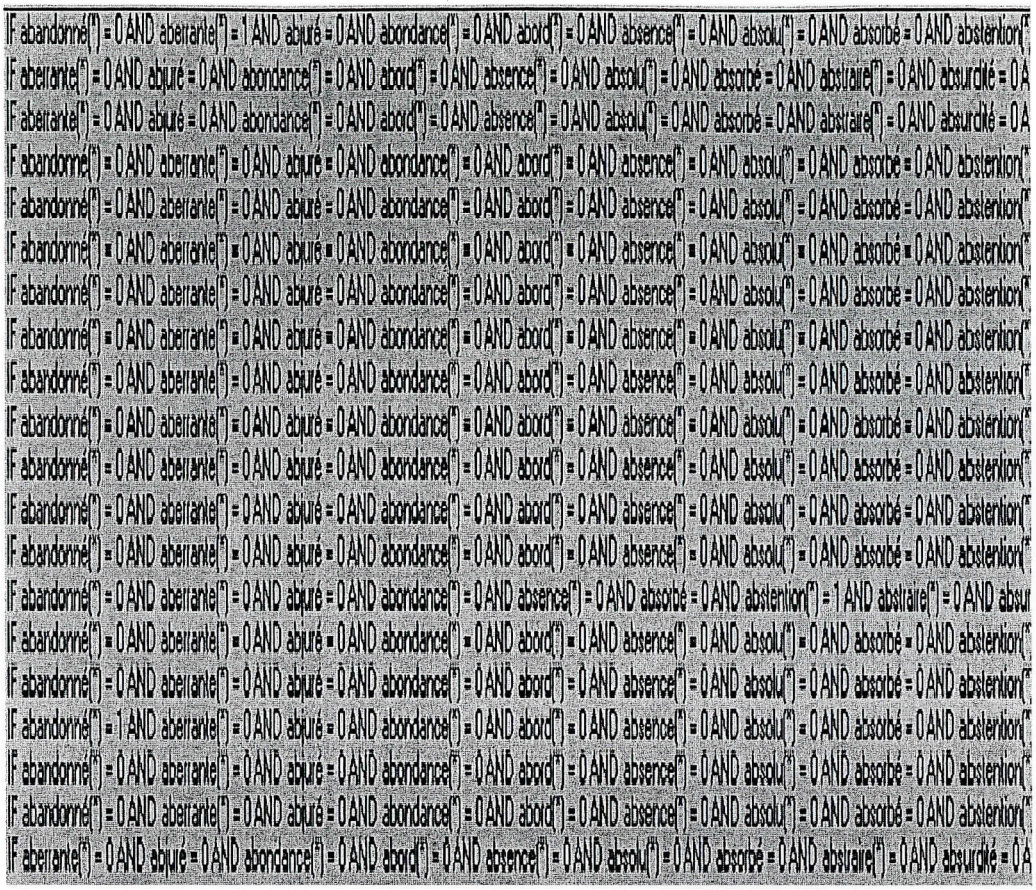

Figure 4.9: Décodage des classificateurs.

La figure 4.9 représente la transformation des classificateurs à des règles de classification. Contrairement aux règles générées par un arbre de décision, l'ordre des conditions est relié à l'ordre du gène dans le phénotype. Elles seront validées par un jeu de tests pour s'assurer de l'intégrité de ces règles.

Une règle représente des conditions à respecter pour associer des segments de texte non classés à une classe quelconque. Pour classer un segment de texte par rapport à une règle, il faut que l'exemple respecte tous les conditions de la règle. Si un attribut est représenté par 0, le mot associé à l'attribut ne doit pas être présent dans le segment, si un attribut est représenté par 1, le mot associé à l'attribut doit obligatoirement présent dans le segment.

CHAPITRE 5 PRÉPARATION DES DONNÉES

5.1 Introduction sur la préparation des données

En apprentissage automatique, pour réussir à faire apprendre à un ordinateur à reconnaître les caractéristiques d'une classe par rapport à la population de données, il faut suivre une démarche structurée.

5.2 CRISP-DM

Selon CRISP-DM (Cross Industry Standard Process for Data Mining) [32], un consensus développé par de grandes entreprises comme SPSS et Daimler-Chrysler, les principales étapes pour le forage de données sont :

La compréhension du domaine
La compréhension des données
La préparation des données
La modélisation
L'évaluation
Le déploiement de la solution

La compréhension du domaine et des données dépend souvent de l'expert du domaine de connaissances. Nous allons particulièrement accorder notre attention, dans ce chapitre, à la préparation des données.

5.2.1 Préparation des données

Souvent, les résultats de l'analyse du forage de données sont reliés directement avec la préparation des données, cette étape doit être faite avec attention. Il ne suffit pas seulement d'aller chercher les données dans une base de données, la plupart du temps, on a une série d'opérations à effectuer pour préparer les données.

La première opération consiste à sélectionner les données, cette étape décide quelles données seront utilisées pour l'analyse. Les critères de sélection relèvent des

objectifs du forage de données, de la qualité des données et des contraintes techniques comme le volume et le type des données.

La deuxième opération consiste à nettoyer les données. Par exemple, certains attributs doivent être discrétisés pour faciliter le processus de classification. On doit aussi s'assurer que le jeu ne contient pas de données incohérentes ou manquantes. Cette opération consiste aussi à s'assurer que chacune des données n'est pas dupliquée.

La troisième opération consiste à construire les données comme les attributs dérivés, où les enregistrements qui sont générés automatiquement à partir de règles primaires.

La quatrième opération consiste à intégrer les données, c'est-à-dire de combiner différentes tables et enregistrement de façon à créer de nouveaux enregistrements.

La dernière opération consiste à formater le jeu d'enregistrements pour les outils utilisés pour le modelage.

La préparation des données demande une compréhension absolue des données et du domaine de connaissances analysé, pour cela, on doit identifier les objectifs à réaliser. Cette étape produira l'ensemble de données qu'on soumettra aux techniques de les évaluer avec les algorithmes de classifications choisis.

5.2.2 Préparation du jeu d'apprentissage

La construction d'un arbre de décision est basée principalement sur un jeu d'apprentissages, on se sert aussi d'un jeu de tests pour valider la classification.

La préparation du jeu d'apprentissages est une étape cruciale dans le processus pour que les règles qu'on veut extraire soient plus précises. On doit avoir un aperçu de chacun des cas possibles. On doit éviter les exemples similaires et qui appartiennent à des classes différentes, ces cas entraîneront des erreurs de classification.

5.2.3 Préparation du jeu de tests

Le jeu de tests est le jeu qui permet de valider la classification. Il a les mêmes caractéristiques que le jeu d'apprentissages, en général, les exemples contenus dans le jeu de tests sont différents du jeu d'apprentissages pour voir la validité des règles de décision produites par l'arbre. Il faut s'assurer que toutes les classes sont présentes dans les deux jeux de façon que le résultat de la classification ne soit pas trop biaisé.

Description d'un exemple

Un exemple est une instance du jeu d'apprentissages, il contient toutes les caractéristiques se rapportant aux différentes classes et une classe qui lui est assignée.

Description d'une classe

Une classe est l'association d'idées qui regroupe des objets. Une classe est composée de plusieurs caractéristiques distinctes, certaines sont plus importantes que les autres. Par exemple, on peut classifier les fleurs selon leur durée de vie, le climat dans lequel elles poussent.

Description d'une caractéristique

Une caractéristique d'une classe est un attribut qui différencie une classe d'une autre, une caractéristique peut convenir à plusieurs classes. Une classe possède plusieurs caractéristiques.

5.3 Qu'est-ce que la classification?

La classification est l'action de regrouper en classe, les objets qui répondent aux caractéristiques de la classe. En apprentissage automatique, on distingue deux grandes classes de méthodologies de classification : la classification supervisée et la classification non supervisée. Cependant, on peut faire un mixte des deux approches pour une classification semi-supervisée.

5.3.1 Classification supervisée

Ce sont des méthodes de classification dites déterministes [20]. Elles ont besoin qu'on connaisse déjà la classe de chacune des instances. Ces méthodes se servent généralement d'un jeu d'apprentissages comme base pour apprendre à reconnaître les concepts qu'on veut lui enseigner. Lorsqu'une nouvelle connaissance est introduite, on se sert des connaissances déjà acquises pour analyser les caractéristiques communes ou différentes de cette nouvelle connaissance. Le passé permet de prédire l'avenir. Si on est capable d'analyser les caractéristiques d'un évènement passé, il est plus facile de prédire un évènement futur.

5.3.2 Classification non supervisée

Ce sont des méthodes de classification dite probabilistes. On ne connaît pas la classification des instances à classifier. À la fin du processus de classification non supervisée, les instances appartiennent à une classe générée par la classification. Une méthode qui est souvent utilisée par les chercheurs en intelligence artificielle est le clustering.

5.4 Classification à partir d'un lexique

On a des lexiques de mots, chacun des lexiques est associé à une classe, les mots peuvent appartenir à une ou plusieurs classes. On veut savoir comment une

classification s'effectue. On veut apprendre quels mots sont les plus importants pour déterminer les caractéristiques nécessaires d'une classe donnée.

On a choisi une représentation binaire pour représenter les exemples, cela ne pose pas de problème pour le traitement avec un arbre de décision et pour les algorithmes génétiques, c'est une représentation naturelle pour les chromosomes. De plus, c'est une représentation qui est facile et rapide à générer.

5.4.1 Établir les attributs

Généralement lorsqu'on parle de classification de texte à partir de mots, le nombre de mots est considérablement élevé, avant d'établir la liste d'attributs à traiter, on doit nécessairement réduire au minimal le nombre de mots.

Dans notre classification de texte, nous avions plusieurs lexiques fabriqués à l'aide de GRAMEXCO [33], chacun des lexiques était associé à une classe. Pour commencer, nous avons constitué une liste de mots en concaténant les mots de chacun des lexiques. Dans la liste de mots, chacun des mots est unique. Dans l'extraction des mots, il faut nettoyer le texte en éliminant les symboles inutiles comme les apostrophes, les virgules ou encore les parenthèses, etc. Pour établir la liste finale de mots, on doit éliminer les mots qu'on utilise couramment dans l'écriture d'un texte comme les articles, les verbes d'état. Cette étape est effectuée avec la supervision d'un expert.

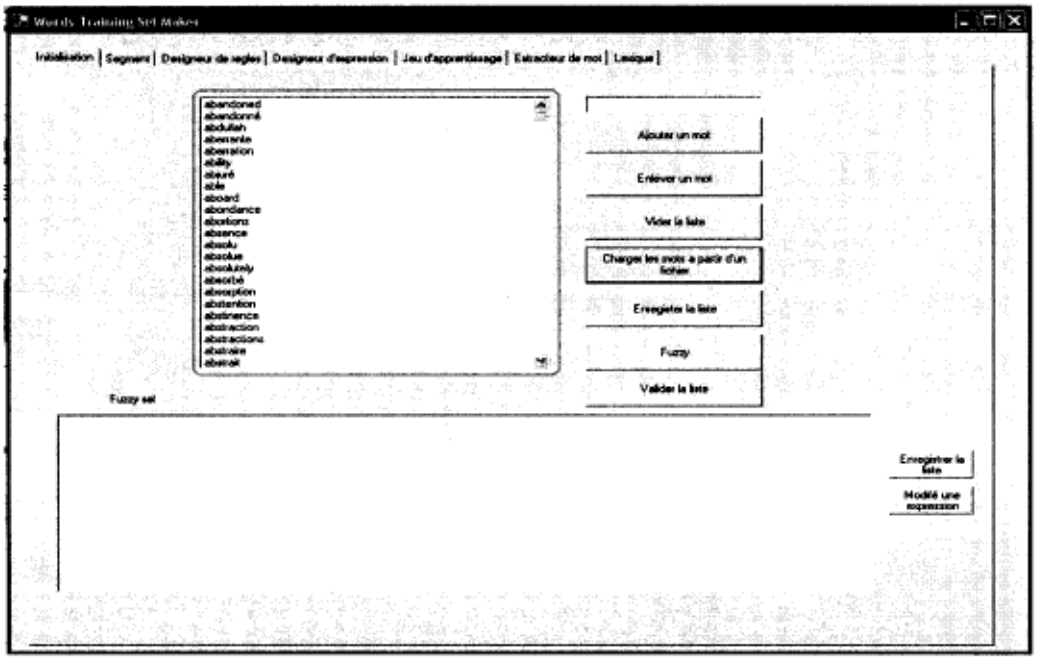

Figure 5.1: Initialisation de la liste de mots.

Lors de la validation de la liste, on établira le nombre de colonnes de la matrice pour fabriquer le jeu de données. Chacun des mots sera représenté par une colonne et la dernière colonne de la ligne représentera la classe associée aux segments de texte (voir figure 5.1).

	Mot 1	Mot 2	Mot 3	....	…	…	Mot n	Classe
Segment 1								
Segment 2								
…								
…								
Segment m								

Table 5-1 : Définition du jeu de données.

Avec les arbres de décision, cette liste servira dans le choix des attributs à évaluer pour la construction de l'arbre de décision. Avec l'algorithme génétique, cette liste sera utilisée dans la définition du chromosome type.

5.4.2 Compression du lexique avec les N-Grammes de caractères

Certains mots dans les lexiques ont une certaine similitude entre eux, certaines fois le mot est le même, la différence est souvent dans le genre et le nombre du mot. Cependant, la similitude peut-être un petit peu plus complexe à déterminer, par exemple, certains mots peuvent avoir une signification proche d'un autre mot, comme 'nation ' et 'nationalisme '.

Pour réduire ces mots, on commence par trier la liste de mots. On utilisera une méthode de recherche de similitude des mots basée sur les n-grammes de caractères. Cette méthode cherche à savoir si deux mots sont proches au niveau lexical.

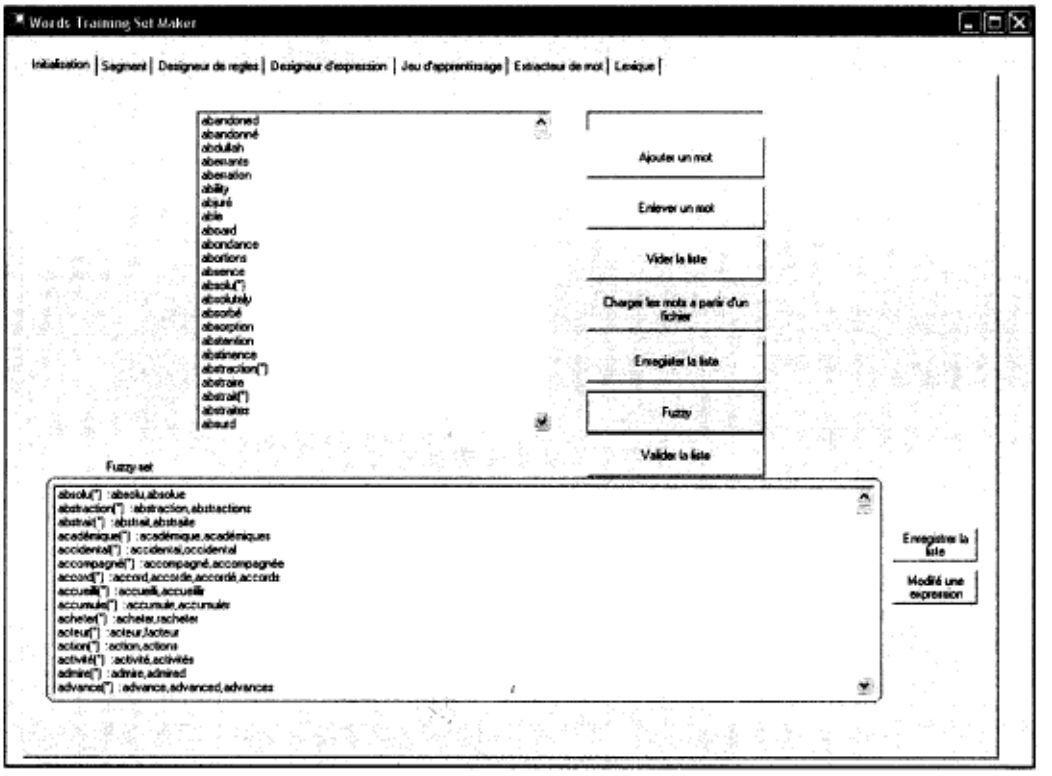

Figure 5.2: Fabrication de la liste de similitude des mots.

Les n-grammes de caractères sont des suites de n caractères. Par exemple, les bi-grammes contiennent toutes les suites de 2 caractères contenues dans un mot, les trigrammes sont des suites de 3 caractères et les quadri-grammes sont une suite de 4 caractères [33].

Avec les mots qui sont similaires, on va établir des sous-groupes de mots qui seront uniquement représentés par un ensemble des mots similaires. Pour ce faire, on commence par trouver les n-grammes communs contenus dans les mots. Ensuite, on divise les n-grammes communs par l'union des ensembles des n-grammes des mots Cette fonction représente le coefficient de Jaccard, elle calcule le nombre de n-grammes communs par rapport à tout l'ensemble des n-grammes contenus dans les deux mots.

(5.1)

N-gramme(mot 1) $\cap$ N-gramme(mot 2) / N-gramme (mot 1) $\cup$ N-gramme (mot 2)

La Figure 5.2 représente la liste de mots et l'ensemble des mots similaires. La liste de mots est composé soit de mots qui n'ont pas de similarité avec les autres ou encore des sous-ensembles de mots représentés par des *. Elle sert aussi de dictionnaire des classes pendant la préparation du jeu de données. Chacun des sous-ensembles de mots est composé de plusieurs mots similaires. Voici quelques exemples de sous-ensembles de mots similaires :

absolu(*): absolu, absolue
abstraction(*): abstraction, abstractions
abstrait(*) : abstrait, abstraite, abstraites
allemand(*) : allemand, allemande, allemandes, allemands
allow(*) : allow, allows
communications(*) :communications, telecommunications, télécommunications
consider(*) : consider, considered
considérée(*) : considérée, considérés
eventual(*) : eventual, eventually
experience(*) : experience, experienced, experiences
expérience(*) : expérience, expériences
explained(*) : explained, unexplained
fonctionnement(*) : fonctionnement, fonctionnent
gouvernement(*) : gouvernement, gouvernemental, gouvernent

impossibilité(*) : impossibilité, possibilité, possibilités
intellectuel(*) : intellectuel, intellectuelle, intellectuelles, intellectuels
production(*) : production, reproduction
question(*) : question, questioned, questions
relative(*) : relative, relatively, relatives
resolution(*) : resolution, solution
respect(*) : respect, respecté, respects
slavophile(*) : slavophile, slavophiles
université(*) :université, university

Ces regroupements permettent donc de réduire la taille du lexique des classes. Ils augmentent l'importance des mots contenus dans ces regroupements par rapport aux autres mots. Pour chacun des mots présents dans un regroupement, on se servira de l'attribut représenté par ce regroupement. Chacun de ses regroupements représentera leurs mots pendant la préparation des données. Les attributs représentent des regroupements de mots. Par exemple : Si on a le regroupement suivant : absolu(*): absolu, absolue

(absolu(*) = 1) signifie que le segment de texte contient un des deux mots.
(absolu(*) = 0) signifie que le segment de texte ne contient aucun des mots.

Pour établir un dictionnaire des classes avec des regroupements intelligents, on doit s'assurer qu'on choisit un seuil acceptable. Ce seuil doit être en mesure de regrouper des mots de même nature terminologiques. Un seuil trop petit risque d'apporter des bruits. Les bruits sont des incohérences dans le sens des mots associés ensemble. Quand on associe automatiquement des mots d'un lexique sur la base des n-grammes, on aura toujours des mots qui sont similaires syntaxiquement et qui ont des différences au niveau sémantique.

Par exemple, le mot 'conservateur ' et le mot 'conservatrice ' ont seulement comme différence le genre du nom, comment peut-on les associer automatiquement. Le mot 'conservateur ' contient 11 bi-grammes et le mot 'conservatrice ' contient 12 bi-

grammes, ils ont 7 bi-grammes communs, soit 'co', 'on', 'nv', 'er', 'rv', 'va' et 'at'. L'union des bi-grammes est la somme des éléments des deux ensembles moins les bi-grammes communs. Si on veut associer ces deux mots, le seuil doit être plus petit que 43.75%. On arrive a ce chiffre en utilisant la formule 4.1, on a 7 bi-grammes communs dans les deux mots vérifiés, le premier mot contient 11 bi-grammes et le deuxième mot contient 12 bi-grammes. L'intersection des mots est représentée par les bi-grammes communs aux deux mots et l'union est représenté par la somme du nombre de bi-grammes communs aux deux mots, on enlève le nombre aussi le nombre de bi-grammes communs de la somme.

$$7/ (11+12-7) = 7/16 == 43,75\%$$

On peut aller plus loin dans l'analyse de correspondances avec des n-grammes de caractères, on peut vérifier si les n-grammes de caractères sont successifs, c'est-à-dire qu'on analyse les liens entre les n-grammes. Reprenons l'exemple précédent avec les mots conservateur et conservatrice. Dans ces mots, on observe qu'il y a 6 liens successifs entre les bi-grammes communs, le premier mot contient 10 liens entre les bi-grammes du mot et le deuxième mot contient 11 liens entre les bi-grammes du mot. Pour associer ces deux mots de même nature dans leurs sens, on aura besoin d'un seuil de 40 %. Cette méthode ne compte pas seulement des n-grammes communs dans les mots, mais elle tient compte aussi de leurs positions dans les mots.

$$6 / (10 + 11- 6) = 6/15 = 40\%$$

Malgré le fait que les mots soient proches au niveau syntaxique, ce n'est pas suffisant pour dire qu'ils sont similaires. Pour affirmer qu'ils sont similaires, ils doivent aussi avoir un sens commun. Il faut donc analyser la liste de mots avec un expert.

CHAPITRE 6 EXPÉRIMENTATION ET RÉSULTATS

6.1 Fabrication du jeu de données

Les données doivent souvent être prétraitées pour être préparées à la phase d'évaluation. Il faut qu'on adapte la présentation des données pour les rendre conformes aux algorithmes qu'on veut tester. Dans notre problème de classification, on détermine si la présence ou l'absence d'un mot peut nous aider à classifier des segments de textes.

Les segments de texte ont été générés à partir de GRAMEXCO [33], chacun des segments est associé à une classe. Une classe n'est pas exclusive à un segment.

La fabrication du jeu de données est faite à partir de l'analyse de segments de texte. Chacun des jeux est composé de segments de texte transformé en instances binaires. Chacun des attributs des instances représente soit un mot ou soit un regroupement de mots. Pour chacun des attributs représentant le dictionnaire des mots, on regarde si le mot est présent à l'intérieur du segment de texte.

Jeux de données	***Description***	***Nb Mots***	***Nb regroupement de mot***
Jeu A	Jeu sans transformation	7839	0
Jeu B	Quadri-Grammes seuil 85%	7598	232
Jeu D	Quadri-Grammes seuil 75%	7098	670
Jeu C	Liens Bi-Grammes seuil 85%	7476	344
Jeu E	Liens Bi-Grammes seuil 75%	6854	869

Table 6-1 : Jeux de données.

6.1.1 Information sur les classes

Les mots appartiennent à une ou plusieurs classes, ils sont soit en français ou en anglais dépendant des classes. L'étude du lexique peut nous apporter des connaissances de base sur les mots présents ou non dans les classes. Chacun des jeux de la table 6-1 contient des segments de texte encodé, c'est-à-dire que chacun des segments est représenté comme un tableau de dimension n, où n représente le nombre de mots contenu dans le jeu. Chacune des positions du tableau représente un

mot, si le mot est présent dans le segment, la position du mot sera représentée par 1, sinon elle sera représentée par 0. Les classes contenues dans chacun des jeux sont caractérisées par les mots de leurs lexiques respectifs.

Classe	***Langage***	***Jeu A***	***Jeu B***	***Jeu C***	***Jeu D***	***Jeu E***
Classe 1	Français	113	112	112	112	110
Classe 2	Français	61	61	61	61	61
Classe 3	Français	280	277	273	273	269
Classe 4	Français	270	265	258	258	253
Classe 5	Français	36	36	36	36	36
Classe 6	Français	61	61	61	61	60
Classe 7	Français	120	115	114	114	114
Classe 8	Français	61	61	61	61	61
Classe 9	Français	64	64	63	63	63
Classe 10	Français	107	106	106	107	106
Classe 11	Français	38	38	38	38	38
Classe 12	Français	70	70	70	70	70
Classe 13	Français	40	40	40	40	40
Classe 14	Français	65	65	64	64	40
Classe 15	Français	81	80	79	79	79
Classe 16	Français	285	280	270	270	269
Classe 17	Français	69	68	68	68	68
Classe 18	Français	72	72	72	72	72
Classe 19	Français	103	103	100	100	100
Classe 20	Français/Anglais	188	186	185	185	184
Classe 21	Anglais	162	160	159	159	158
Classe 22	Anglais	133	133	131	131	129
Classe 23	Anglais	372	370	367	367	363
Classe 24	Anglais	172	169	167	167	167
Classe 25	Anglais	387	386	384	384	383
Classe 26	Anglais	378	376	269	369	364
Classe 27	Anglais	467	466	456	456	453
Classe 28	Anglais	435	432	428	428	426
Classe 29	Anglais	360	358	351	351	351
Classe 30	Anglais	82	81	79	79	75
Classe 31	Anglais	24	24	23	23	23
Classe 32	Français	277	274	268	268	266
Classe 33	Français	146	146	145	145	144
Classe 34	Français	264	260	256	256	256
Classe 35	Français	255	250	243	243	242
Classe 36	Français	262	255	251	251	249
Classe 37	Français	266	256	253	253	250
Classe 38	Français	249	241	236	236	235
Classe 39	Français	259	254	252	252	251
Classe 40	Français	278	275	269	269	266
Classe 41	Français	298	293	282	282	278
Classe 42	Français	284	281	274	274	270
Classe 43	Français	291	285	282	282	281

Classe 44	Français	277	269	264	265	263
Classe 45	Français	301	300	296	296	294
Classe 46	Français	216	215	213	213	210
Classe 47	Français	171	170	166	166	164
Classe 48	Français	505	497	484	484	481
Classe 49	Français	353	351	344	344	343
Classe 50	Anglais	182	181	175	175	174
Classe 51	Anglais	279	279	277	277	277
Classe 52	Anglais	446	446	442	442	439
Classe 53	Français	362	358	346	346	344
Classe 54	Français	267	267	261	261	259
Classe 55	Anglais	870	868	852	852	835
Classe 56	Français	352	350	340	340	335

Table 6-2: Nombre de mots présents dans les lexiques des classes.

La table 6.2 est la liste des classes, ce tableau contient le nombre de mots pour chacune des classes. Le jeu A est composé de tous les mots du dictionnaire, sans aucun regroupement. Les jeux B et C représentent les données traitées avec les quadri-grammes et les jeux D et E sont traitées avec les bi-grammes successifs. Chacun des jeux contient des segments de textes associés à une classe en particulier. Une classe est définie par son lexique de mots.

À partir du lexique des classes, on peut acquérir des connaissances sur le jeu de données primaire. Le lexique des classes nous apprend la dispersion des mots à travers les classes, certaines classes contiennent plus de mots que d'autres. L'utilisation des n-grammes de caractères pour regrouper les mots similaires, c'est-à-dire qu'ils se ressemblent syntaxiquement et qu'ils ont un sens commun, permet de réduire raisonnablement les attributs à vérifier lors de la classification. Les quadri-grammes de caractères regroupent assez bien les mots similaires. Mais l'étude entre les liens entre les bi-grammes de caractères permet de regrouper les mots similaires plus efficacement que les quadri-grammes de caractères. La réduction du dictionnaire de mots équivaut à 10% pour les quadri-grammes et à 13% pour les bi-grammes successifs à un seuil de similarité de 75%. L'annexe A nous présente des exemples de regroupement de mots effectué à l'aide de n-grammes.

La distribution du nombre de mots n'est pas la même pour chacune des classes. L'utilisation de méthodes de réduction n'affecte pas vraiment les petites classes,

mais elle affecte les classes les plus populeuses. Par exemple, la classe 55 a passé de 870 mots dans le dictionnaire original à 835 mots avec la méthode des bi-grammes successifs.

On peut constater avec la table 6.2, que la distribution des mots entre les classes est à peu près la même entre le jeu C et D. Pour un seuil plus élevé avec l'étude entre les liens entre les n-grammes de caractères, on a le même résultat que celui des quadri-grammes de caractères avec un seuil plus petit. Même si la réduction des mots à vérifier pour la classification est utile pour accélérer le traitement, elle a certains inconvénients. Certains regroupements contiennent des mots qui sont similaires au niveau syntaxique, mais qu'ils n'ont aucun sens en commun, pour vous donner un exemple, les mots 'acteur ' et 'facteur ' se ressemblent seulement au niveau syntaxique.

6.2 Observation sur les arbres de décision

Les schémas des arbres de décision fournissent généralement une vue sur les mots importants pour pouvoir distinguer les similitudes et les différences entre les classes. Plus le mot est proche de la racine, plus le mot est important dans la classification. Les figures 7.1 à 7.10 (voir l'annexe B) sont le résultat de la classification des jeux A à E.

Jeu	C4.5 Gain			
	nb feuilles/règles	nb feuilles sur apprentissage	nb nœud dans l'arbre	plus que 2 règles par classes
A	67	11	133	9
B	64	8	127	8
C	64	8	127	8
D	64	8	127	8
E	64	8	127	8

Table 6-3 : Résultats des jeux selon l'algorithme C4.5.

Jeu	CART(Gini)			
	nb feuilles/règles	nb feuilles sur apprentissage	nb nœud dans l'arbre	plus que 2 règles par classes
A	56	0	111	0
B	56	0	111	0
C	56	0	111	0
D	56	0	111	0
E	56	0	111	0

Table 6-4: Résultats des jeux selon l'algorithme CART.

Les algorithmes de construction d'arbres de décisions C4.5 et CART permettent de générer des règles de classification. Ils permettent de schématiser les règles de manière visuelle. Les nœuds intermédiaires représentent un ou plusieurs mots, les nœuds terminaux représentent les résultats de la classification et les branches représentent leurs présences ou leurs absences dans le jeu d'apprentissages.

L'algorithme C4.5 génère les plus petits arbres de décision possible, c'est-à-dire qu'il distribue les exemples de façon équitable. Il permet de faire une bonne répartition des classes. Malgré les règles qui sont en sur apprentissage, c'est-à-dire qu'une certaine règle classe seulement une instance. Les figures 7.1, 7.3, 7.5, 7.7 et 7.9 ont été générées par l'algorithme C4.5

L'algorithme CART réparti les classes en niveaux, plus la classe est proche de la racine, plus le nombre de segments de cette classe est élevé. C'est un phénomène de régression. Les nœuds intermédiaires représentent généralement des mots qui sont uniquement dans les classes associées au nœud. Les arbres générés par CART sont profonds. Les règles produites par les arbres par l'algorithme CART regroupent les classes de tous les exemples des jeux d'apprentissage. Chacune des règles représente une seule classe et chacune des classes est représentée par une seule règle. Les règles contiennent donc les éléments communs dans tous les segments classés par une règle. Les figures 7.2, 7.4, 7.6, 7.8 et 7.10 ont été générées par l'algorithme CART

6.2.1 Comprendre la classification avec les règles extraites d'un arbre de décision

Les règles de décision sont extraites à partir d'un arbre de décision. Elles nous dressent un schéma sur les mots utilisés ou non pendant la classification. Les tables 6.5 et 6.6 représentent les nombres de mots présent (1) ou absents(0) dans les règles selon les classes. L'addition du nombre des mots présents et des mots absents représente le niveau du nœud de la classe, par exemple, dans la règle qui représente la classe 1 comporte 2 mots présents et 5 mots absents, alors le nœud qui représente la classe 1 dans l'arbre est situé au niveau 7. L'annexe C présente des exemples de règles de classification généré à l'aide d'un arbre de décision.

6.2.2 Observation sur les règles de décision générées par C4.5

		Jeu A		Jeu B		Jeu C		Jeu D		Jeu E	
Classe	***Langage***	***1***	***0***	***1***	***0***	***1***	***0***	***1***	***0***	***1***	***0***
Classe 1	Français	2	5	2	5	2	5	2	5	4	3
Classe 2	Français	3	4	3	3	2	6	2	6	2	6
Classe 3	Français	5,2	2,5	0,5	9,2	1,5	8,2	1,5	8,2	1,5	8,2
Classe 4	Français	4,1	3,8	2,4	5,3	1,4	7,3	1,4	7,3	1,3	7,4
Classe 5	Français	1	5	1	6	2	5	2	5	2	5
Classe 6	Français	1,1	6,5	1	5	1	5	1	5	1	4
Classe 7	Français	5	1	3	3	2	4	2	4	2	4
Classe 8	Français	3	4	3	4	2	5	2	5	2	5
Classe 9	Français	2	5	2	5	1	7	1	7	1	7
Classe 10	Français	2	5	3	3	3	3	3	3	3	3
Classe 11	Français	0	8	1	7	0	9	0	9	0	9
Classe 12	Français	2	3	2	3	2	3	2	3	2	3
Classe 13	Français	2	5	2	6	1	7	1	7	1	7
Classe 14	Français	1	6	1	7	2	5	2	5	2	5
Classe 15	Français	2	6	3	4	2	6	2	6	2	6
Classe 16	Français	2,4	7,2	1	6	2	5	2	5	2	5
Classe 17	Français	1	6	2	5	2	5	2	5	2	5
Classe 18	Français	2	4	2	5	3	4	3	4	3	4
Classe 19	Français	2	5	2	5	3	4	2	5	2	5
Classe 20	Français/Anglais	3	2	3	2	3	2	3	2	2	3
Classe 21	Anglais	2	3	2	3	2	3	2	3	2	3
Classe 22	Anglais	4	1	4	1	4	1	4	1	3	2
Classe 23	Anglais	3,2	5,4	2,3	4,5	2,3	4,4	2,3	4,4	3,4	4,1
Classe 24	Anglais	4	1	4	1	4	1	4	1	3	2
Classe 25	Anglais	3	2	3	2	3	2	3	2	3	2
Classe 26	Anglais	2	3	2	3	2		2	3	3,4	2,1
Classe 27	Anglais	2	3	2	3		3	2	3	3	2
Classe 28	Anglais	3	1	3	1	3	1	3	1	4	1
Classe 29	Anglais	3,4	2,1	3,4	2,1	3,4	2,1	3,4	2,1	2	3

Classe 30	Anglais	2	6	2	6	2	5	2	5	2	5
Classe 31	Anglais	1	7	1	7	1	6	1	6	1	6
Classe 32	Français	3	3	3	3	3	3	3	3	3	4
Classe 33	Français	3	4	3	4	3	4	2	4	2	5
Classe 34	Français	2	4	2	4	1	6	3	2	1	6
Classe 35	Français	2	5	1	6	2	4	1	6	2	4
Classe 36	Français	3	2	2	4	3	3	3	3	3	3
Classe 37	Français	2	4	3	3	3	3	3	3	3	3
Classe 38	Français	2	4	2	4	2	4	2	4	2	4
Classe 39	Français	3,3	3,3	3	2	3	2	3	2	3	2
Classe 40	Français	2	5	1,3	8,3	3,4	4,2	3,4	4,2	3,4	4,2
Classe 41	Français	1	6	1,4	6,2	1,3	6,3	1,3	6,3	1,3	6,3
Classe 42	Français	4	2	4	2	4	1	4	1	4	1
Classe 43	Français	3,3,3	4,4, 3	3	2	4	2	3	2	3	2
Classe 44	Français	1,3	7,3	2,2	4,5	1,2	6,4	1,2	6,4	1,2	6,4
Classe 45	Français	3,3	4,3	2,3	4,4	2,3	5,4	2,3	5,4	2,4	5,3
Classe 46	Français	3	4	3	4	3	4	3	4	4	3
Classe 47	Français	4	3	4	3	4	3	4	3	3	4
Classe 48	Français	2,4	5,2	3	4	2	5	2	5	3	3
Classe 49	Français	3	3	2	4	3	3	3	3	3	3
Classe 50	Anglais	2	6	2	6	2	5	2	5	2	5
Classe 51	Anglais	3	2	3	2	3	2	3	2	3	2
Classe 52	Anglais	3	2	3	2	3	2	3	2	4	1
Classe 53	Français	3	4	3	4	2	5	3	4	3	4
Classe 54	Français	2	5	2	5	3	4	3	4	3	4
Classe 55	Anglais	5	0	5	0	5	0	5	0	5	0
Classe 56	Français	4	3	4	3	4	3	4	3	2	5

Table 6-5 : Résultats des arbres de décision C4.5 selon les jeux.

La table 6.5 contient l'information sur les mots présents ou non des règles par rapport à leurs classes. Ces règles ont été extraites d'arbres générés par l'algorithme C4.5. On remarque que le nombre de mots présents dans une condition est proportionnel aux nombres de mots absents dans cette même condition. Chacune des classes est associée à un lexique de mots différents. Certains mots peuvent être dans plusieurs classes.

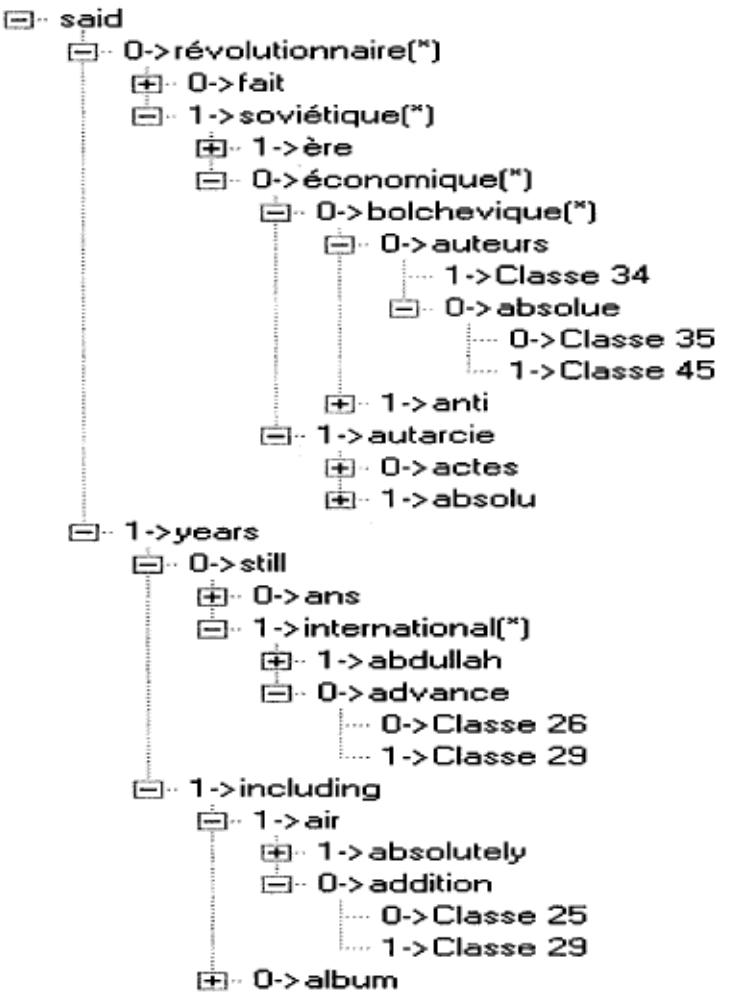

Figure 6.1:Partie d'un arbre de décision C4.5.

Voici des exemples de règles extraits à partir d'un arbre de décision C4.5 de la figure 6.1 :

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 0 AND bolchevique(*) = 0 AND auteurs = 0 AND absolue = 0 THEN Classe 35

IF said = 1 AND years = 1 AND including = 0 AND album = 0 AND able = 0 THEN Classe 27

Chacune de règles précédentes contient l'information nécessaire pour identifier la bonne classification des segments de texte dans le jeu de tests. Dans certains cas, pour une classe, on aura plus qu'une règle.

IF said = 1 AND years = 0 AND still = 1 AND international(*) = 0 AND advance = 1 THEN Classe 29

IF said = 1 AND years = 1 AND including = 1 AND air = 0 AND addition = 1 THEN Classe 29

L'algorithme C4.5 construit un arbre de décision le plus petit possible, l'algorithme choisit d'abord les mots les plus discriminants, plus le mot est proche de la racine de l'arbre, plus le mot est utilisé dans les segments de texte du jeu d'apprentissages.

Dans les arbres générés par l'algorithme C4.5, on observe que le mot 'said' est utilisé comme racine dans tous les arbres. Ce mot sépare la plupart des classes anglaises et les classes françaises. Dans les faits, ce mot est le plus fréquent dans le jeu d'apprentissages. Si on se reporte à sa définition, ce mot anglais est utilisé pour rapporter des paroles, c'est normal que ce mot se retrouve comme racine des arbres. Ce mot permet donc d'effectuer une bonne séparation entre les segments anglais et les segments français, cependant ce n'est pas suffisant pour englober tous les segments anglais. Quand le mot 'said' est présent dans un segment de texte, les mots choisis pour les nœuds intermédiaires sont anglais et quand ce mot est absent, la majorité des mots sont français.

Malgré le fait que la classification automatique des segments de texte soit efficace, certaines classes sont exprimées avec plusieurs règles. Avec le jeu original, on obtient 11 feuilles qui classent un seul segment de texte et 9 classes qui sont représentées avec plusieurs règles. Pour vous montrer un exemple à partir de la table 6.5, la classe 43 qui contient trois segments de texte est classifiée sur 3 feuilles. Ce phénomène est appelé le sur apprentissage, c'est-à-dire que chacun des segments de texte est associé une seule feuille.

Les n-grammes favorisent le regroupement des mots similaires, cette action permet d'améliorer les règles, tout en diminuant le nombre d'attributs à traiter lors de la construction de l'arbre. De plus, ils permettent de diminuer le sur apprentissage.

Cependant, l'utilisation de n-grammes de caractères peut introduire un certain bruit dans le jeu d'apprentissages.

6.2.3 Observation sur les règles de décision générées par CART

		Jeu A		Jeu B		Jeu C		Jeu D		Jeu E	
Classe	***Langage***	***1***	***0***	***1***	***0***	***1***	***0***	***1***	***0***	***1***	***0***
Classe 1	Français	2	34	2	34	1	38	2	34	1	38
Classe 2	Français	2	28	2	28	2	28	2	28	2	28
Classe 3	Français	1	5	1	5	1	6	1	6	1	6
Classe 4	Français	1	7	1	7	1	8	1	8	1	8
Classe 5	Français	1	35	1	35	1	31	1	35	1	32
Classe 6	Français	1	38	1	38	1	40	1	38	1	40
Classe 7	Français	2	27	2	27	2	27	2	27	2	27
Classe 8	Français	1	34	1	34	1	34	1	34	1	35
Classe 9	Français	1	39	1	39	1	41	1	39	1	41
Classe 10	Français	2	30	2	30	2	29	2	30	2	29
Classe 11	Français	1	40	1	40	2	30	1	40	3	31
Classe 12	Français	1	37	1	37	1	39	1	37	1	39
Classe 13	Français	1	32	1	32	1	32	1	32	1	33
Classe 14	Français	1	31	1	31	1	31	1	31	1	31
Classe 15	Français	1	33	1	33	1	33	1	33	1	34
Classe 16	Français	1	11	1	11	1	10	1	11	1	10
Classe 17	Français	1	35	1	35	1	28	1	35	1	28
Classe 18	Français	2	31	2	31	2	26	2	31	2	26
Classe 19	Français	2	33	2	33	1	37	2	33	1	37
Classe 20	Français/Anglais	1	28	1	28	1	32	1	28	1	31
Classe 21	Anglais	1	27	1	27	2	27	1	27	2	27
Classe 22	Anglais	1	23	1	23	1	23	1	23	1	23
Classe 23	Anglais	1	3	1	3	1	4	1	4	1	4
Classe 24	Anglais	1	29	1	29	1	29	1	29	1	29
Classe 25	Anglais	1	10	1	10	1	1	1	1	1	1
Classe 26	Anglais	1	2	1	2	1	3	1	3	1	3
Classe 27	Anglais	1	6	1	6	1	7	1	7	2	7
Classe 28	Anglais	1	1	1	1	1	2	1	2	1	2
Classe 29	Anglais	1	4	1	4	1	5	1	5	1	5
Classe 30	Anglais	1	30	1	30	2	31	1	30	1	32
Classe 31	Anglais	0	41	0	41	0	42	0	41	0	42
Classe 32	Français	1	9	1	9	1	14	1	10	1	14
Classe 33	Français	2	32	2	32	1	35	2	32	1	36
Classe 34	Français	1	21	1	18	1	18	1	18	1	18
Classe 35	Français	1	17	1	17	1	17	1	17	1	17
Classe 36	Français	1	15	1	15	1	15	1	15	1	15
Classe 37	Français	1	20	1	20	1	20	1	20	1	20
Classe 38	Français	1	8	1	8	1	9	1	9	1	9
Classe 39	Français	1	18	1	19	1	19	1	19	1	19
Classe 40	Français	1	19	1	21	1	21	1	21	1	21
Classe 41	Français	1	12	1	12	1	11	1	12	1	11

Classe 42	Français	1	14	1	14	1	13	1	14	1	13
Classe 43	Français	1	13	1	13	1	12	1	13	1	12
Classe 44	Français	1	16	1	16	1	16	1	16	1	16
Classe 45	Français	1	23	1	23	1	23	1	23	1	23
Classe 46	Français	2	26	2	26	2	26	2	26	2	26
Classe 47	Français	1	28	1	28	1	28	1	28	1	28
Classe 48	Français	1	0	1	0	1	0	1	0	1	0
Classe 49	Français	2	22	2	22	2	22	2	22	2	22
Classe 50	Anglais	2	24	2	24	2	24	2	24	2	24
Classe 51	Anglais	1	36	1	36	1	36	1	36	2	30
Classe 52	Anglais	2	25	2	25	1	28	2	25	1	28
Classe 53	Français	2	27	2	27	2	27	2	27	2	27
Classe 54	Français	2	29	2	29	2	25	2	29	2	25
Classe 55	Anglais	1	25	1	25	1	25	1	25	1	25
Classe 56	Français	2	26	2	26	2	30	2	26	2	30

Table 6-6: Résultats des arbres de décision CART selon les jeux.

La table 6.6 contient l'information sur les mots présents ou non des règles par rapport à leurs classes. Ces règles ont été extraites d'arbres générés par l'algorithme CART. On remarque que le nombre de mots présents dans une condition est nettement inférieur aux nombres de mots absents dans cette même condition. Pour chacune des règles, il suffit d'un ou deux mots présents pour classifier un segment de texte.

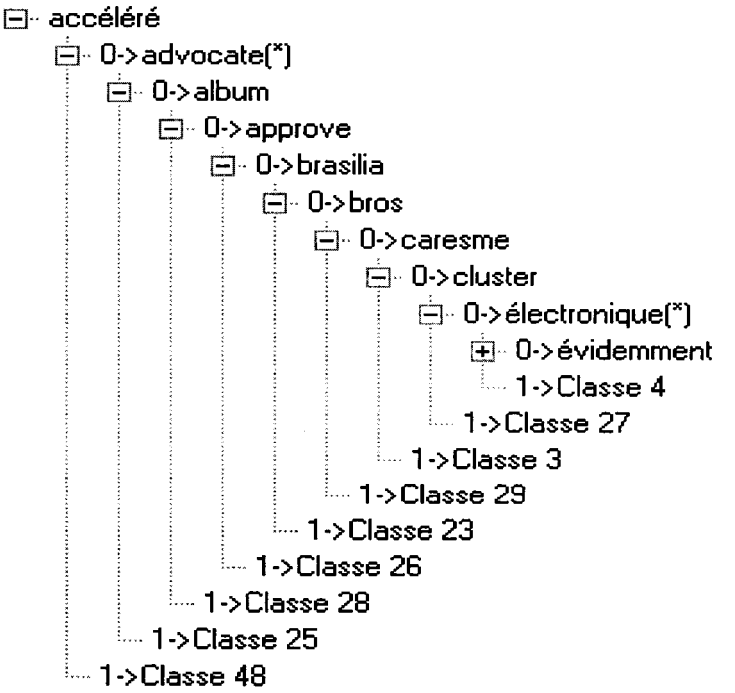

Figure 6.2:Partie d'un arbre de décision CART.

Voici des exemples de règles extraits à partir d'un arbre de décision de la figure 6.2 :

IF accéléré = 1 THEN Classe 48

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND approve = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster = 1 THEN Classe 27

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND approve = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster = 0 AND électronique(*) = 1 THEN Classe 4

L'algorithme CART construit un arbre de décision afin de favoriser les classes les plus populaires, plus le niveau d'un nœud d'une classe est petit, plus le nombre d'exemples de cette classe est élevé par rapport aux autres classes.

Dans la plupart des règles, on a seulement besoin d'un ou deux mots présents pour séparer une classe par rapport à l'ensemble des autres classes.

Tous les arbres ont une racine commune, ils utilisent le mot 'accéléré ' comme racine. CART commence par classer les classes les plus fréquentes dans le jeu d'apprentissages et prend le premier mot dans la liste des mots.

Chacune des classes est exprimée avec seulement une seule règle, ce phénomène permet donc d'analyser les éléments communs de chacune des classes. Si on utilise les règles extrapolées générées avec CART, on aura tous les éléments communs entre les segments d'une classe.

6.3 Observation avec un algorithme génétique

Pour la réussite de la classification avec un algorithme génétique, la fonction d'évaluation est capitale, elle doit être spécifique au problème de classification. La fonction d'évaluation est décrite au chapitre 4 à la section 4.6.7.

Décodage des classificateurs

Le décodage des classificateurs consiste à transformer les classificateurs en règles. Le but de l'algorithme génétique est de produire des règles de classification

acceptables, c'est-à-dire que les règles extraites des classificateurs sont conformes au jeu de tests.

6.3.1 Paramètre de la simulation d'un algorithme génétique

Pour fonctionner, un algorithme génétique dépend de plusieurs paramètres. Voici les paramètres que nous avons utilisés pour tester l'algorithme génétique.

Nombre maximum d'itérations : 200
Nombre maximum de classificateurs dans la population : 1000
Proportion de la population sélectionnée pour l'hybridation : 50%
Méthode utilisée pour le croisement : Croisement à k-points
Probabilité de croisement : 90%
Probabilité de mutation : 5%

Environnement :

Environnement	*Nb mots*
Environnement A	7839
Environnement B	7598
Environnement C	7476
Environnement D	7098
Environnement E	6854

Table 6-7: Tableau des environnements.

Dans la table 6.7, les environnements représentent le jeu utilisé pendant l'évaluation de l'algorithme génétique, chacun des environnements contiennent les mêmes informations que les jeux utilisés pour la construction des arbres de décision. Les lexiques des classes sont utilisés pour évaluer les classificateurs. Ils servent de base de référence sur les classes. Le nombre de gènes utilisés pour un environnement est relié au nombre de mots contenu dans le dictionnaire de mots selon sa méthode de réduction de mots.

Mot 1	Mot 2	...	Mot n

Figure 6.3: Phénotype utilisé pour chacun des environnements.

Le phénotype décrit à la figure 6.3 est la représentation de la signification des segments de textes, chacun des mots du dictionnaire correspond à un gène. L'ordre des mots dans le phénotype est alphabétique.

Classificateurs

Un classificateur représente une règle encodée. Chacun des gènes d'un classificateur représente soit un mot ou un regroupement de mots. Ils sont évalués à partir d'un environnement de test.

Population initiale

Pour comparer les arbres de décision et les algorithmes génétiques, nous avons voulu analyser le comportement d'une population initiale de règles extraites à partir d'arbres de décision par rapport à une population de règles générées aléatoirement.

Populations initiales de règles aléatoires

La population de règles est composée essentiellement de règles aléatoires, chacun des gènes était choisi au hasard selon 3 symboles (0,1,#). Les classificateurs générés ne classent pas vraiment le jeu de tests, mais ils introduisent une bonne diversité de classificateurs dans l'espace de recherche. Les figures 6.4 à 6.9 représentent les graphiques de la performance moyenne des classificateurs générés aléatoirement dans le temps.

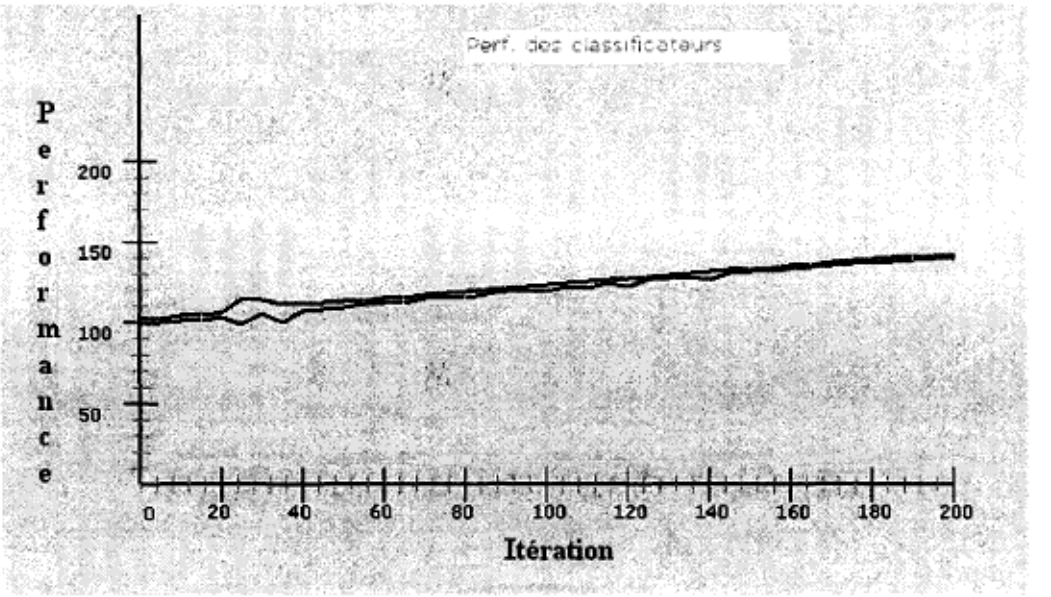

Figure 6.4: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement A.

La figure 6.4 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs aléatoires de l'environnement A. Les nombres de gènes correspondent au nombre de mots contenu dans le dictionnaire original.

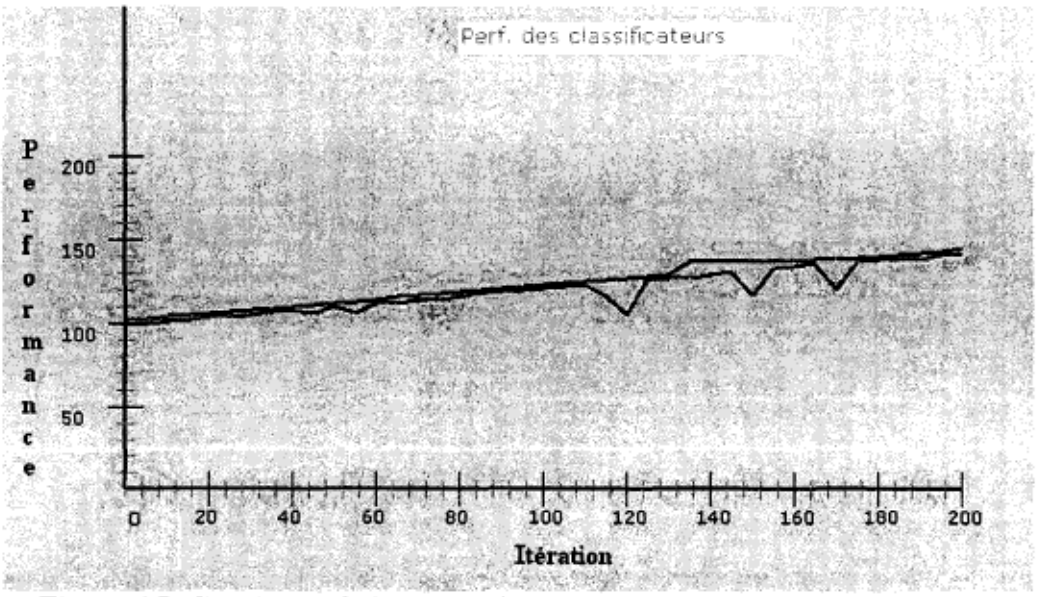

Figure 6.5: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement B.

La figure 6.5 représente la simulation de l'algorithme génétique sur des classificateurs aléatoires de l'environnement B. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les quadri-grammes de caractères avec un seuil de 85%.

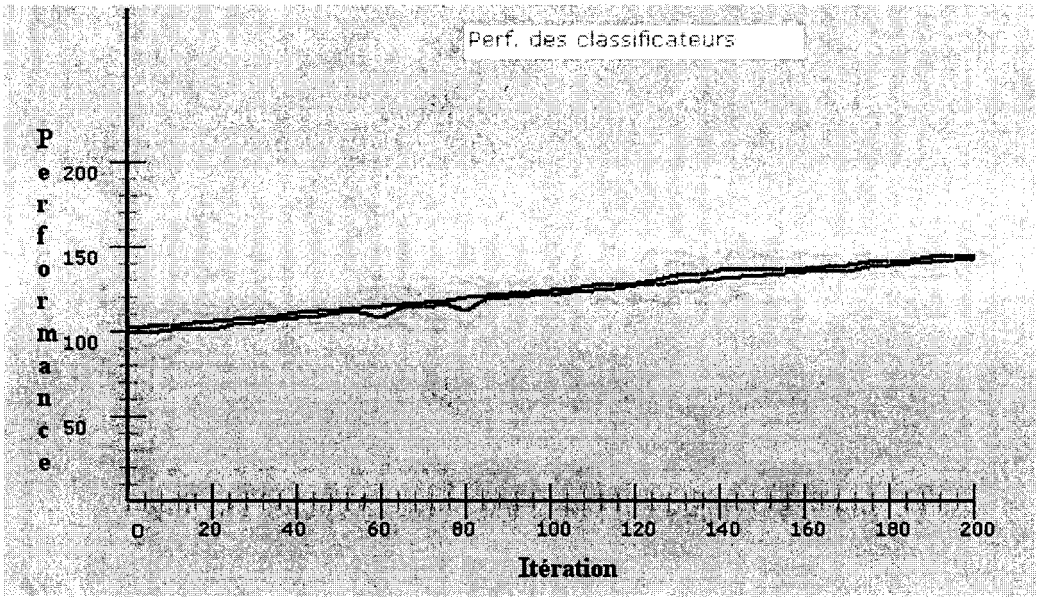

Figure 6.6: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement C.

La figure 6.6 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs de l'environnement C. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les quadri-grammes de caractères avec un seuil de 75%.

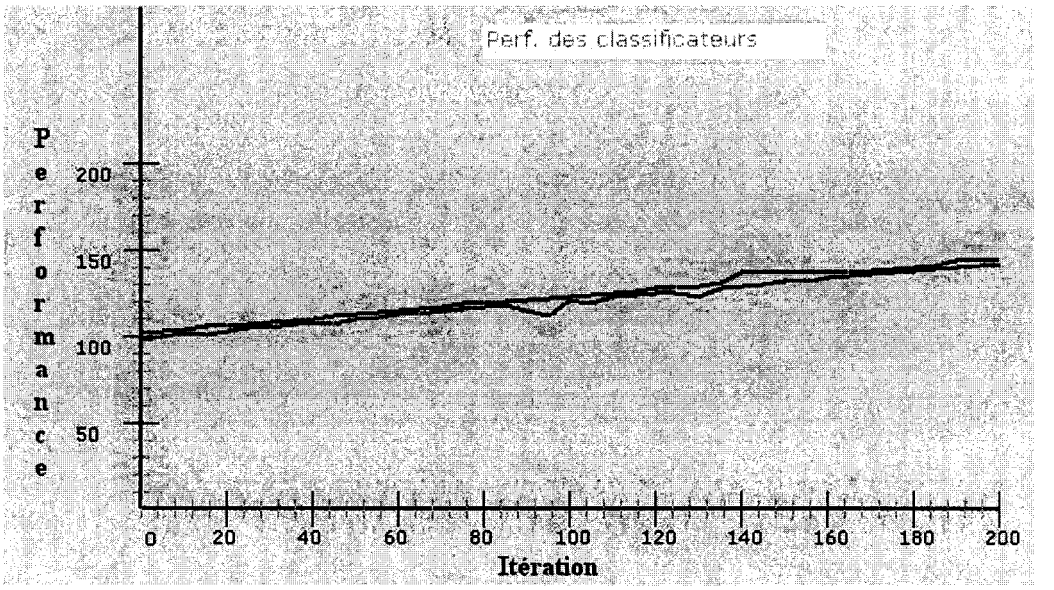

Figure 6.7: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement D.

La figure 6.7 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs aléatoires de l'environnement D. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les bi-grammes de caractères successifs avec un seuil de 85%.

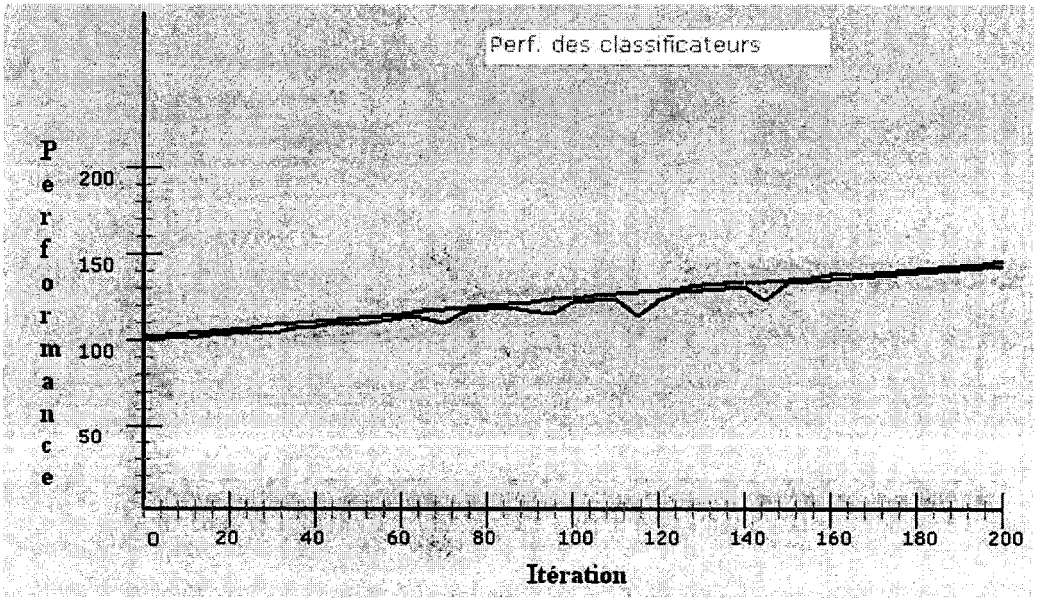

Figure 6.8: Graphique de la simulation de l'AG avec règles aléatoires pour l'environnement E.

La figure 6.8 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs aléatoires de l'environnement D. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les bi-grammes de caractères successifs avec un seuil de 75%.

Observation sur les règles aléatoires

Les gènes des classificateurs aléatoires sont diversifiés, les classificateurs ne sont pas ni trop spécifiques, ni trop génériques. Ils permettent donc une évolution des classificateurs constante. Si on observe les figures 6.4 à 6.8, on remarque que la courbe d'évolution est ascendante. La performance maximale est constamment en évolution d'une itération à l'autre.

Malgré l'évolution de la performance des classificateurs par rapport à la fonction d'évaluation, l'algorithme génétique ne produit pas de classificateurs valides (voir Annexe D).

Populations initiales de règles générées par des arbres de décision

La population de règles est composée de règles extraites d'arbres de décision. Les classificateurs ont été encodée selon avec les conditions des règles. Pour représenter les mots qui ne sont pas compris dans les règles, les gènes de ces mots étaient représentés par le symbole #. Les figures 6.9 à 6.13 représentent les graphiques de la performance moyenne des classificateurs extraits des règles extrapolées des arbres de décision dans le temps. Une règle extrapolée est une règle à laquelle on a ajouté les mots similaires dans les instances classifiées par cette règle.

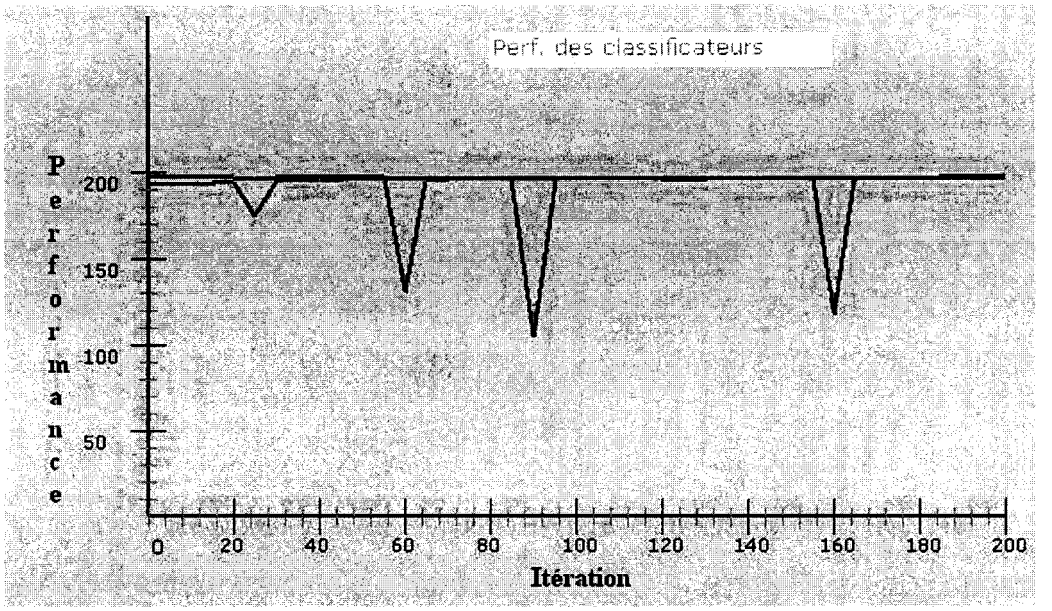

Figure 6.9: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement A.

La figure 6.9 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles extrapolées pour l'environnement A. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire original.

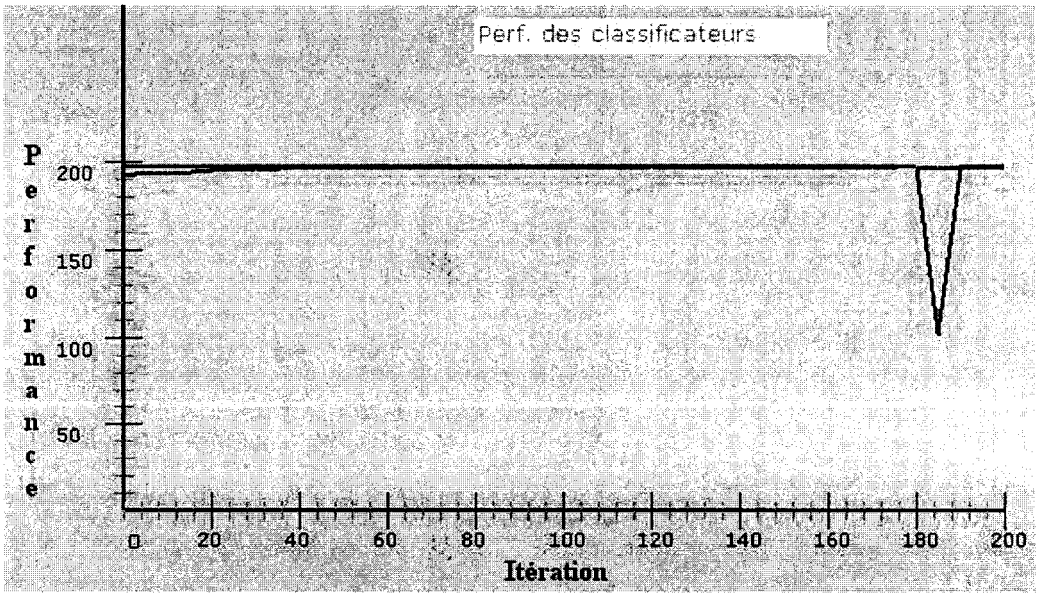

Figure 6.10: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement B.

La figure 6.10 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles extrapolées pour l'environnement A. Les nombres de gènes correspondent au nombre de mots contenu dans le dictionnaire traité avec les quadri-grammes de caractères avec un seuil de 85%.

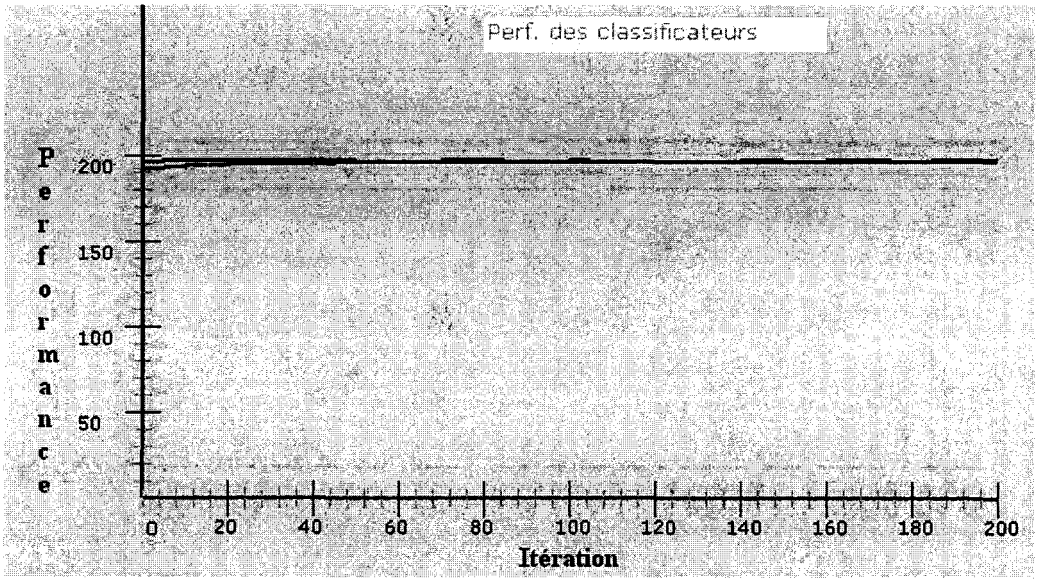

Figure 6.11: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement C.

La figure 6.11 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles extrapolées pour l'environnement C. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les quadri-grammes de caractères avec un seuil de 75%.

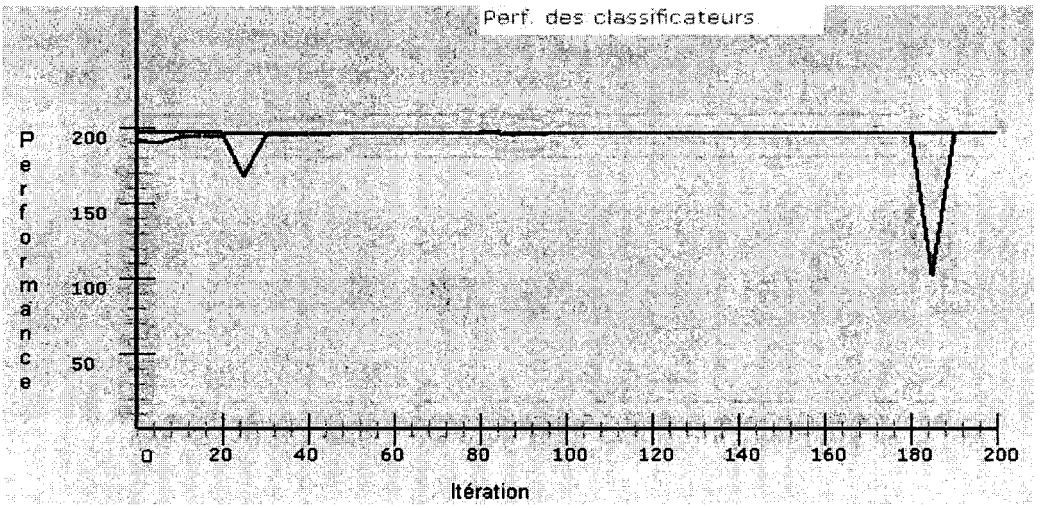

Figure 6.12: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement D.

La figure 6.12 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles extrapolées pour l'environnement D. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les bi-grammes de caractères successifs avec un seuil de 85%.

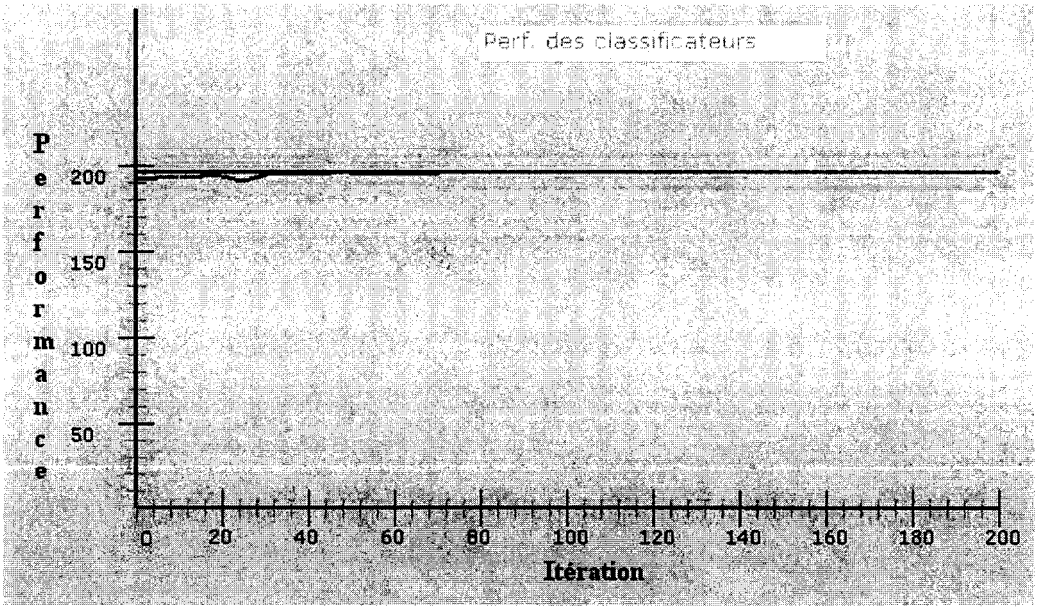

Figure 6.13: Simulation de l'AG avec règles extrapolées extraites d'AD pour l'environnement E.

La figure 6.13 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles extrapolées pour l'environnement E. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les bi-grammes de caractères successifs avec un seuil de 75%.

Les classificateurs extraits à partir des règles de décision extrapolées sont spécifiques, les gènes de ces classificateurs ne contiennent pas trop de symboles génériques. La valeur maximale est déjà forte au départ et elle n'évolue pas d'une itération à l'autre. La performance moyenne des classificateurs converge vers la performance maximale, c'est-à-dire que les classificateurs vers le maximum local de la fonction d'évaluation.

Lorsque les classificateurs sont trop spécifiques, l'algorithme génétique produit des règles acceptables (voir Annexe A), mais quand on avance dans le temps, les classificateurs générés convergent vers une classe en particulier.

Les figures 6.14 à 6.16 représentent les graphiques de la performance moyenne des classificateurs extraits des règles obtenues des arbres de décision dans le temps.

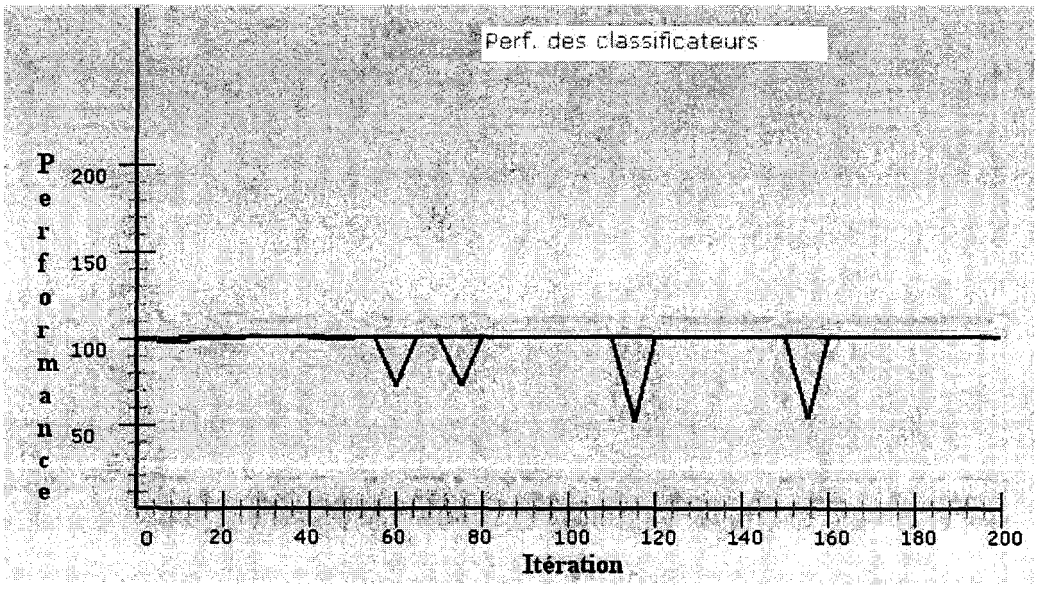

Figure 6.14 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement A.

La figure 6.14 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles génériques des arbres pour l'environnement A. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire original.

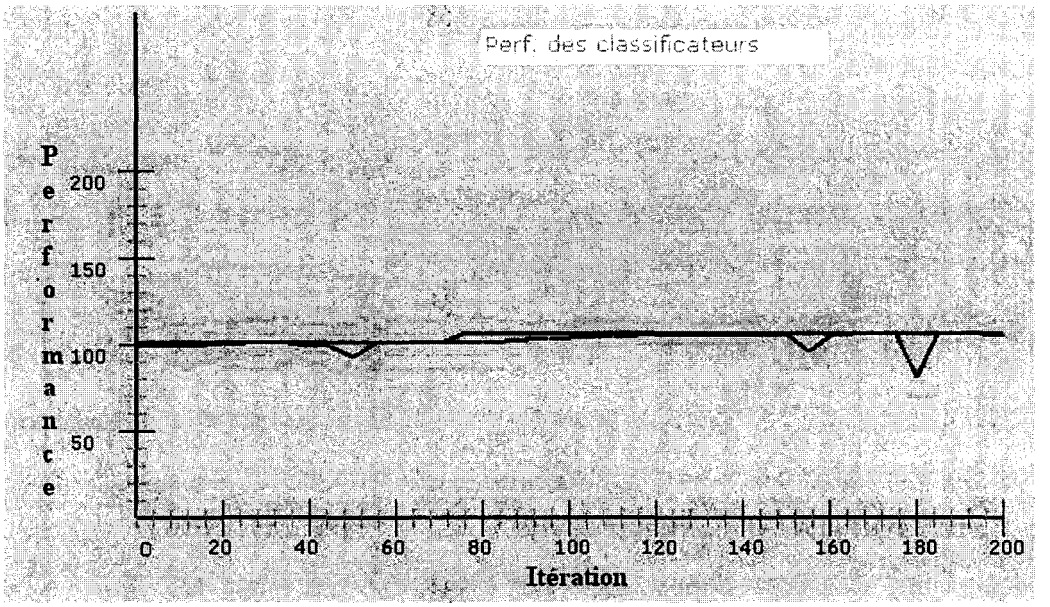

Figure 6.15 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement B.

La figure 6.15 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles génériques des arbres pour l'environnement B. Les nombres de gènes correspondent au nombre de mots contenu dans le dictionnaire traité avec les quadri-grammes de caractères avec un seuil de 85%.

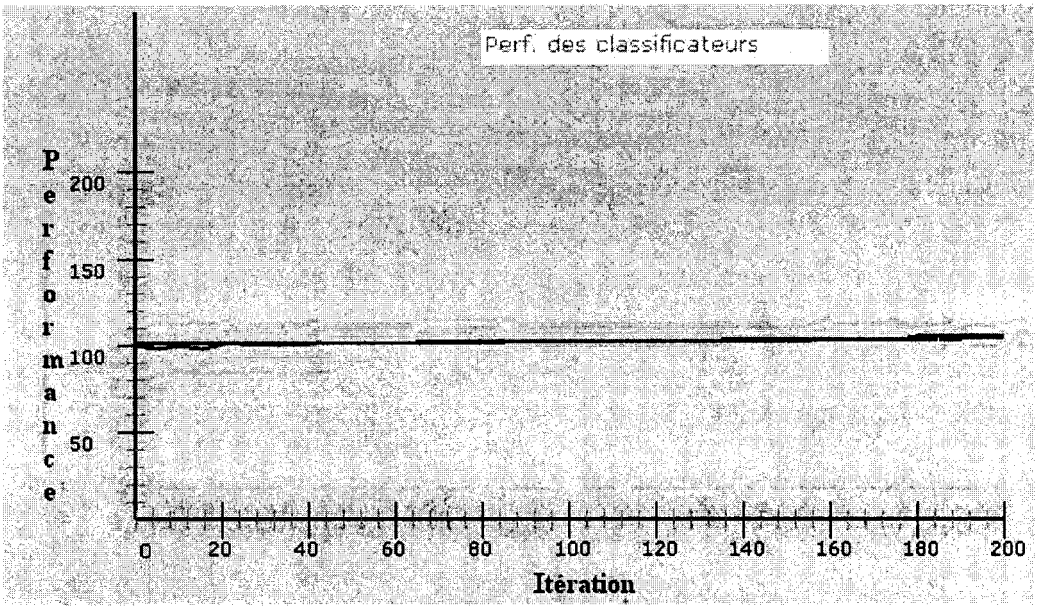

Figure 6.16 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement C.

La figure 6.16 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles génériques des arbres pour l'environnement C. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les quadri-grammes de caractères avec un seuil de 75%.

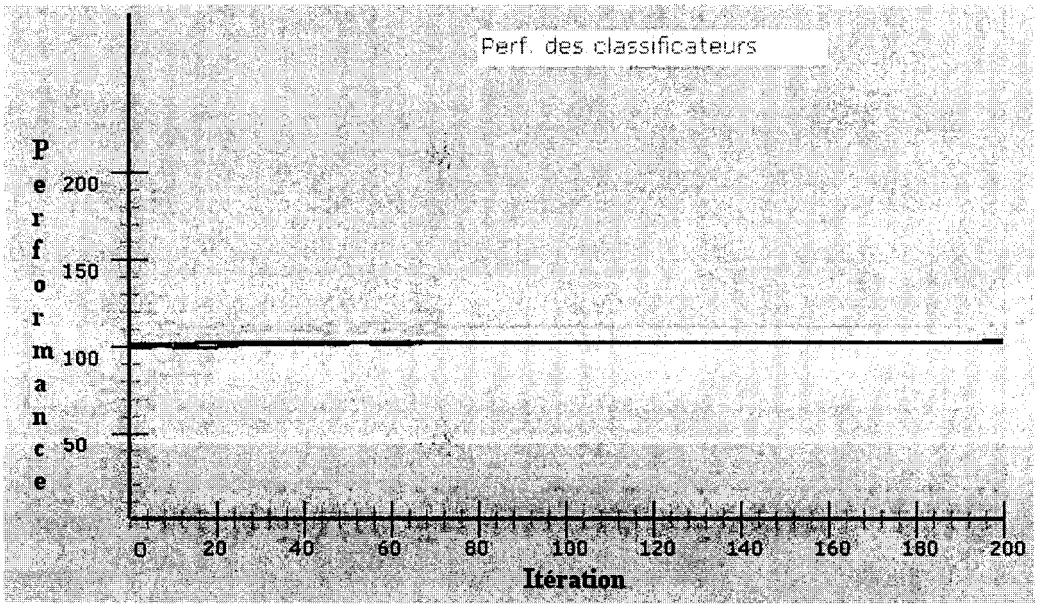

Figure 6.17 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement D.

La figure 6.17 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles génériques des arbres pour l'environnement D. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les bi-grammes de caractères successifs avec un seuil de 85%.

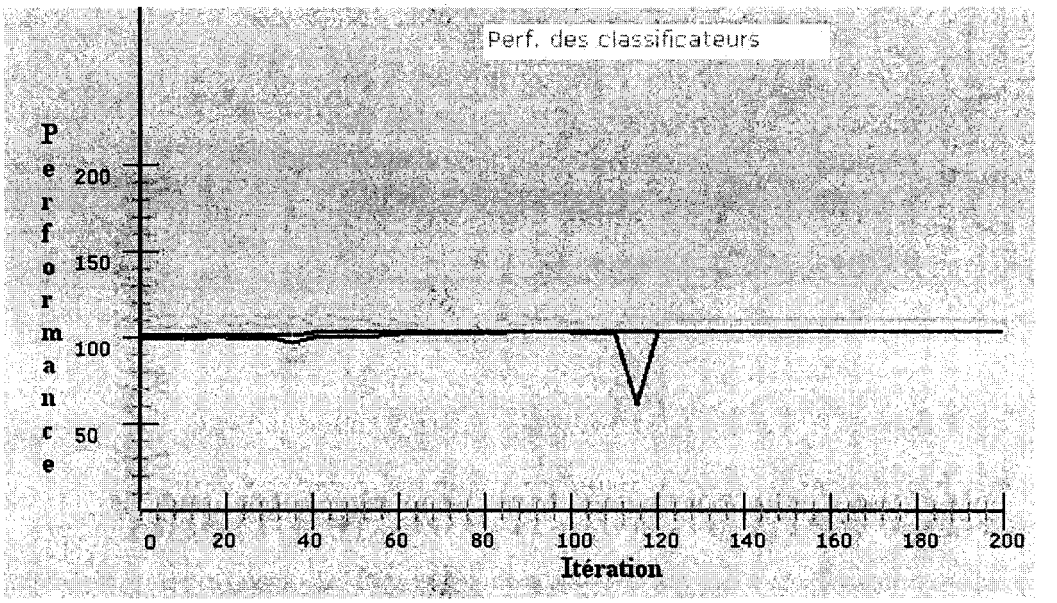

Figure 6.18 : Simulation de l'AG avec règles génériques extraites d'AD pour l'environnement E.

La figure 6.18 représente le graphique de la simulation de l'algorithme génétique sur des classificateurs représentant les règles génériques des arbres pour l'environnement E. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les bi-grammes de caractères successifs avec un seuil de 75%.

Les classificateurs extrayant des règles sans extrapolation sont trop génériques, ils ne permettent pas l'évolution des classificateurs. Plus on avance dans le temps, plus l'algorithme génétique converge vers une classe.

En général, les règles extraites des arbres de décision réussissent bien avec l'algorithme génétique. Malgré le nombre élevé de règles acceptables, certaines règles ne sont pas complètes, c'est-à-dire qu'on peut les associer à une seule classe.

Les figures 6.19 à 6.23 représentent les graphiques de la performance moyenne des classificateurs dans le temps. La population contient des règles aléatoires et des règles extraites d'arbres de décision.

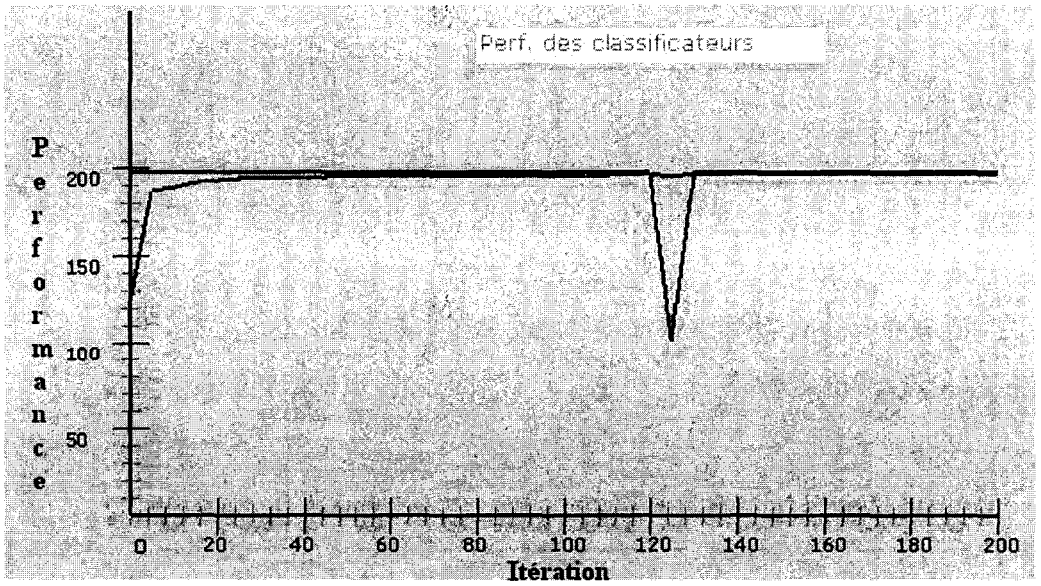

Figure 6.19 : Simulation de l'AG sur les fichiers combinés pour l'environnement A.

La figure 6.19 représente le graphique de la simulation de l'algorithme génétique sur tous les classificateurs pour l'environnement A. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire original.

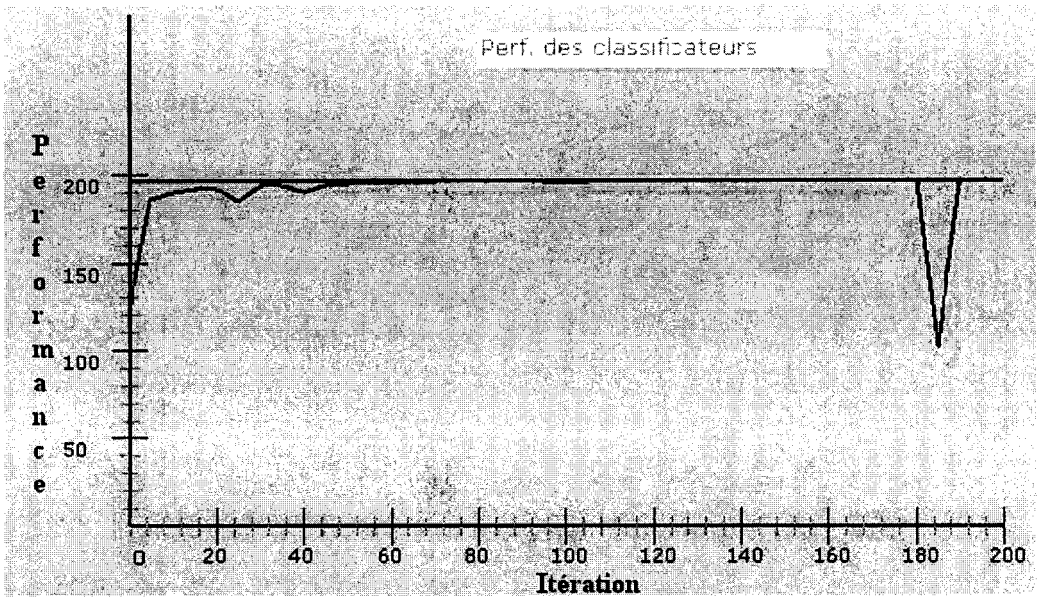

Figure 6.20: Simulation de l'AG sur les fichiers combinés pour l'environnement B.

La figure 6.20 représente le graphique de la simulation de l'algorithme génétique sur tous les classificateurs pour l'environnement B. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les quadri-grammes de caractères avec un seuil de 85%.

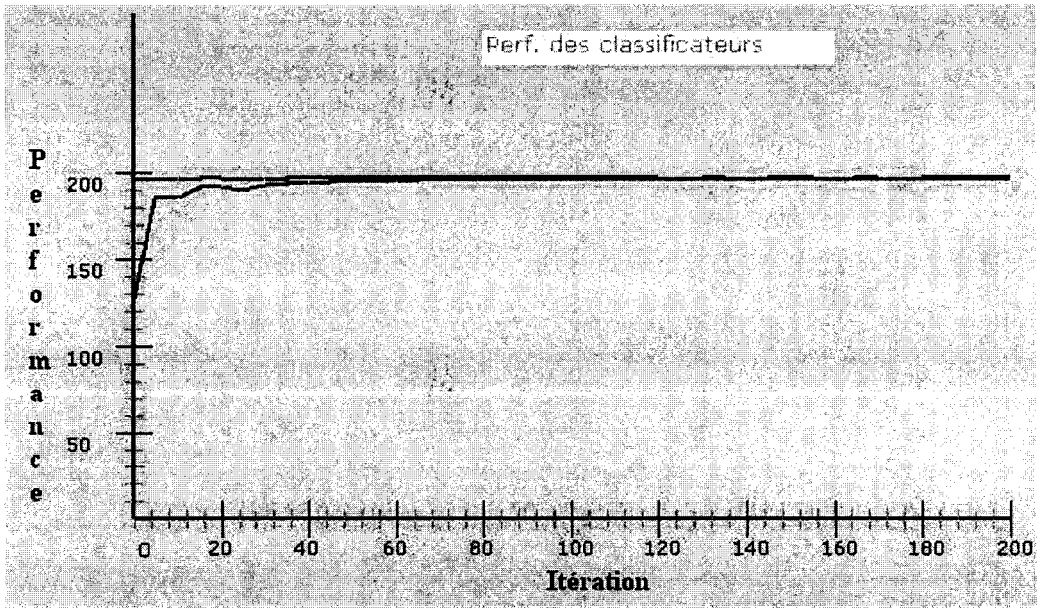

Figure 6.21: Simulation de l'AG sur les fichiers combinés pour l'environnement C.

La figure 6.21 représente le graphique de la simulation de l'algorithme génétique sur tous les classificateurs pour l'environnement C. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les quadri-grammes de caractères avec un seuil de 75%.

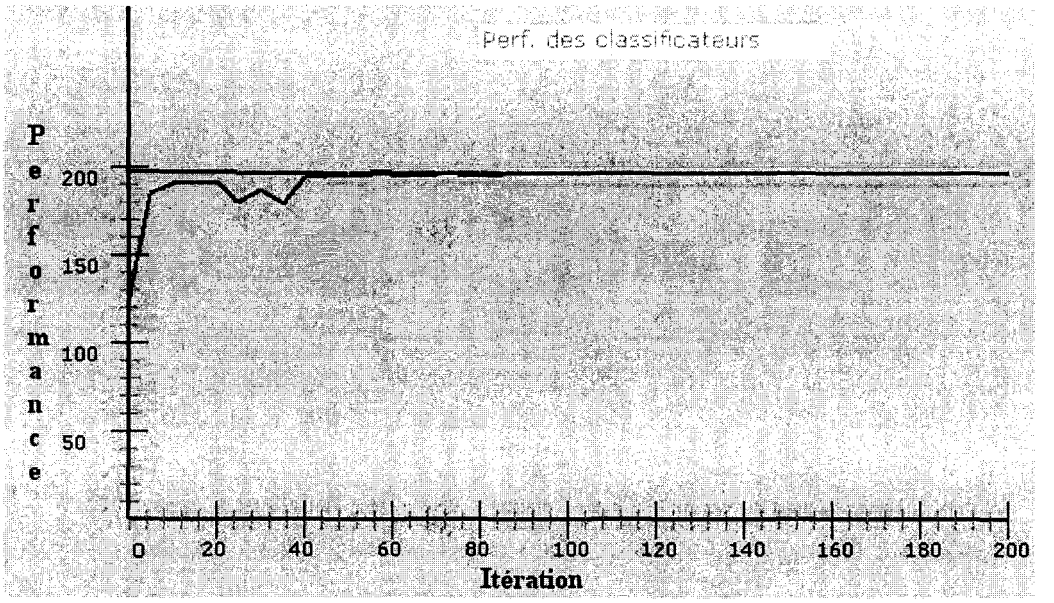

Figure 6.22: Simulation de l'AG sur les fichiers combinés pour l'environnement D.

La figure 6.22 représente le graphique de la simulation de l'algorithme génétique sur tous les classificateurs pour l'environnement D. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les bi-grammes de caractères successifs avec un seuil de 85%.

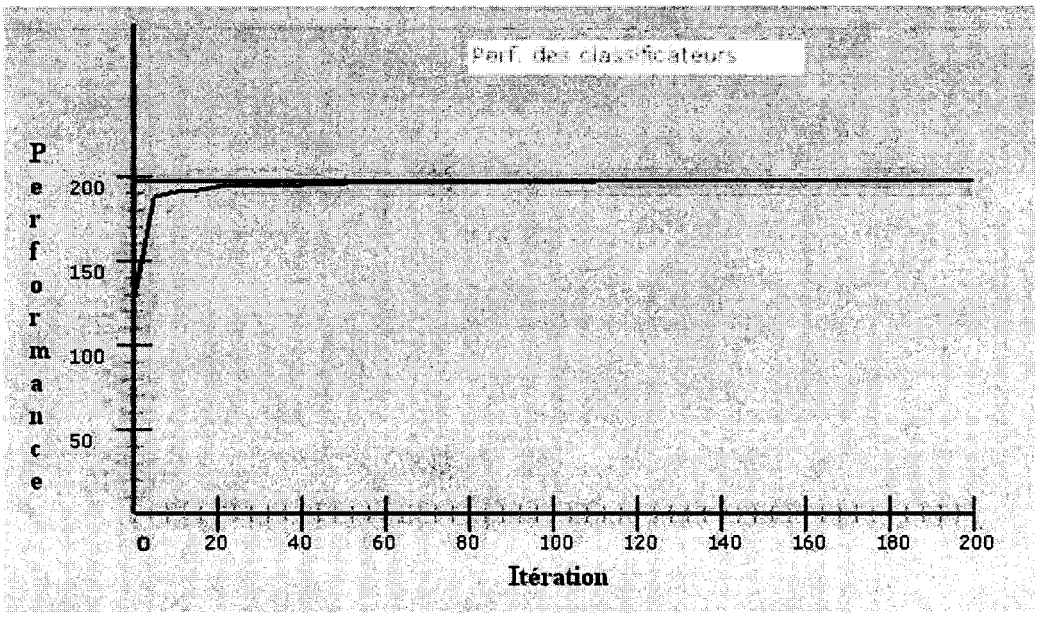

Figure 6.23: Simulation de l'AG sur les fichiers combinés pour l'environnement E.

La figure 6.23 représente le graphique de la simulation de l'algorithme génétique sur tous les classificateurs pour l'environnement D. Le nombre de gènes correspond au nombre de mots contenu dans le dictionnaire traité avec les bi-grammes de caractères successifs avec un seuil de 75%.

Les fichiers combinés représentent les règles initiales des autres fichiers, ils comprennent les fichiers aléatoires, les fichiers fabriqués à partir des arbres et les fichiers lexicaux. Comme le fichier est l'addition des autres fichiers, les classificateurs représentent une bonne base de recherche pour la production de nouveaux classificateurs à partir des classificateurs de la population existante.

L'évolution des classificateurs progresse beaucoup dans les 60 premières itérations. Après les 60 premières itérations, ils ont atteint presque le maximum de la fonction d'évaluation. On a une bonne diversité des gènes dans la population des classificateurs. L'algorithme génétique produit des règles acceptables tout au long des 200 itérations (voir annexe D).

Dans les figures 6.5, 6.9, 6.10, 6.12, 6.14, 6.15, 6.18, 6.19 et 6.20, on observe que la courbe de la moyenne fait des chutes drastiques. Ces chutes sont occasionnées par la création d'individus faibles, ces individus n'ont pas une grande espérance de vie dans le processus d'un algorithme génétique.

6.3.2 Interprétation des résultats d'un algorithme génétique

L'interprétation des résultats d'un algorithme génétique est complexe, les résultats sont liés à la fonction d'évaluation. Malgré son caractère aléatoire, un algorithme génétique produit quand même des règles plus ou moins acceptables.

La population de règles doit être représentative de l'espace de recherche. Si la population est trop aléatoire, l'espace de recherche sera diversifié, mais les règles que l'algorithme générique produit ne seront pas performantes rapidement. Avec les règles extraites des arbres, l'espace de recherche de l 'AG n'est pas vraiment diversifiées, mais les classificateurs réussissent rapidement. De plus, la validité des règles produites est augmentée.

Voici des exemples de règles:

IF abandonné(*) = 0 AND aberrante(*) = 1 AND ... AND conséquence(*) = 0 AND conservateur(*) = 0 AND .... AND winners = 0 AND world = 0 AND www = 0 AND xix = 0 AND xv = 0 AND xvie = 0 AND xx = 0 AND ya = 0 AND yaroslavsky = 0 AND york = 0 AND zapadnik(*) = 0 AND zeitschrift = 0 AND zeproject = 0 AND zeroual = 0 AND zodiaque = 0 THEN Classe 1

IF abandonné(*) = 0 AND aberrante(*) = 0 AND abjuré = 0 AND abondance(*) = 0 AND abord(*) = 0 AND absence(*) = 0 AND absolu(*) = 0 AND absorbé = 0 AND abstention(*) = 0 AND abstraire(*) = 0 AND absurdité = 0 AND abusé(*) = 0 AND académie(*) = 0 AND accéléré(*) = 0 AND accent(*) = 1 AND enterrent(*) = 1 AND enthousiasme = 0 AND entièrement(*) = 0 AND entraîneur = 0 AND entreprises(*) = 0 AND ... AND yaroslavsky = 0 AND york = 0 AND zapadnik(*) = 0 AND zeitschrift = 0 AND zeproject = 0 AND zeroual = 0 AND zodiaque = 0 THEN Classe 2

Les règles ne doivent pas être trop spécifiques, ni trop génériques. Une règle trop spécifique répond à un nombre limité de segments. Une règle trop générique est souvent incomplète, elle augmente le nombre de classifications erronées.

La diminution du nombre de gènes contribue à l'augmentation du nombre de règles acceptables. Les dictionnaires réduits produisent plus de règles qui correspondent aux jeux de tests utilisés pour valider les règles. De plus, la réduction du nombre de gènes favorise la rapidité d'exécution de l'algorithme génétique.

En général, les algorithmes génétiques réussissent bien avec les règles à condition que l'espace de recherche soit bien représenté et que la fonction d'évaluation soit adéquate, cette fonction a été définie au chapitre 4.6.7

6.4 Comparaison entre les arbres de décision et les algorithmes génétiques

Les arbres de décision et les algorithmes génétiques produisent des règles de classification à partir d'un jeu de données. Les arbres de décision produisent un bon schéma de classification des segments de textes du jeu d'apprentissages, tandis qu'un algorithme génétique travaille à partir d'une population de règles existantes et les combine pour faire d'autres règles.

Pour les arbres de décision, l'ordre des nœuds est déterminant dans le cheminement de la classification. Pour l'algorithme génétique, l'ordre n'apporte pas d'information sur la pertinence d'un gène par rapport à un autre, contrairement à un nœud d'un arbre de décision.

Les arbres de décision cherchent à établir les plus petites règles possible. Pour certains nœuds, on aurait pu choisir un autre attribut qui aurait un gain similaire. Pour compléter les règles de classification de texte, on a besoin d'extrapoler ces règles avec les exemples classés par les branches. Les arbres de décision travaillent récursivement pour classer les exemples des branches du nœud précédent.

Les règles extrapolées sont une bonne base de référence pour classer des textes, elles contiennent les attributs nécessaires pour identifier les classes. Ces attributs sont présents dans tous les segments de texte du jeu d'apprentissages.

Un algorithme génétique est un processus itératif, il travaille sur la population de règles, durant une itération, on sélectionne des règles pour les coupler afin de créer les nouvelles règles. L'algorithme génétique favorise la compétition entre les règles les plus performantes par rapport à une fonction d'évaluation. La fonction d'évaluation décrite à la section 4.6.7, a une grande influence sur la pertinence des nouveaux individus de la future génération.

Malgré l'augmentation de la performance des classificateurs, l'algorithme génétique favorise les individus performants au détriment des individus moins performants. Ceci contribue à la perte d'information sur certaines classes moins performantes.

L'algorithme génétique produit des règles plus performantes lorsque le fichier initial comprend des règles extraites à partir d'un arbre, ces règles reflètent les caractéristiques principales des différentes classes. L'impact d'un algorithme génétique sur les classificateurs dépend de sa population initiale. Les règles de classification extraites des arbres de décision sont une bonne base de connaissance pour produire d'autres règles valides. Lorsque la population initiale est aléatoire, l'algorithme génétique améliore la performance des classificateurs constamment, mais il ne permet pas de produire des règles valides.

La compression du lexique des mots avec les n-grammes contribue grandement à la pertinence des règles de classification produites par les arbres de décision et les algorithmes génétiques. Pour les arbres de décision, le regroupement de mots similaires diminue le nombre de choix potentiel pour un nœud et en augmentant les chances des regroupements de mots. Pour l'algorithme génétique, la diminution du nombre de gènes augmente la rapidité d'exécution de ses opérateurs génétiques.

Un algorithme génétique peut être utilisé en concordance avec l'algorithme de construction d'un arbre, il peut servir de mesure de segmentation ou encore être utilisé lors de l'élagage.

En général, les arbres de décision présentent une meilleure diversité des classes et un meilleur schéma de la classification qu'avec un algorithme génétique. De plus, la construction d'un arbre s'effectue rapidement comparée à un algorithme génétique qui nécessite un grand nombre d'itérations pour arriver à un résultat acceptable.

CHAPITRE 7 CONCLUSION

Dans une classification de texte, le nombre de mots est très élevé. La réduction du nombre de mots maximise la classification en regroupant les mots similaires. L'annexe A donne des exemples de regroupement de mots similaire. Cela favorise la rapidité du calcul des algorithmes de classification. Cependant, cette réduction peut occasionner du bruit dans la classification, pour éviter du bruit inutilement, la préparation des données doit être effectuée avec attention.

Les algorithmes génétiques, que l'on a présentés au chapitre 3, sont des outils très puissants pour l'optimisation de problème. La force des AG réside dans ces opérateurs génétiques, le choix judicieux de ces opérateurs a une grande influence sur la réussite des AG. Comme ils sont fondés sur la théorie de l'évolution, ils permettent une bonne prédiction sur l'évolution des prochaines générations d'individus. Le mécanisme d'un AG permet une recherche efficace, car il utilise implicitement les similarités entre les individus pour extraire de l'information sur les régions de l'espace où la fonction d'évaluation présente les plus fortes valeurs. Il est très intéressant de les utiliser quand l'espace de recherche est très grand ou complexe.

Les algorithmes génétiques sont très complexes à gérer, il y a beaucoup de facteurs à considérer pour la réussite, la fonction d'évaluation constitue le critère le plus important, les autres critères dépendent des paramètres et de la population initiale de l'AG.

La population initiale des règles doit être assez diversifiée, si la population est trop spécifique, la fonction d'évaluation converge vers un maximum local à cette fonction et lorsque la population initiale est trop générique, les règles convergent vers des règles qui sont généralement incomplètes. Les règles de décision extraites des arbres représentent généralement une bonne source initiale pour réussir avec un algorithme

génétique. Lorsque ces règles sont extrapolées, la fonction d'évaluation est proche de sa performance optimale et lorsque les règles représentent uniquement les caractéristiques de l'arbre, la performance des règles évolue lentement. La population de règles doit être diversifiée, afin que l'espace de recherche de l'algorithme soit assez grand et qu'il ait la chance de produire de nouvelles règles performantes et acceptables.

De l'autre côté, les arbres de décision, que l'on a présentés au chapitre 2, sont moins complexes à gérer, ils donnent une bonne idée du cheminement de la classification. Les arbres de décision représentent une illustration de la classification du texte. Ils permettent de voir les attributs les plus importants, plus un attribut est proche de la racine, plus il est important dans la classification. Les arbres de décision recherchent le moyen de minimiser les règles. Cependant, les règles générées avec un arbre de décision ne prennent pas en compte tous les mots du lexique, on a besoin d'extrapoler les règles pour avoir un résultat plus précis. Les deux méthodes utilisées pour créer les arbres de décision peuvent être considérées comme complémentaires. La méthode C4.5 donne une meilleure répartition des classes, mais elle augmente le nombre de règles. La méthode CART échelonne les classes selon le nombre d'exemples selon le nombre d'exemples dans la classe, chacune des classes est représentée seulement par une règle.

Si les algorithmes génétiques sont dépendants de sa population de règles diversifiées, les arbres de décision sont dépendants des jeux d'apprentissages. Les exemples qui composent un jeu d'apprentissages doivent être représentatifs de l'ensemble des classes.

Pour conclure, les deux algorithmes de classification sont de bons outils pour trouver des règles de classification textuelle, cependant les arbres de décision réussissent plus rapidement qu'un algorithme génétique, les règles produites par l'arbre sont plus représentatives de l'ensemble des classes, tandis qu'un algorithme génétique nécessite un grand nombre de facteurs pour pouvoir réussir.

Annexe A

Exemple d'utilisation de n-gramme

Les n-grammes sont des suites de n caract ères contenues dans les mots. Pour accélérer le traitement des algorithmes de classification. L'annexe A nous présente des exemples de regroupements de mots faits à l'aide de n-gramme. Pour réduire le lexique de mots, nous avons utilisé deux techniques de réduction de mots : les quadri-grammes et les bi-grammes succesifs.

Les quadri-grannes sont des suites de 4 caractères contenus dans un mot. Les bi-grammes succesifs sont les suites de 2 caractères contenues dans un mot, contraiment au quadri-grammes, on vérifie si les bi-grammes sont successifs.

Pour les exemples de l'annexe A, nous avons utilisé un seuil de 75% pour traîter les mots. Ce seuil est assez petit de façon à regrouper les mots similaires sans avoir trop de bruits, c'est-à-dire des mots similaires à l'écriture, mais qui n'ont pas le même sens. Plus le seuil sera petit, plus le bruit sera important.

Étude sur les Quadri-grammes avec un seuil de 75%	
absolu(*) :absolu,absolue	assure(*) :assure,assurer
abstraction(*) :abstraction,abstractions	astronaut(*) :astronaut,astronauts
abstrait(*) :abstrait,abstraite	astronomer(*) :astronomer,astronomers
académique(*) :académique,académiques	atlantiste(*) :atlantiste,atlantistes
accidental(*) :accidental,occidental	atmosphere(*) :atmosphere,atmospheres
accompagné(*) :accompagné,accompagnée	attitude(*) :attitude,attitudes
accord(*) :accord,accorde,accordé,accords	attribué(*) :attribué,attribués
accueilli(*) :accueilli,accueillir	augure(*) :augure,augurer
accumule(*) :accumule,accumuler	australie(*) :australie,australiens
acheter(*) :acheter,racheter	auteur(*) :auteur,auteurs
acteur(*) :acteur,facteur	autorité(*) :autorité,autorités
action(*) :action,actions	avocat(*) :avocat,avocats
activité(*) :activité,activités	badger(*) :badger,badgers
admire(*) :admire,admired	ballade(*) :ballade,ballades
advance(*) :advance,advanced,advances	bancaire(*) :bancaire,bancaires
adversaire(*) :adversaire,adversaires	basket(*) :basket,baskets
advocate(*) :advocate,advocates	bataille(*) :bataille,batailles
affair(*) :affair,affaire	belgian(*) :belgian,belgians
affirme(*) :affirme,affirmer	believe(*) :believe,believed,believes
afghan(*) :afghan,afghans	benefit(*) :benefit,benefits
africa(*) :africa,african	besoin(*) :besoin,besoins
ajoute(*) :ajoute,ajouter	blanche(*) :blanche,blanches
allait(*) :allait,fallait	blessé(*) :blessé,blessés
allemand(*) :allemand,allemande,allemands	bolchevique(*) :bolchevique,bolcheviques
ambition(*) :ambition,ambitions	bolchevisme(*) :bolchevisme,bolchevismes
america(*) :america,american	bourgeois(*) :bourgeois,bourgeoisie
américain(*) :américain,américaine,américains	bourse(*) :bourse,bourses
amérique(*) :amérique,amériques	budget(*) :budget,budgets
amoralisme(*) :amoralisme,moralisme	capital(*) :capital,capitale
amorce(*) :amorce,amorcer	capitaliste(*) :capitaliste,capitalistes
analyste(*) :analyste,analystes	cavalier(*) :cavalier,cavaliers
anarchiste(*) :anarchiste,anarchistes	célèbre(*) :célèbre,célèbres
ancien(*) :ancien,anciens,ancient	central(*) :central,centrale
announce(*) :announce,announced	centralisation(*) :centralisation,décentralisation
antisémite(*) :antisémite,antisémites	centre(*) :centre,centres
apparaissent(*) :apparaissent,paraissent	cercle(*) :cercle,cercles
apparent(*) :apparent,apparente	challenger(*) :challenger,challenges
appeal(*) :appeal,appeals	champion(*) :champion,champions
apport(*) :apport,apporte,apports,rapport	chance(*) :chance,chances
approche(*) :approche,approches,rapproche	change(*)
approve(*) :approve,approved	:change,changed,changer,changes,échange
archaïsant(*) :archaïsant,archaïsante,archaïsants	changement(*) :changement,changements
argument(*) :argument,arguments	chapel(*) :chapel,chapela
aristocrate(*) :aristocrate,aristocrates	character(*) :character,characters
arrête(*) :arrête,arrêter	charge(*) :charge,charged,charges
article(*) :article,articles	chemical(*) :chemical,chemicals
artificial(*) :artificial,artificially	chéquier(*) :chéquier,chéquiers
asiatique(*) :asiatique,asiatiques	chimère(*) :chimère,chimères
assistant(*) :assistant,assistants	chrétien(*) :chrétien,chrétiens

christoph(*) :christoph,christopher
cidentale(*) :cidentale,occidentale
civilisation(*) :civilisation,civilisations
claire(*) :claire,claires
classe(*) :classe,classes
classique(*) :classique,classiques
client(*) :client,clients
climat(*) :climat,climate
cluster(*) :cluster,clusters
colonial(*) :colonial,coloniale
combat(*) :combat,combats
commence(*) :commence,commencer
commission(*) :commission,commissions
commun(*) :commun,commune,communs
communiste(*) :communiste,communistes
compare(*) :compare,compared,comparer
compétence(*) :compétence,compétences
complaint(*) :complaint,complaints
complet(*) :complet,complete
compte(*) :compte,compter
concentrée(*) :concentrée,concentrées
conception(*) :conception,conceptions
concret(*) :concret,concrete
condition(*) :condition,conditions
conduit(*) :conduit,conduite
conflit(*) :conflit,conflits
connaissance(*) :connaissance,connaissances
connaissent(*) :connaissent,reconnaissent
connaître(*) :connaître,reconnaître
connect(*) :connect,connecté
connotation(*) :connotation,connotations
conscient(*) :conscient,consciente,conscients
conséquence(*) :conséquence,conséquences
conservateur(*) :conservateur,conservateurs
conservatrice(*) :conservatrice,conservatrices
conserve(*) :conserve,conserver
considérée(*) :considérée,considérés
constante(*) :constante,constantes
constate(*) :constate,constater
constitue(*) :constitue,constituer
constitution(*) :constitution,constitutionnel
constitutionnelle(*)
:constitutionnelle,constitutionnelles,institutionnelle
construction(*) :construction,constructions
construit(*) :construit,déconstruit
contender(*) :contender,contenders
contextualisation(*)
:contextualisation,décontextualisation
continent(*) :continent,continental
continue(*) :continue,continued,continuer
contraceptive(*) :contraceptive,contraceptives
contradictoire(*) :contradictoire,contradictoires
contrainte(*) :contrainte,contraints
contrast(*) :contrast,contraste
controverse(*) :controverse,controversy
conviction(*) :conviction,convictions
courant(*) :courant,courante,courants
courrier(*) :courrier,courriers
course(*) :course,courses
courte(*) :courte,courtes
create(*) :create,created
créateur(*) :créateur,créateurs
critique(*) :critique,critiques
culture(*) :culture,culturel,cultures
culturelle(*) :culturelle,culturelles
cyclique(*) :cyclique,cycliques
danger(*) :danger,dangers
danilevski(*) :danilevski,danilevsky
decision(*) :decision,decisions
découvert(*) :découvert,découverte
découvrir(*) :découvrir,redécouvrir
défend(*) :défend,défendu
défini(*) :défini,définir
dégage(*) :dégage,dégager
delegation(*) :delegation,delegations
demand(*)
:demand,demanda,demande,demandé
demander(*) :demander,demandera
demeure(*) :demeure,demeurer
démocratique(*) :démocratique,démocratiques
dénoncé(*) :dénoncé,énoncé
dépendance(*) :dépendance,indépendance
déplacement(*) :déplacement,placement
déposée(*) :déposée,déposées
désabusé(*) :désabusé,désabusés
design(*) :design,designs
détail(*) :détail,détails
détenu(*) :détenu,détenue
determine(*) :determine,determined
development(*) :development,developments
développe(*)
:développe,développent,développer
devenu(*) :devenu,devenue,devenus
diamond(*) :diamond,diamonds
didactique(*) :didactique,didactiques
différent(*)
:différent,différente,différentes,différents
difficile(*) :difficile,difficiles
difficult(*) :difficult,difficulté,difficultés
dimension(*) :dimension,dimensions
direct(*) :direct,directe,directs
director(*) :director,directors

dirigé(*) :dirigé,dirigée
dirigeant(*) :dirigeant,dirigeants
disciple(*) :disciple,disciples
disease(*) :disease,diseases
disparu(*) :disparu,disparus
disponible(*) :disponible,disponibles
disposition(*) :disposition,dispositions
distance(*) :distance,distances
distincte(*) :distincte,distinctes
distributor(*) :distributor,distributors
dizaine(*) :dizaine,dizaines
doctrine(*) :doctrine,doctrines
document(*) :document,documents
dominate(*) :dominate,dominated
droite(*) :droite,droites
earning(*) :earning,learning
économie(*) :économie,économies
économique(*) :économique,économiques
écrivain(*) :écrivain,écrivains
éditée(*) :éditée,éditées
éditeur(*) :éditeur,éditeurs
édition(*) :édition,éditions
effect(*) :effect,effects
effort(*) :effort,efforts
église(*) :église,églises
élabore(*) :élabore,élaborer
élaboré(*) :élaboré,élaborée
électeurs(*) :électeurs,lecteurs
élection(*) :élection,élections
électoral(*) :électoral,électorale
électronique(*) :électronique,électroniques
éloigne(*) :éloigne,éloigner
employé(*) :employé,employés
employee(*) :employee,employees
encontre(*) :encontre,rencontre
encyclopédique(*)
:encyclopédique,encyclopédiques
encyclopédiste(*)
:encyclopédiste,encyclopédistes
énergie(*) :énergie,énergies,nergie
enfant(*) :enfant,enfants
engage(*) :engage,engager
engagé(*) :engagé,engagée
énigmatique(*) :énigmatique,énigmatiques
ennemi(*) :ennemi,ennemis
enquête(*) :enquête,enquêtes
enregistré(*) :enregistré,enregistrées,enregistrés
enseignant(*) :enseignant,enseignants
entendu(*) :entendu,entendue,entendus
entreprise(*) :entreprise,entreprises
énumératif(*) :énumératif,énumératifs
énumération(*) :énumération,énumérations
environment(*) :environment,environmental
envoyer(*) :envoyer,renvoyer
équilibre(*) :équilibre,équilibrent
équipe(*) :équipe,équipes
équivalent(*) :équivalent,équivalente
erreur(*) :erreur,erreurs
eserver(*) :eserver,eservers
espace(*) :espace,espaces
estime(*) :estime,estimer
établir(*) :établir,rétablir
étayée(*) :étayée,étayées
ethnie(*) :ethnie,ethnies
ethnique(*) :ethnique,ethniques
étrange(*) :étrange,trange
eurasiatique(*) :eurasiatique,eurasiatiques
eurasie(*) :eurasie,eurasien
eurasiste(*) :eurasiste,eurasistes
européen(*) :européen,européens
européenne(*) :européenne,européennes
évident(*) :évident,évidente
évolution(*)
:évolution,évolutions,révolution,volution
except(*) :except,excepté
exemple(*) :exemple,exemples
exhibitor(*) :exhibitor,exhibitors
existe(*) :existe,exister
experience(*)
:experience,experienced,experiences
expérience(*) :expérience,expériences
explained(*) :explained,unexplained
explique(*) :explique,expliquer
exploite(*) :exploite,exploiter
exprime(*) :exprime,exprimer
extend(*) :extend,extends
extérieur(*) :extérieur,extérieure,extérieures
extradition(*) :extradition,tradition
faible(*) :faible,faibles
famille(*) :famille,familles
fasciste(*) :fasciste,fascistes
favori(*) :favori,favoris
fellow(*) :fellow,fellows
female(*) :female,females
fiable(*) :fiable,fiables
figure(*) :figure,figures
filmmaker(*) :filmmaker,filmmakers
finding(*) :finding,findings
fonction(*) :fonction,fonctions
formation(*) :formation,information
formelle(*) :formelle,formelles
fourni(*) :fourni,fournir,fournit
fragment(*) :fragment,fragments
français(*) :français,française

francophone(*) :francophone,francophones
futuriste(*) :futuriste,futuristes
gauchiste(*) :gauchiste,gauchistes
généra(*) :généra,général
generation(*) :generation,generations
géopolitique(*) :géopolitique,géopolitiques
géopolitiquement(*)
:géopolitiquement,politiquement
george(*) :george,georges
glissement(*) :glissement,glissements
global(*) :global,globale
glossaire(*) :glossaire,glossaires
gouvernement(*)
:gouvernement,gouvernemental
government(*) :government,governmental
grande(*) :grande,grandes
grandi(*) :grandi,grandit
groupe(*) :groupe,groupes
groupements(*) :groupements,regroupements
habillement(*) :habillement,habillements
handle(*) :handle,handles
happen(*) :happen,happens
health(*) :health,healthy
héroïque(*) :héroïque,héroïques
histoire(*) :histoire,histoires
historien(*) :historien,historiens
historique(*) :historique,historiques
honneur(*) :honneur,honneurs
hospital(*) :hospital,hospitals
humain(*) :humain,humains
humaniste(*) :humaniste,humanistes
idéaliste(*) :idéaliste,idéalistes
identified(*) :identified,identifier
identitaire(*) :identitaire,identitaires
idéologique(*) :idéologique,idéologiques
idéologue(*) :idéologue,idéologues
illusion(*) :illusion,illusions
immédiat(*) :immédiat,immédiate
impérial(*) :impérial,impériale
important(*)
:important,importante,importantes,importants
importe(*) :importe,imported
impose(*) :impose,imposed,imposer
impossibilité(*) :impossibilité,possibilité
imprécatoire(*) :imprécatoire,imprécatoires
incapable(*) :incapable,incapables
include(*) :include,included,includes
increase(*) :increase,increased
indépendante(*) :indépendante,indépendantes
indication(*) :indication,indications
individuel(*) :individuel,individuelle
industrie(*) :industrie,industriel
inégalité(*) :inégalité,inégalités
influence(*) :influence,influenced,influencer
influencé(*) :influencé,influencés
informatique(*) :informatique,informatiques
inspiration(*) :inspiration,inspirational
inspire(*) :inspire,inspired
instinct(*) :instinct,instincts
institut(*) :institut,institute,instituts
instrument(*) :instrument,instruments
insuffisamment(*) :insuffisamment,suffisamment
insuffisante(*) :insuffisante,suffisante
intellectuel(*)
:intellectuel,intellectuelle,intellectuelles,intellectuels
intention(*) :intention,intentions
intéresse(*) :intéresse,intéressent
intéressé(*) :intéressé,intéressés
intérêt(*) :intérêt,intérêts
intérieur(*) :intérieur,intérieure,intérieures
international(*) :international,internationale
internationaliste(*)
:internationaliste,internationalistes,ternationaliste
internaute(*) :internaute,internautes
interprété(*) :interprété,interprétées
intervention(*) :intervention,interventions
introduce(*) :introduce,introduced
inventaire(*) :inventaire,inventaires
invisible(*) :invisible,invisibles
involve(*) :involve,involved
israel(*) :israel,israeli
khrouchtchev(*) :khrouchtchev,khrouchtchevien
laisse(*) :laisse,laisser
langue(*) :langue,langues
laying(*) :laying,playing
leader(*) :leader,leaders
lettre(*) :lettre,lettres
libéral(*) :libéral,libérale
lightly(*) :lightly,slightly
logiciel(*) :logiciel,logiciels
logique(*) :logique,logiques
lutionnaires(*) :lutionnaires,volutionnaires
lyakhovic(*) :lyakhovic,lyakhovich
machine(*) :machine,machines
majeur(*) :majeur,majeure
maladie(*) :maladie,maladies
manager(*) :manager,managers
manifeste(*) :manifeste,manifestent,manifester
marché(*) :marché,marchés
maritime(*) :maritime,maritimes
marxiste(*) :marxiste,marxistes

matière(*) :matière,matières
mécanisme(*) :mécanisme,mécanismes
medicine(*) :medicine,medicines
member(*) :member,members
menace(*) :menace,menaces
mesure(*) :mesure,mesurer
métier(*) :métier,métiers
milieu(*) :milieu,milieux
militaire(*) :militaire,militaires
milliard(*) :milliard,milliards
million(*) :million,millions
mineur(*) :mineur,mineurs
ministre(*) :ministre,ministres
minorité(*) :minorité,minorités
minute(*) :minute,minutes
modèle(*) :modèle,modèles
moderne(*) :moderne,modernes
modification(*) :modification,modifications
monarchiste(*) :monarchiste,monarchistes
mondial(*) :mondial,mondiale
mongole(*) :mongole,mongoles
mouvement(*) :mouvement,mouvements
municipal(*) :municipal,municipales
musical(*) :musical,musicale
musique(*) :musique,musiques
musulman(*) :musulman,musulmans
mystique(*) :mystique,mystiques
narodniki(*) :narodniki,narodnikis
nation(*) :nation,nations
national(*) :national,nationale
nationaliste(*)
:nationaliste,nationalistes,rationaliste
nature(*) :nature,naturel
nécessaire(*) :nécessaire,nécessaires
nigeria(*) :nigeria,nigerian
normal(*) :normal,normale
nostalgique(*) :nostalgique,nostalgiques
numéro(*) :numéro,numéros
objectif(*) :objectif,objectifs
obligatoire(*) :obligatoire,obligatoires
observation(*) :observation,observations
observe(*) :observe,observer
occidentaliste(*) :occidentaliste,occidentalistes
oeuvre(*) :oeuvre,oeuvres
office(*) :office,officer,offices
official(*) :official,officials
olympic(*) :olympic,olympics
onomasiologique(*)
:onomasiologique,onomasiologiques
opérateur(*) :opérateur,opérateurs
opposant(*) :opposant,opposants
oppose(*) :oppose,opposed
option(*) :option,options
organisation(*) :organisation,organisations
organize(*) :organize,organized,organizer
orient(*) :orient,oriente
orientale(*) :orientale,orientales
orientation(*) :orientation,orientations
original(*) :original,originale
orthodoxe(*) :orthodoxe,orthodoxes
ouvert(*) :ouvert,ouverte
ouvrage(*) :ouvrage,ouvrages
palestinian(*) :palestinian,palestinians
parallèle(*) :parallèle,parallèles
parent(*) :parent,parenté,parents
partenaire(*) :partenaire,partenaires
particularité(*) :particularité,particularités
particulier(*) :particulier,particuliers
partisan(*) :partisan,partisans
passion(*) :passion,passions
patient(*) :patient,patients
patriarcal(*) :patriarcal,patriarcale
patriotic(*) :patriotic,patriotica
patriotique(*) :patriotique,patriotiques
pensée(*) :pensée,pensées
période(*) :période,périodes
personnalité(*) :personnalité,personnalités
personne(*) :personne,personnel,personnes
petite(*) :petite,petites
peuple(*) :peuple,peuples
phénomène(*) :phénomène,phénomènes
philosophe(*) :philosophe,philosophes
philosophique(*) :philosophique,philosophiques
picture(*) :picture,pictures
plainte(*) :plainte,plaintes
planet(*) :planet,planets
player(*) :player,players
poétique(*) :poétique,poétiques
polémique(*) :polémique,polémiques
politique(*) :politique,politiques
populaire(*) :populaire,populaires
population(*) :population,populations
populiste(*) :populiste,populistes
pornographique(*)
:pornographique,pornographiques
portrait(*) :portrait,portraits
position(*) :position,positions
possible(*) :possible,possibles
potchvenniki(*) :potchvenniki,potchvennikis
pouvoir(*) :pouvoir,pouvoirs

pratique(*) :pratique,pratiquer,pratiques
précis(*) :précis,précise,précisé
précurseur(*) :précurseur,précurseurs
preference(*) :preference,reference
prematch(*) :prematch,rematch
premier(*) :premier,premiers
première(*) :première,premières
prenant(*) :prenant,prenante
prendre(*) :prendre,rendre
présent(*) :présent,présente,présenté
présentent(*) :présentent,représentent
président(*) :président,présidents,résident
présumer(*) :présumer,résumer
principal(*) :principal,principale,principales
principe(*) :principe,principes
principle(*) :principle,principles
prison(*) :prison,prisons
problème(*) :problème,problèmes
procédé(*) :procédé,procédés
proche(*) :proche,proches
produce(*) :produce,produced
product(*) :product,products
production(*) :production,reproduction
produit(*) :produit,produite,produits
profilé(*) :profilé,profilés
profond(*) :profond,profonds
profondeur(*) :profondeur,profondeurs
progres(*) :progres,progress
project(*) :project,projects
promotion(*)
:promotion,promotional,promotions
propose(*) :propose,proposer
proposition(*) :proposition,propositions
propre(*) :propre,propres
propriétaire(*) :propriétaire,propriétaires
prosecute(*) :prosecute,prosecuted
prosecutor(*) :prosecutor,prosecutors
prouve(*) :prouve,prouver
provide(*) :provide,provides
psychologique(*)
:psychologique,psychologiques
puissance(*) :puissance,puissances
qualifié(*) :qualifié,qualifiée
qualité(*) :qualité,qualités
québécois(*) :québécois,québécoise
question(*) :question,questions
racheté(*) :racheté,rachetés
racine(*) :racine,racines
raison(*) :raison,raisons
rapprochement(*)
:rapprochement,rapprochements
razzie(*) :razzie,razzies
réactionnaire(*) :réactionnaire,réactionnaires
réaliste(*) :réaliste,réalistes
réalité(*) :réalité,réalités
reason(*) :reason,reasons
recensement(*) :recensement,recensements
recherche(*) :recherche,recherches
recognize(*) :recognize,recognized
reconnu(*) :reconnu,reconnue
record(*) :record,records
recueil(*) :recueil,recueils
récurrence(*) :récurrence,récurrences
redécouverte(*) :redécouverte,redécouvertes
réelle(*) :réelle,réelles
réforme(*) :réforme,réformes
refugee(*) :refugee,refugees
regard(*) :regard,regarde,regards
région(*) :région,régions
registre(*) :registre,registres
relation(*) :relation,relations
relative(*) :relative,relatives
release(*) :release,released,releases
religieuse(*) :religieuse,religieuses
religion(*) :religion,religions
répertoire(*) :répertoire,répertoires
report(*) :report,reports
reporter(*) :reporter,reporters
représentant(*) :représentant,représentants
representative(*) :representative,representatives
resource(*) :resource,resources
respect(*) :respect,respecté,respects
résultat(*) :résultat,résultats
retourne(*) :retourne,retourner
retrait(*) :retrait,retraite
retrouve(*) :retrouve,retrouver
réussite(*) :réussite,réussites
révélé(*) :révélé,révélés
révolutionnaire(*)
:révolutionnaire,révolutionnaires
robert(*) :robert,roberto
rocket(*) :rocket,rockets
roster(*) :roster,rosters
russia(*) :russia,russian
rwandan(*) :rwandan,rwandans
salvadorienne(*) :salvadorienne,salvadoriennes
sample(*) :sample,samples
satellite(*) :satellite,satellites
savoir(*) :savoir,savoirs
school(*) :school,schools
science(*) :science,sciences
scientist(*) :scientist,scientists
second(*) :second,seconde,seconds

séduit(*) :séduit,séduits
semaine(*) :semaine,semaines
sénateur(*) :sénateur,sénateurs
series(*) :series,xseries
serveur(*) :serveur,serveurs
service(*) :service,services
sharpe(*) :sharpe,sharper
siècle(*) :siècle,siècles
signifiant(*) :signifiant,signifiante
similaire(*) :similaire,similaires
simple(*) :simple,simples
slavophile(*) :slavophile,slavophiles
social(*) :social,sociale
socialiste(*) :socialiste,socialistes
société(*) :société,sociétés
source(*) :source,sources
soviet(*) :soviet,soviets
soviétique(*) :soviétique,soviétiques
spéculation(*) :spéculation,spéculations
spirituel(*) :spirituel,spirituelle,spirituels
stalinien(*) :stalinien,stalinienne
strike(*) :strike,strikes
structure(*) :structure,structurel
student(*) :student,students
successful(*) :successful,successfully
suggest(*) :suggest,suggests
superficiel(*)
:superficiel,superficielle,superficiels
support(*) :support,supports
supporter(*) :supporter,supporters
surhumain(*)
:surhumain,surhumaines,surhumains
survenu(*) :survenu,survenue
symbol(*) :symbol,symbole
system(*) :system,systems
système(*) :système,systèmes
technologie(*) :technologie,technologies
télécom(*) :télécom,télécoms
tendance(*) :tendance,tendances
tension(*) :tension,tensions
terminé(*) :terminé,terminée
terminologique(*)
:terminologique,terminologiques
terrorist(*) :terrorist,terroriste,terroristes
thalassocratique(*)
:thalassocratique,thalassocratiques
theater(*) :theater,theaters
théologie(*) :théologie,théologien
théorie(*) :théorie,théories
théorique(*) :théorique,théoriques
thousand(*) :thousand,thousands
tomorrow(*) :tomorrow,tomorrows
traditionaliste(*) :traditionaliste,traditionalistes
traditionnel(*)
:traditionnel,traditionnelle,traditionnelles,traditionnels
traduit(*) :traduit,traduite
transformation(*) :transformation,transformations
transition(*) :transition,transitional
treatment(*) :treatment,treatments
trouble(*) :trouble,troubles
typique(*) :typique,typiques
ultérieur(*) :ultérieur,ultérieure
unanimous(*) :unanimous,unanimously
universel(*) :universel,universelle
université(*) :université,university
usager(*) :usager,usagers
utilisateur(*) :utilisateur,utilisateurs
utilisé(*) :utilisé,utilisés
utopiste(*) :utopiste,utopistes
valeur(*) :valeur,valeurs
vasiliev(*) :vasiliev,vasilievs
vérité(*) :vérité,vérités
vernaculaire(*) :vernaculaire,vernaculaires
victime(*) :victime,victimes
victor(*) :victor,victory
victoria(*) :victoria,victorian
vieille(*) :vieille,vieilles
vilain(*) :vilain,vilains
violation(*) :violation,violations
violence(*) :violence,violences
violent(*) :violent,violente
visage(*) :visage,visages
vision(*) :vision,visions
visionnaire(*) :visionnaire,visionnaires
warner(*) :warner,warners
weapon(*) :weapon,weapons
william(*) :william,williams
wonderful(*) :wonderful,wonderfully
zambia(*) :zambia,zambian

Étude sur les liens entre les Bi-grammes avec un seuil de 75%	
absolu(*) :absolu,absolue abstraction(*) :abstraction,abstractions abstrait(*) :abstrait,abstraite,abstraites académique(*) :académique,académiques accidental(*) :accidental,occidental accompagné(*) :accompagné,accompagnée accord(*) :accord,accorde,accordé,accords accueilli(*) :accueilli,accueillir accumule(*) :accumule,accumuler acheter(*) :acheter,racheter acteur(*) :acteur,facteur actif(*) :actif,actifs action(*) :action,actions activité(*) :activité,activités actuelle(*) :actuelle,factuelles added(*) :added,padded admire(*) :admire,admired advance(*) :advance,advanced,advances adversaire(*) :adversaire,adversaires advocate(*) :advocate,advocates affair(*) :affair,affaire affirme(*) :affirme,affirmer afghan(*) :afghan,afghans africa(*) :africa,african aggression(*) :aggression,agressions aggressive(*) :aggressive,agressive aires(*) :aires,naires ajoute(*) :ajoute,ajouter album(*) :album,albums aliénant(*) :aliénant,aliénantes allait(*) :allait,fallait allemand(*) :allemand,allemande,allemandes,allemands allow(*) :allow,allows ambition(*) :ambition,ambitions america(*) :america,american américain(*) :américain,américaine,américains amérique(*) :amérique,amériques amoralisme(*) :amoralisme,moralisme amorce(*) :amorce,amorcer analyste(*) :analyste,analystes anarchiste(*) :anarchiste,anarchistes ancien(*) :ancien,anciens,ancient animé(*) :animé,animée,animés année(*) :année,années announce(*) :announce,announced antisémite(*) :antisémite,antisémites aperçu(*) :aperçu,perçu apparaissent(*) :apparaissent,paraissent	approach(*) :approach,approached approche(*) :approche,approches,rapproche,rapprocher approve(*) :approve,approved archaïsant(*) :archaïsant,archaïsante,archaïsants argument(*) :argument,arguments aristocrate(*) :aristocrate,aristocrates arrêt(*) :arrêt,arrête,arrêté article(*) :article,articles artificial(*) :artificial,artificially asiatique(*) :asiatique,asiatiques assistant(*) :assistant,assistants assure(*) :assure,assurer astronaut(*) :astronaut,astronauts astronomer(*) :astronomer,astronomers athée(*) :athée,athées atlantiste(*) :atlantiste,atlantistes atmosphere(*) :atmosphere,atmospheres attitude(*) :attitude,attitudes attribué(*) :attribué,attribués augure(*) :augure,augurer australia(*) :australia,australie auteur(*) :auteur,auteurs autorité(*) :autorité,autorités avantages(*) :avantages,davantage avocat(*) :avocat,avocats award(*) :award,awards badger(*) :badger,badgers ballade(*) :ballade,ballades bancaire(*) :bancaire,bancaires basket(*) :basket,baskets bataille(*) :bataille,batailles begin(*) :begin,begins being(*) :being,beings belgian(*) :belgian,belgians believe(*) :believe,believed,believes benefit(*) :benefit,benefits besoin(*) :besoin,besoins birth(*) :birth,births blanc(*) :blanc,blancs blanche(*) :blanche,blanches blessé(*) :blessé,blessés bolchevique(*) :bolchevique,bolcheviques bolchevisme(*) :bolchevisme,bolchevismes bourgeois(*) :bourgeois,bourgeoisie

bourse(*) :bourse,bourses
boxer(*) :boxer,boxers
bright(*) :bright,right
budget(*) :budget,budgets
cadre(*) :cadre,cadres
capacité(*) :capacité,incapacité
capital(*) :capital,capitale
capitaliste(*) :capitaliste,capitalistes
carat(*) :carat,carats
carte(*) :carte,cartes
cassation(*) :cassation,passation
cavalier(*) :cavalier,cavaliers
céano(*) :céano,océano
célèbre(*) :célèbre,célèbres
center(*) :center,enter
central(*) :central,centrale
centralisation(*)
:centralisation,décentralisation
centre(*) :centre,centres
cercle(*) :cercle,cercles
challenger(*) :challenger,challenges
champ(*) :champ,champs
champion(*) :champion,champions
chance(*) :chance,chances
change(*)
:change,changed,changer,changes,échange
changement(*) :changement,changements
chapel(*) :chapel,chapela
character(*) :character,characters
charge(*) :charge,charged,charges
chemical(*) :chemical,chemicals
chéquier(*) :chéquier,chéquiers
chimère(*) :chimère,chimères
choose(*) :choose,chose
chrétien(*) :chrétien,chrétiens,chrétienté
christoph(*) :christoph,christopher
cidentale(*) :cidentale,occidentale
civilisation(*) :civilisation,civilisations
claim(*) :claim,claims
clair(*) :clair,claire
class(*) :class,classe,classé
classique(*) :classique,classiques
client(*) :client,clients
climat(*) :climat,climate
close(*) :close,closer
cluster(*) :cluster,clusters
colle(*) :colle,collet
colonial(*) :colonial,coloniale
combat(*) :combat,combats
commence(*)
:commence,commencent,commencer
commission(*) :commission,commissions
commu(*) :commu,commun
communications(*)
:communications,telecommunications,télécommunications
communiste(*) :communiste,communistes
compare(*) :compare,compared,comparer
compétence(*) :compétence,compétences
complaint(*) :complaint,complaints
complet(*) :complet,complete
comprend(*) :comprend,comprendre
compte(*) :compte,compter
concentrée(*) :concentrée,concentrées
conception(*) :conception,conceptions
concret(*) :concret,concrete
condition(*) :condition,conditions
conduit(*) :conduit,conduite
conflit(*) :conflit,conflits
connaissance(*) :connaissance,connaissances
connaissent(*) :connaissent,reconnaissent
connaître(*) :connaître,reconnaître
connect(*) :connect,connecté
connotation(*) :connotation,connotations
connu(*) :connu,connus
conscient(*) :conscient,consciente,conscients
conséquence(*) :conséquence,conséquences
conservateur(*) :conservateur,conservateurs
conservatrice(*) :conservatrice,conservatrices
conserve(*) :conserve,conserver
consider(*) :consider,considered
considérée(*) :considérée,considérés
constante(*) :constante,constantes
constate(*) :constate,constatent,constater
constitue(*) :constitue,constituer
constitution(*) :constitution,constitutionnel
constitutionnelle(*) :constitutionnelle,constitutionnelles
construction(*) :construction,constructions
construit(*) :construit,déconstruit
contender(*) :contender,contenders
contextualisation(*)
:contextualisation,décontextualisation
continent(*) :continent,continental
continue(*) :continue,continued,continuer
contraceptive(*) :contraceptive,contraceptives
contradictoire(*) :contradictoire,contradictoires
contrainte(*) :contrainte,contraints
contrast(*) :contrast,contraste
controverse(*) :controverse,controversy
conviction(*) :conviction,convictions
corse(*) :corse,corses
courant(*) :courant,courante,courants
courrier(*) :courrier,courriers

cours(*) :cours,course
court(*) :court,courte,courts
couvre(*) :couvre,ouvre
crates(*) :crates,rates
create(*) :create,created
créateur(*) :créateur,créateurs
crime(*) :crime,crimes
critique(*) :critique,critiques
culture(*) :culture,culturel,cultures
culturelle(*) :culturelle,culturelles
cyclique(*) :cyclique,cycliques
danger(*) :danger,dangers
danilevski(*) :danilevski,danilevsky
début(*) :début,débuté
décadente(*) :décadente,décadents
decision(*) :decision,decisioned,decisions
décision(*) :décision,indécision
découvert(*) :découvert,découverte
découvrir(*) :découvrir,redécouvrir
décrit(*) :décrit,écrit
défend(*) :défend,défendu
défini(*) :défini,définir
dégage(*) :dégage,dégager
delegation(*) :delegation,delegations
demand(*) :demand,demanda,demande,demandé
demander(*) :demander,demandera
demeure(*) :demeure,demeurer
démocrates(*) :démocrates,mocrates
démocratique(*) :démocratique,démocratiques
dénoncé(*) :dénoncé,énoncé
dépendance(*) :dépendance,indépendance
déplacement(*) :déplacement,placement
déploiement(*) :déploiement,déploient
déposée(*) :déposée,déposées
désabusé(*) :désabusé,désabusés
design(*) :design,designs
détail(*) :détail,détails
détenu(*) :détenu,détenue
determine(*) :determine,determined
détournant(*) :détournant,tournant
development(*) :development,developments
développe(*) :développe,développent,développer
devenu(*) :devenu,devenue,devenus
diamond(*) :diamond,diamonds
didactique(*) :didactique,didactiques
différent(*)
:différent,différente,différentes,différents
difficile(*) :difficile,difficiles
difficult(*) :difficult,difficulté,difficultés

dimension(*) :dimension,dimensions
direct(*) :direct,directe,directs
director(*) :director,directors
dirigé(*) :dirigé,dirigée
dirigeant(*) :dirigeant,dirigeants
disciple(*) :disciple,disciples
disease(*) :disease,diseases
disparu(*) :disparu,disparus
disponible(*) :disponible,disponibles
disposition(*) :disposition,dispositions
distance(*) :distance,distances
distincte(*) :distincte,distinctes
distributor(*) :distributor,distributors
diversité(*) :diversité,diversity
dizaine(*) :dizaine,dizaines
doctrine(*) :doctrine,doctrines
document(*) :document,documents
dominate(*) :dominate,dominated
donne(*) :donne,donner
drive(*) :drive,driven,driver,drives
droit(*) :droit,droite,droits
earning(*) :earning,learning
école(*) :école,écoles
économie(*) :économie,économies
économique(*) :économique,économiques
écrivain(*) :écrivain,écrivains
éditée(*) :éditée,éditées
éditeur(*) :éditeur,éditeurs
édition(*) :édition,éditions
education(*) :education,éducation
effect(*) :effect,effects
effet(*) :effet,effets
effort(*) :effort,efforts
église(*) :église,églises
egypt(*) :egypt,egypte
eight(*) :eight,eighth,weight
élabore(*) :élabore,élaborer
élaboré(*) :élaboré,élaborée
électeurs(*) :électeurs,lecteurs
élection(*) :élection,élections
électoral(*) :électoral,électorale
électronique(*) :électronique,électroniques
éloigne(*) :éloigne,éloigner
ement(*) :ement,tement,vement
employé(*) :employé,employés
employee(*) :employee,employees
encontre(*) :encontre,rencontre
encyclopédique(*)
:encyclopédique,encyclopédiques

encyclopédiste(*)
:encyclopédiste,encyclopédistes
énergie(*) :énergie,énergies,nergie
enfant(*) :enfant,enfants
engage(*) :engage,engager
engagé(*) :engagé,engagée
engagement(*) :engagement,engagent
énigmatique(*) :énigmatique,énigmatiques
ennemi(*) :ennemi,ennemis
enquête(*) :enquête,enquêtes
enregistré(*) :enregistré,enregistrées,enregistrés
enseignant(*) :enseignant,enseignants
entendu(*) :entendu,entendue,entendus
entraîne(*) :entraîne,entraîneur
entreprise(*) :entreprise,entreprises
énumératif(*) :énumératif,énumératifs
énumération(*) :énumération,énumérations
environment(*) :environment,environmental
envoyer(*) :envoyer,renvoyer
équilibre(*) :équilibre,équilibrent
équipe(*) :équipe,équipes
équivalent(*) :équivalent,équivalente
erreur(*) :erreur,erreurs
eserver(*) :eserver,eservers
espace(*) :espace,espaces,space
estate(*) :estate,state
estime(*) :estime,estimer
établir(*) :établir,rétablir
étayée(*) :étayée,étayées
ethnie(*) :ethnie,ethnies
ethnique(*) :ethnique,ethniques
étiquette(*) :étiquette,étiquetté
étrange(*) :étrange,trange
étude(*) :étude,études
eurasiatique(*) :eurasiatique,eurasiatiques
eurasie(*) :eurasie,eurasien
eurasiste(*) :eurasiste,eurasistes
européen(*) :européen,européenne,européens
eventual(*) :eventual,eventually
évident(*) :évident,évidente
évolution(*)
:évolution,évolutions,révolution,volution
except(*) :except,excepté
exchanged(*) :exchanged,exchanges
exemple(*) :exemple,exemples
exhibitor(*) :exhibitor,exhibitors
exile(*) :exile,exiles
existe(*) :existe,exister
experience(*)
:experience,experienced,experiences
expérience(*) :expérience,expériences
explained(*) :explained,unexplained
explique(*) :explique,expliquer
exploite(*) :exploite,exploitent,exploiter
expression(*) :expression,pression
exprime(*) :exprime,exprimer
extend(*) :extend,extends
extérieur(*) :extérieur,extérieure,extérieures
extra(*) :extra,extras
extradition(*) :extradition,tradition
extraordinaire(*)
:extraordinaire,extraordinairement
façon(*) :façon,façons
faible(*) :faible,faibles
famille(*) :famille,familles
fasciste(*) :fasciste,fascistes
favor(*) :favor,favori
fellow(*) :fellow,fellows
female(*) :female,females
fence(*) :fence,fences
fiable(*) :fiable,fiables
field(*) :field,fields
fight(*) :fight,fights
figure(*) :figure,figures
fille(*) :fille,filles
filmmaker(*) :filmmaker,filmmakers
final(*) :final,finals
finding(*) :finding,findings
flight(*) :flight,light
fonction(*) :fonction,fonctions
fonctionnement(*) :fonctionnement,fonctionnent
force(*) :force,forces
formation(*) :formation,information
forme(*) :forme,former,formes
formelle(*) :formelle,formelles
forte(*) :forte,fortes
fourni(*) :fourni,fournir,fournit
fragment(*) :fragment,fragments
français(*) :français,française
francophone(*) :francophone,francophones
frappant(*) :frappant,frappantes
futur(*) :futur,future
futuriste(*) :futuriste,futuristes
gauchiste(*) :gauchiste,gauchistes
généra(*) :généra,général
generation(*) :generation,generations
géopolitique(*) :géopolitique,géopolitiques
géopolitiquement(*)
:géopolitiquement,politiquement
george(*) :george,georges
glissement(*) :glissement,glissements
global(*) :global,globale
glossaire(*) :glossaire,glossaires

gouvernement(*)
:gouvernement,gouvernemental,gouvernent
government(*) :government,governmental
grand(*) :grand,grande,grandi,grands
great(*) :great,greats
ground(*) :ground,round
group(*) :group,groupe,groups
groupements(*) :groupements,regroupements
habillement(*) :habillement,habillements
handle(*) :handle,handles
happen(*) :happen,happens
health(*) :health,healthy
héroïque(*) :héroïque,héroïques
heure(*) :heure,heures
histoire(*) :histoire,histoires
historien(*) :historien,historiens
historique(*) :historique,historiques
homme(*) :homme,hommes
honneur(*) :honneur,honneurs
honor(*) :honor,honoré
hospital(*) :hospital,hospitals
house(*) :house,housed
humain(*) :humain,humains
human(*) :human,humans
humaniste(*) :humaniste,humanistes
idéaliste(*) :idéaliste,idéalistes
identified(*) :identified,identifier
identitaire(*) :identitaire,identitaires
idéologique(*) :idéologique,idéologiques
idéologue(*) :idéologue,idéologues
illusion(*) :illusion,illusions
image(*) :image,images
immédiat(*) :immédiat,immédiate,immédiates
impérial(*) :impérial,impériale,impériales
important(*)
:important,importante,importantes,importants
importe(*) :importe,imported
impose(*) :impose,imposed,imposer
impossibilité(*)
:impossibilité,possibilité,possibilités
impossible(*) :impossible,possible
imprécatoire(*) :imprécatoire,imprécatoires
improbable(*) :improbable,probable
incapable(*) :incapable,incapables
inchangé(*) :inchangé,inchangées
include(*) :include,included,includes
increase(*) :increase,increased
indépendante(*) :indépendante,indépendantes
indication(*) :indication,indications
individu(*) :individu,individual,individuel
industrie(*) :industrie,industriel
inégalité(*) :inégalité,inégalités
influence(*)
:influence,influencé,influenced,influencer
informatique(*) :informatique,informatiques
inspiration(*) :inspiration,inspirational
inspire(*) :inspire,inspired
instinct(*) :instinct,instincts
institut(*) :institut,institute,instituts
institutionnelle(*) :institutionnelle,institutionnels
instrument(*) :instrument,instruments
insuffisamment(*) :insuffisamment,suffisamment
insuffisante(*) :insuffisante,suffisante
intellectuel(*)
:intellectuel,intellectuelle,intellectuelles,intellectuels
intended(*) :intended,unintended
intention(*) :intention,intentions
intéresse(*) :intéresse,intéressé,intéressent
intérêt(*) :intérêt,intérêts
intérieur(*) :intérieur,intérieure,intérieures
international(*) :international,internationale
internationaliste(*)
:internationaliste,internationalistes,ternationaliste
internaute(*) :internaute,internautes
interprété(*) :interprété,interprétées
intervention(*) :intervention,interventions
introduce(*) :introduce,introduced
inventaire(*) :inventaire,inventaires
invisible(*) :invisible,invisibles
involve(*) :involve,involved
isolé(*) :isolé,isolée
israel(*) :israel,israeli
issue(*) :issue,issued,issues
jeune(*) :jeune,jeunes
jewellers(*) :jewellers,jewellery
judge(*) :judge,judges
jugée(*) :jugée,jugées
juive(*) :juive,juives
khrouchtchev(*) :khrouchtchev,khrouchtchevien
kirievski(*) :kirievski,kirievsky
lâche(*) :lâche,lâcher
laisse(*) :laisse,laisser
lancé(*) :lancé,lancée
langue(*) :langue,langues
latin(*) :latin,latine,latins
laying(*) :laying,playing
leader(*) :leader,leaders
lennnox(*) :lennnox,lennox

lettre(*) :lettre,lettres
level(*) :level,levels
libéral(*) :libéral,libérale
lightly(*) :lightly,slightly
liste(*) :liste,listen,listes
livre(*) :livre,livres,livret
logiciel(*) :logiciel,logiciels
logique(*) :logique,logiques
lointain(*) :lointain,lointaines
lutionnaires(*) :lutionnaires,volutionnaires
lyakhovic(*) :lyakhovic,lyakhovich
machine(*) :machine,machines
majeur(*) :majeur,majeure
maladie(*) :maladie,maladies
manager(*) :manager,managers
manifeste(*) :manifeste,manifestent,manifester
march(*) :march,marche,marché
maritime(*) :maritime,maritimes
marxiste(*) :marxiste,marxistes
matière(*) :matière,matières
mécanisme(*) :mécanisme,mécanismes
medicine(*) :medicine,medicines
médiéval(*) :médiéval,médiévales
member(*) :member,members
menace(*) :menace,menaces
mesure(*) :mesure,mesurer
metal(*) :metal,metals
métier(*) :métier,métiers
milieu(*) :milieu,milieux
militaire(*) :militaire,militaires
milliard(*) :milliard,milliards
million(*) :million,millions
mineur(*) :mineur,mineurs
ministre(*) :ministre,ministres
minorité(*) :minorité,minorités
minute(*) :minute,minutes
model(*) :model,models
modèle(*) :modèle,modèles
moderne(*) :moderne,modernes
modes(*) :modes,modest
modification(*) :modification,modifications
monarchiste(*) :monarchiste,monarchistes
mondial(*) :mondial,mondiale
mongole(*) :mongole,mongoles
month(*) :month,months
moral(*) :moral,morale
motif(*) :motif,motifs
mouvement(*) :mouvement,mouvements
municipal(*) :municipal,municipales
musical(*) :musical,musicale
musique(*) :musique,musiques
musulman(*) :musulman,musulmans
mystique(*) :mystique,mystiques
narodniki(*) :narodniki,narodnikis
natio(*) :natio,nation
national(*) :national,nationale,nationales
nationaliste(*) :nationaliste,nationalistes
nature(*) :nature,naturel
nécessaire(*) :nécessaire,nécessaires
nigeria(*) :nigeria,nigerian
night(*) :night,nights
noble(*) :noble,nobles
normal(*) :normal,normale
nostalgique(*) :nostalgique,nostalgiques
numéro(*) :numéro,numéros
objectif(*) :objectif,objectifs
objet(*) :objet,objets
obligatoire(*) :obligatoire,obligatoires
observation(*) :observation,observations
observe(*) :observe,observer
occidentaliste(*) :occidentaliste,occidentalistes
oeuvre(*) :oeuvre,oeuvres
office(*) :office,officer,offices
official(*) :official,officials
offre(*) :offre,offrez
olympic(*) :olympic,olympics
onomasiologique(*)
:onomasiologique,onomasiologiques
opérateur(*) :opérateur,opérateurs
opposant(*) :opposant,opposants
oppose(*) :oppose,opposed
opposition(*) :opposition,position
option(*) :option,options
orange(*) :orange,range
orbit(*) :orbit,orbits
organisation(*) :organisation,organisations
organize(*) :organize,organized,organizer
orient(*) :orient,oriente
orientale(*) :orientale,orientales
orientation(*) :orientation,orientations
original(*) :original,originale
orthodoxe(*) :orthodoxe,orthodoxes
oscar(*) :oscar,oscars
oubli(*) :oubli,oublié
ouvert(*) :ouvert,ouverte
ouvrage(*) :ouvrage,ouvrages
palestinian(*) :palestinian,palestinians
paper(*) :paper,papers
parallèle(*) :parallèle,parallèles
parent(*) :parent,parenté,parents
parle(*) :parle,parler,parles
partenaire(*) :partenaire,partenaires

parti(*) :parti,partie,partir
particularité(*) :particularité,particularités
particulier(*) :particulier,particuliers
partisan(*) :partisan,partisans
passe(*) :passe,passed,passer,passes
passé(*) :passé,passés
passion(*) :passion,passions
patient(*) :patient,patients
patriarcal(*) :patriarcal,patriarcale
patriotic(*) :patriotic,patriotica
patriotique(*) :patriotique,patriotiques
pense(*) :pense,penser
pensée(*) :pensée,pensées
période(*) :période,périodes
personnalité(*) :personnalité,personnalités
personne(*) :personne,personnel,personnes
petit(*) :petit,petite,petits
pétition(*) :pétition,répétition
peuple(*) :peuple,peuples
phénomène(*) :phénomène,phénomènes
philosophe(*) :philosophe,philosophes
philosophique(*) :philosophique,philosophiques
picture(*) :picture,pictures
pièce(*) :pièce,pièces
placé(*) :placé,placée
plainte(*) :plainte,plaintes
planet(*) :planet,planets
plant(*) :plant,plants
player(*) :player,players
plein(*) :plein,pleine
poème(*) :poème,poèmes
poète(*) :poète,poètes
poétique(*) :poétique,poétiques
point(*) :point,points
polémique(*) :polémique,polémiques
politique(*) :politique,politiques
populaire(*) :populaire,populaires
population(*) :population,populations
populiste(*) :populiste,populistes
pornographique(*)
:pornographique,pornographiques
porte(*) :porte,porter,portes
porté(*) :porté,portée
portrait(*) :portrait,portraits
positive(*) :positive,positively
potchvenniki(*) :potchvenniki,potchvennikis
pound(*) :pound,pounds
pouvoir(*) :pouvoir,pouvoirs
power(*) :power,powers
pratique(*) :pratique,pratiquer,pratiques

précis(*) :précis,précise,précisé
précurseur(*) :précurseur,précurseurs
preference(*) :preference,reference
prematch(*) :prematch,rematch
premier(*) :premier,premiers
première(*) :première,premières
prenant(*) :prenant,prenante
prendre(*) :prendre,rendre
prennent(*) :prennent,reprennent
présent(*) :présent,présente,présenté
présentent(*) :présentent,représentent
présenter(*) :présenter,présentes
président(*) :président,présidents,résident
press(*) :press,presse
présumer(*) :présumer,résumer
prévisions(*) :prévisions,révision
principal(*) :principal,principale,principales
principe(*) :principe,principes
principle(*) :principle,principles
prison(*) :prison,prisons
problème(*) :problème,problèmes
procédé(*) :procédé,procédés
prochain(*) :prochain,prochaines
proche(*) :proche,proches
produce(*) :produce,produced
product(*) :product,products
production(*) :production,reproduction
produit(*) :produit,produite,produits
profilé(*) :profilé,profilés
profond(*) :profond,profonds
profondeur(*) :profondeur,profondeurs
progres(*) :progres,progress
project(*) :project,projects
promotion(*)
:promotion,promotional,promotions
propose(*) :propose,proposer
proposition(*) :proposition,propositions
propre(*) :propre,propres
propriétaire(*) :propriétaire,propriétaires
prosecute(*) :prosecute,prosecuted
prosecutor(*) :prosecutor,prosecutors
prouve(*) :prouve,prouver
prove(*) :prove,proven
provide(*) :provide,provides
prusse(*) :prusse,russe
psychologique(*)
:psychologique,psychologiques
puissance(*) :puissance,puissances
qualifié(*) :qualifié,qualifiée
qualité(*) :qualité,qualités
québécois(*) :québécois,québécoise
question(*) :question,questioned,questions

races(*) :races,traces
racheté(*) :racheté,rachetés
racine(*) :racine,racines
radio(*) :radio,radios
raison(*) :raison,raisons
rapprochement(*)
:rapprochement,rapprochements
razzie(*) :razzie,razzies
réactionnaire(*) :réactionnaire,réactionnaires
réaliste(*) :réaliste,réalistes
réalité(*) :réalité,réalités
reason(*) :reason,reasons
recensement(*) :recensement,recensements
recherche(*) :recherche,recherches
recognize(*) :recognize,recognized
reconnu(*) :reconnu,reconnue
record(*) :record,records
recueil(*) :recueil,recueils
récurrence(*) :récurrence,récurrences
redécouverte(*) :redécouverte,redécouvertes
réelle(*) :réelle,réelles
réforme(*) :réforme,réformes
refugee(*) :refugee,refugees
regard(*) :regard,regarde,regards
région(*) :région,régions
registre(*) :registre,registres
reign(*) :reign,reigns
relation(*) :relation,relations
relative(*) :relative,relatively,relatives
release(*) :release,released,releases
religieuse(*) :religieuse,religieuses
religion(*) :religion,religions
remarquer(*) :remarquer,remarques
remember(*) :remember,remembered
repeated(*) :repeated,repeatedly
répertoire(*) :répertoire,répertoires
report(*) :report,reports
reporter(*) :reporter,reporters
représentant(*) :représentant,représentants
representative(*) :representative,representatives
resolution(*) :resolution,solution
resource(*) :resource,resources
respect(*) :respect,respecté,respects
ressenti(*) :ressenti,ressenties
résultat(*) :résultat,résultats
retourne(*) :retourne,retourner
retrait(*) :retrait,retraite
retrouve(*) :retrouve,retrouvent,retrouver
réussite(*) :réussite,réussites
révélé(*) :révélé,révélés
révolutionnaire(*)
:révolutionnaire,révolutionnaires
revue(*) :revue,revues
robert(*) :robert,roberto
rocket(*) :rocket,rockets
roman(*) :roman,romans
roster(*) :roster,rosters
rouge(*) :rouge,rouges
russia(*) :russia,russian
rwandan(*) :rwandan,rwandans
salvadorienne(*) :salvadorienne,salvadoriennes
sample(*) :sample,samples
satellite(*) :satellite,satellites
savoir(*) :savoir,savoirs
school(*) :school,schools
science(*) :science,sciences
scientist(*) :scientist,scientists
score(*) :score,scores
second(*) :second,seconde,seconds
séduit(*) :séduit,séduits
semaine(*) :semaine,semaines
sénateur(*) :sénateur,sénateurs
serge(*) :serge,sergei
série(*) :série,séries
series(*) :series,xseries
serveur(*) :serveur,serveurs
service(*) :service,services
seule(*) :seule,seules
shape(*) :shape,shaped
sharpe(*) :sharpe,sharper
siècle(*) :siècle,siècles
siège(*) :siège,sièges
signifiant(*) :signifiant,signifiante
similaire(*) :similaire,similaires
simon(*) :simon,simone
simple(*) :simple,simples
situe(*) :situe,situer
slave(*) :slave,slaves
slavophile(*) :slavophile,slavophiles
social(*) :social,sociale
socialiste(*) :socialiste,socialistes
société(*) :société,sociétés
sorte(*) :sorte,sortes
souhaite(*) :souhaite,souhaitent
source(*) :source,sources
soviet(*) :soviet,soviets
soviétique(*) :soviétique,soviétiques
spéculation(*) :spéculation,spéculations
spirituel(*) :spirituel,spirituelle,spirituels
stade(*) :stade,stades
stalinien(*) :stalinien,stalinienne
stand(*) :stand,stands
stone(*) :stone,stones

strike(*) :strike,strikes
structure(*) :structure,structurel
student(*) :student,students
style(*) :style,styles
successful(*) :successful,successfully
suggest(*) :suggest,suggests
suite(*) :suite,suited
suivi(*) :suivi,suivit
superficiel(*) :superficiel,superficielle,superficiels
support(*) :support,supports
supporter(*) :supporter,supporters
surhumain(*) :surhumain,surhumaines,surhumains
survenu(*) :survenu,survenue
swords(*) :swords,words
symbol(*) :symbol,symbole
system(*) :system,systems
système(*) :système,systèmes
table(*) :table,tables
technologie(*) :technologie,technologies
télécom(*) :télécom,télécoms
tendance(*) :tendance,tendances
tension(*) :tension,tensions
tenté(*) :tenté,tentée
terme(*) :terme,termes
terminé(*) :terminé,terminée
terminologique(*) :terminologique,terminologiques
terre(*) :terre,terres
terrorist(*) :terrorist,terroriste,terroristes
texte(*) :texte,textes
thalassocratique(*)
:thalassocratique,thalassocratiques
theater(*) :theater,theaters
thème(*) :thème,thèmes
théologie(*) :théologie,théologien
théorie(*) :théorie,théories
théorique(*) :théorique,théoriques
thèse(*) :thèse,thèses
thing(*) :thing,things
think(*) :think,thinks
third(*) :third,thirds
thousand(*) :thousand,thousands
title(*) :title,titles
titre(*) :titre,titres
tomorrow(*) :tomorrow,tomorrows
total(*) :total,totale
traditionaliste(*) :traditionaliste,traditionalistes
traditionnel(*)
:traditionnel,traditionnelle,traditionnelles,traditionnels
traduit(*) :traduit,traduite
traffic(*) :traffic,trafic
transformation(*)
:transformation,transformations
transition(*) :transition,transitional
travaille(*) :travaille,travaillé
treatment(*) :treatment,treatments
trial(*) :trial,trials
trouble(*) :trouble,troubles
typique(*) :typique,typiques
ultérieur(*) :ultérieur,ultérieure
ultra(*) :ultra,ultras
unanimous(*) :unanimous,unanimously
universel(*) :universel,universelle
université(*) :université,university
usage(*) :usage,usager
utilisateur(*) :utilisateur,utilisateurs
utilisé(*) :utilisé,utilisés
utopiste(*) :utopiste,utopistes
vague(*) :vague,vagues
valeur(*) :valeur,valeurs
valorisée(*) :valorisée,valorisés
vasiliev(*) :vasiliev,vasilievs
veine(*) :veine,veines
vérité(*) :vérité,vérités
vernaculaire(*) :vernaculaire,vernaculaires
victime(*) :victime,victimes
victor(*) :victor,victory
victoria(*) :victoria,victorian,victorians
vieille(*) :vieille,vieilles
vilain(*) :vilain,vilains
violation(*) :violation,violations
violence(*) :violence,violences
violent(*) :violent,violente
visage(*) :visage,visages
vision(*) :vision,visions
visionnaire(*) :visionnaire,visionnaires
warner(*) :warner,warners
weapon(*) :weapon,weapons
william(*) :william,williams
wonderful(*) :wonderful,wonderfully
youth(*) :youth,youths
zambia(*) :zambia,zambian
zapadnik(*) :zapadnik,zapadnikis

Annexe B

Arbres de décision générés

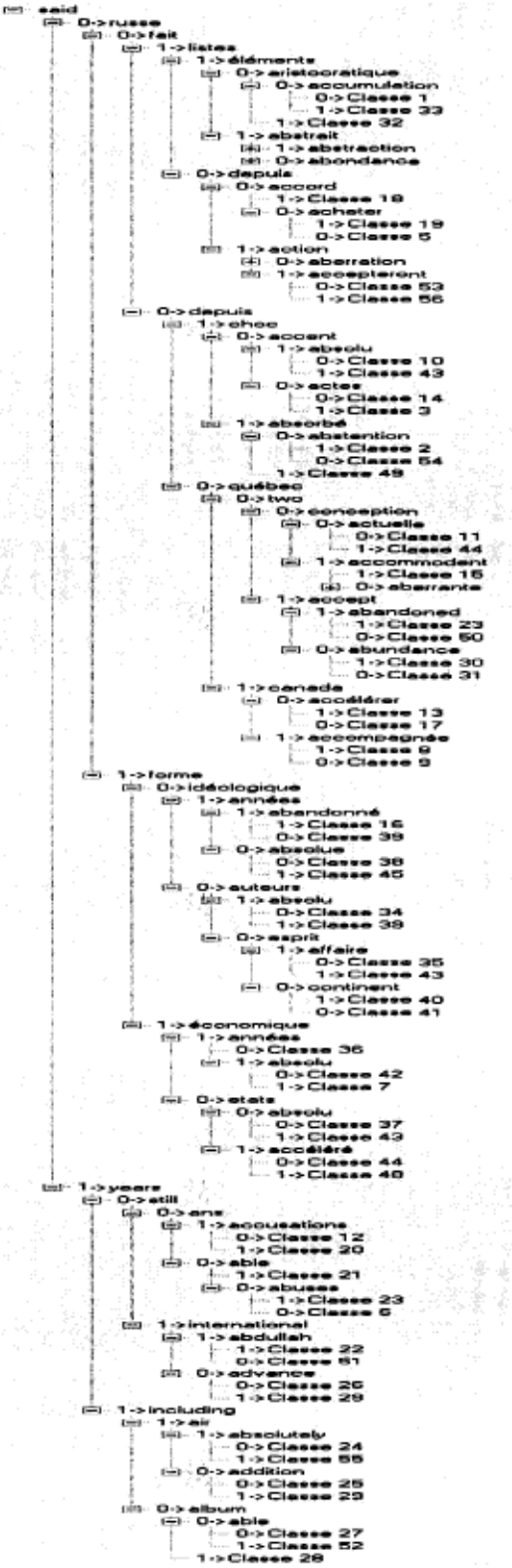

Figure 7.1: Arbre de décision C4.5 avec le jeu A

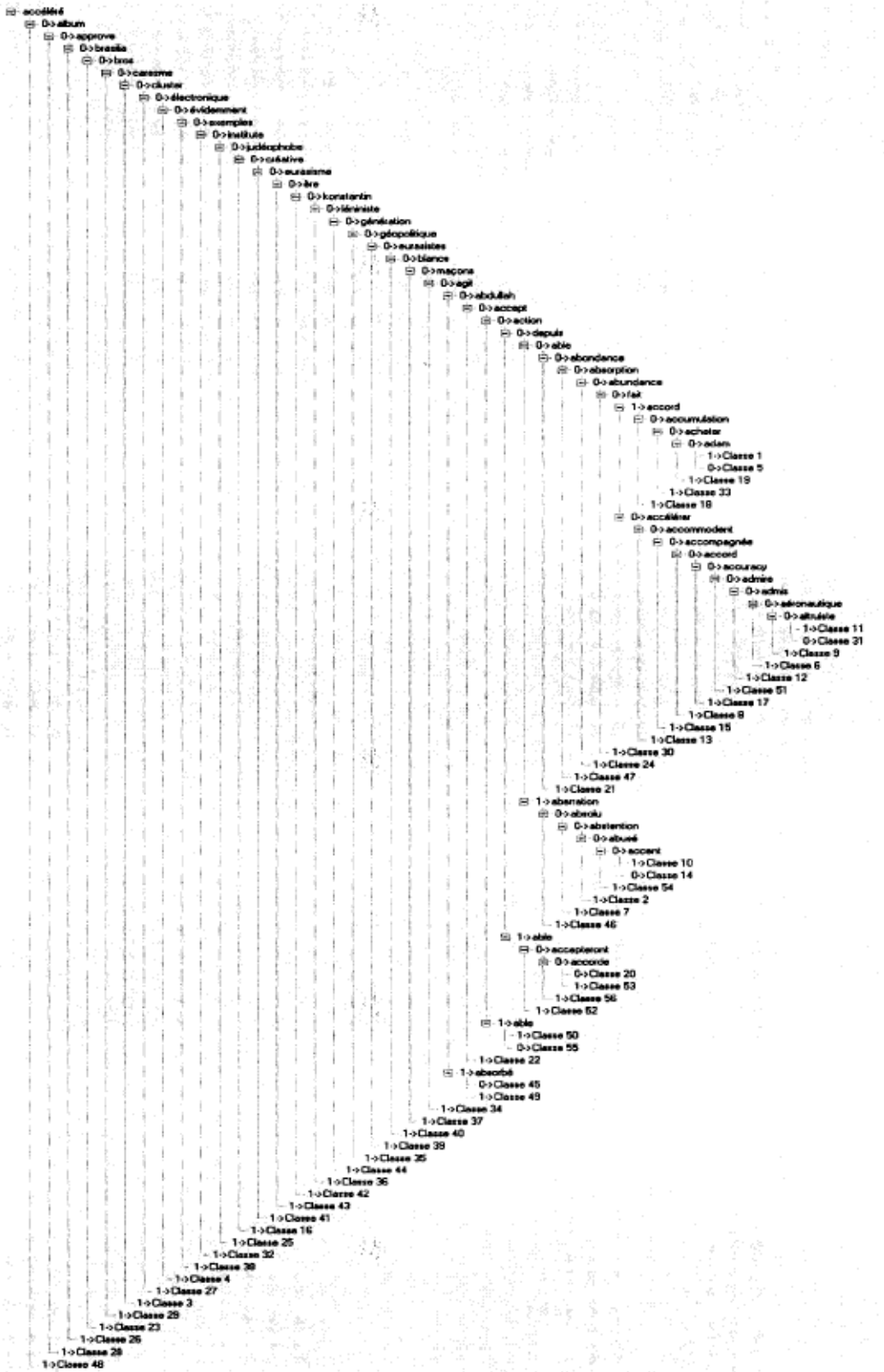

Figure 7.2: Arbre de décision CART avec le Jeu A

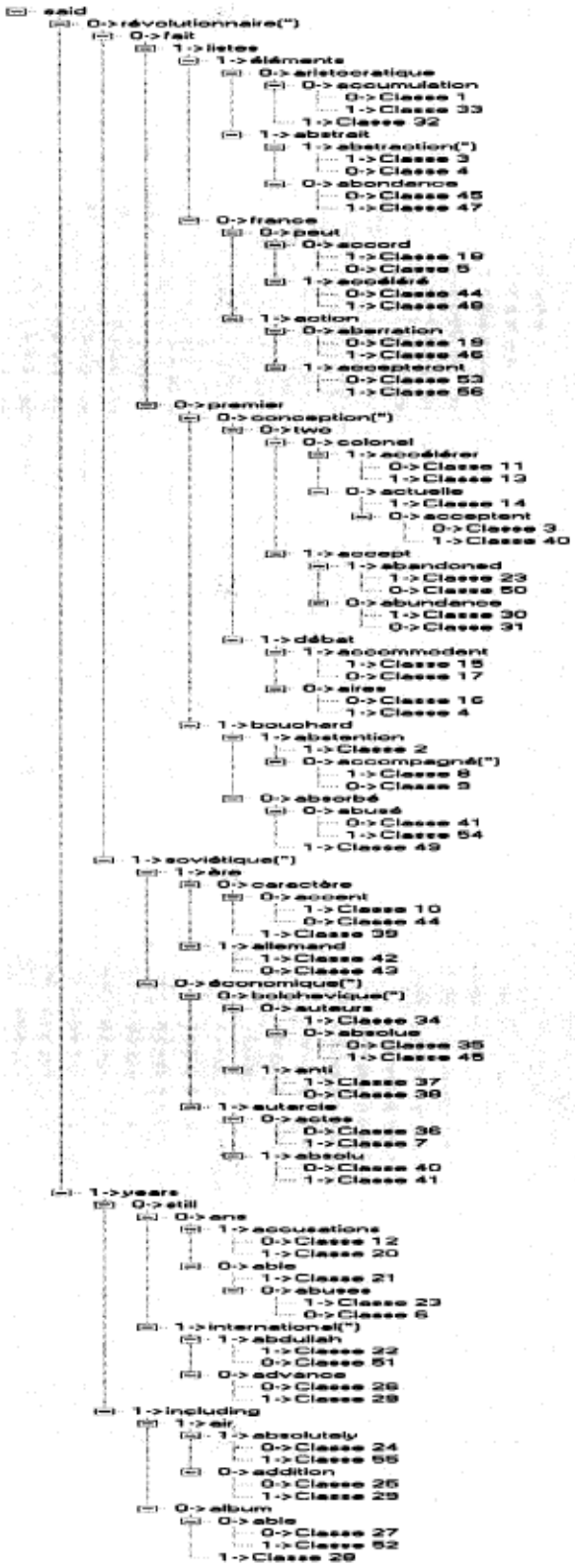

Figure 7.3: Arbre de décision C4.5 avec le jeu B

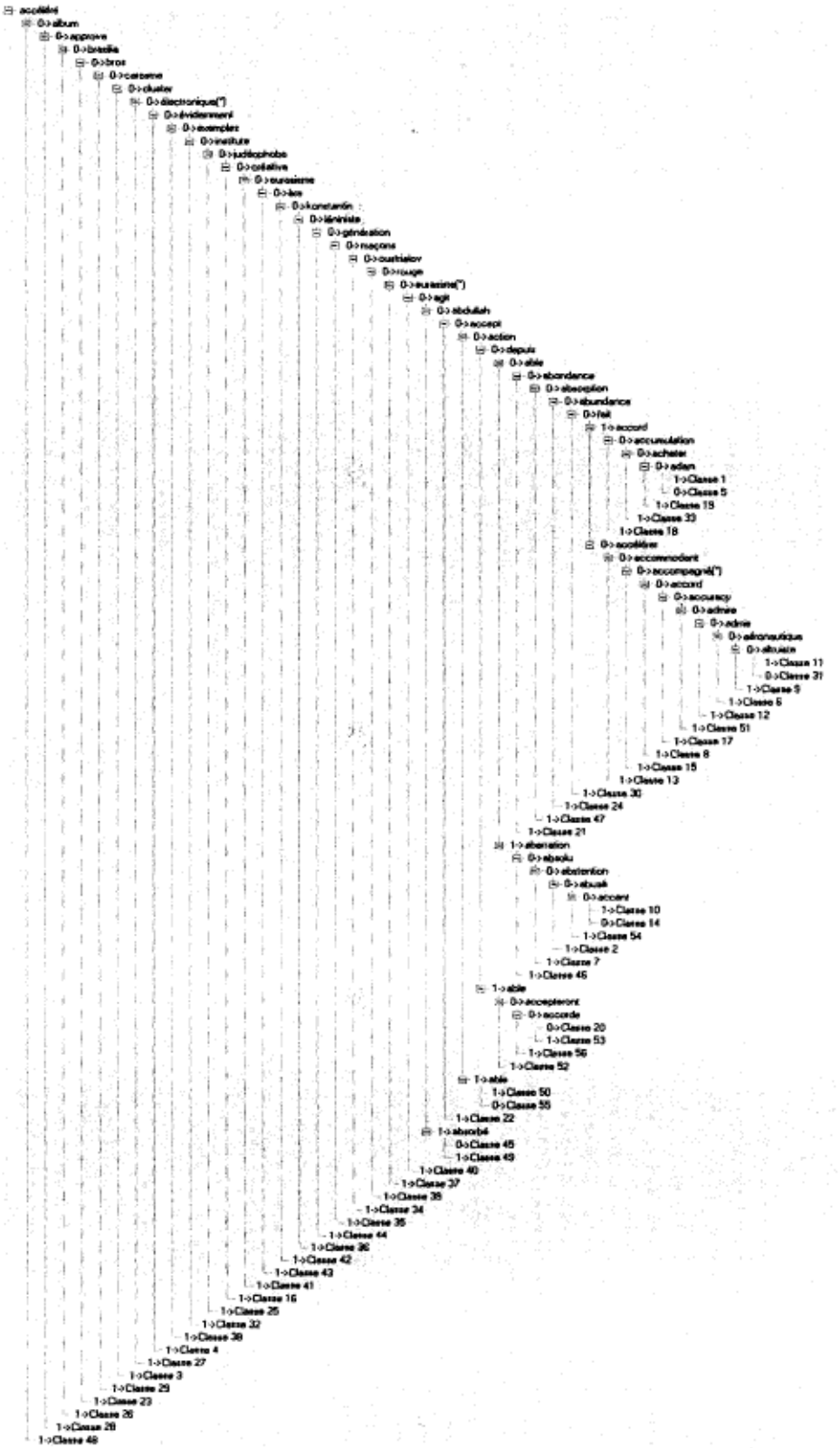

Figure 7.4: Arbre de décision CART avec le Jeu B

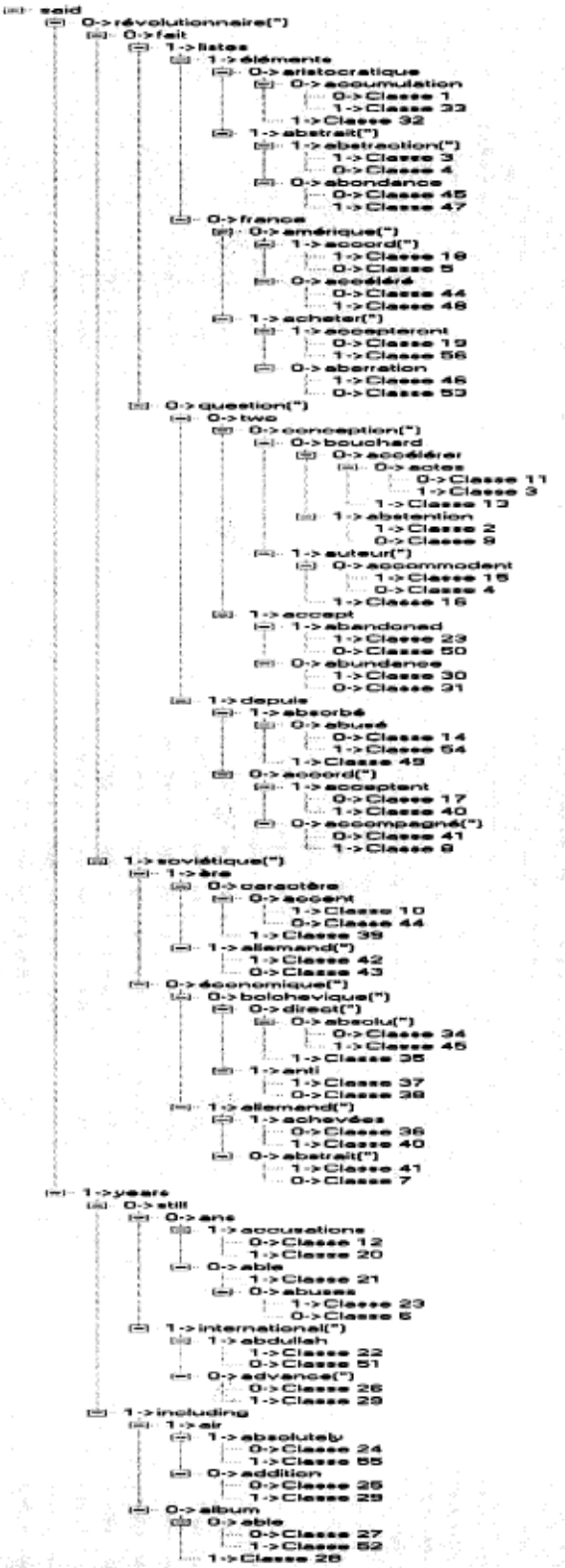

Figure 7.5: Arbre de décision C4.5 avec le Jeu C

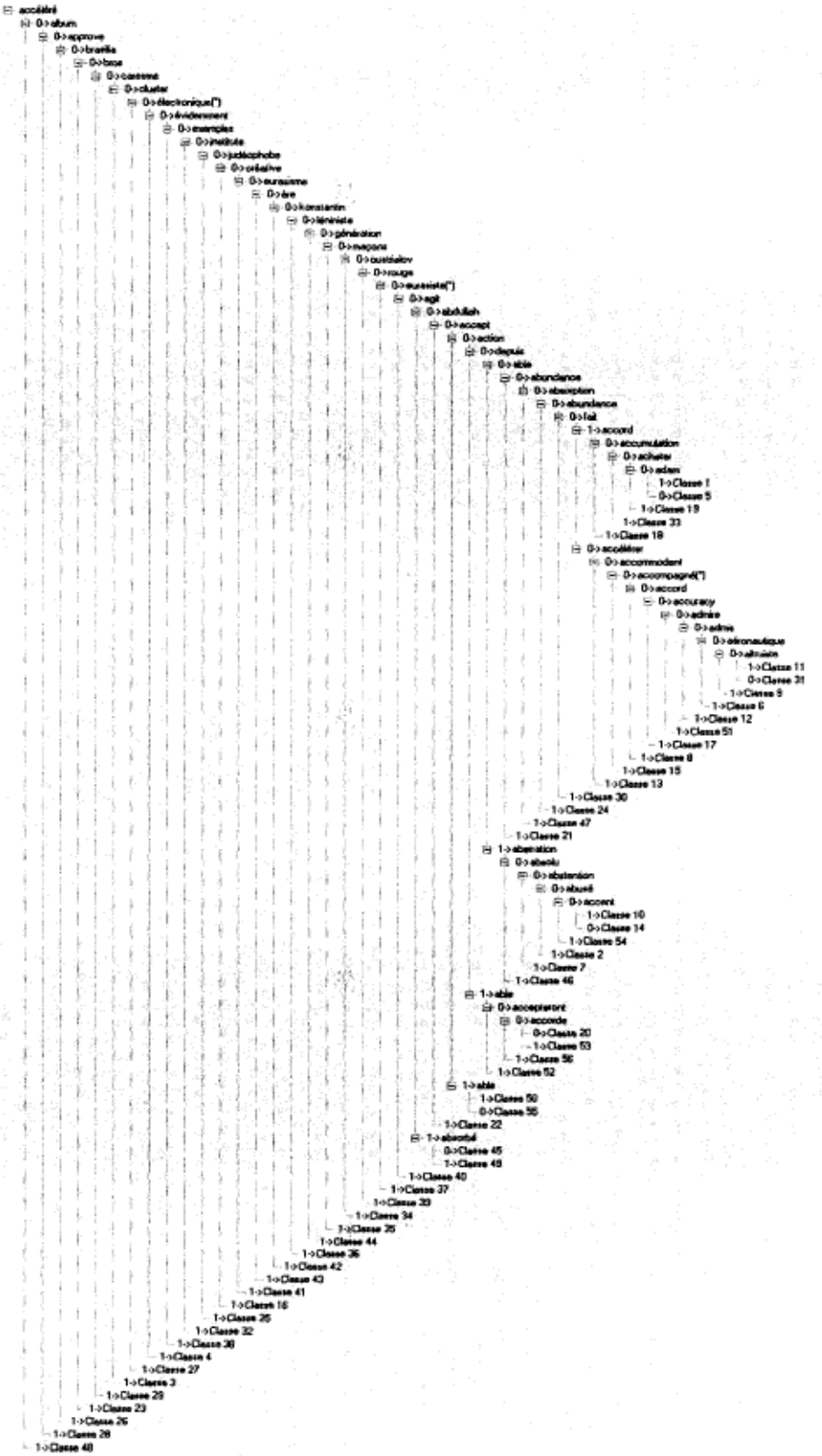

Figure 7.6: Arbre de décision CART avec le Jeu C

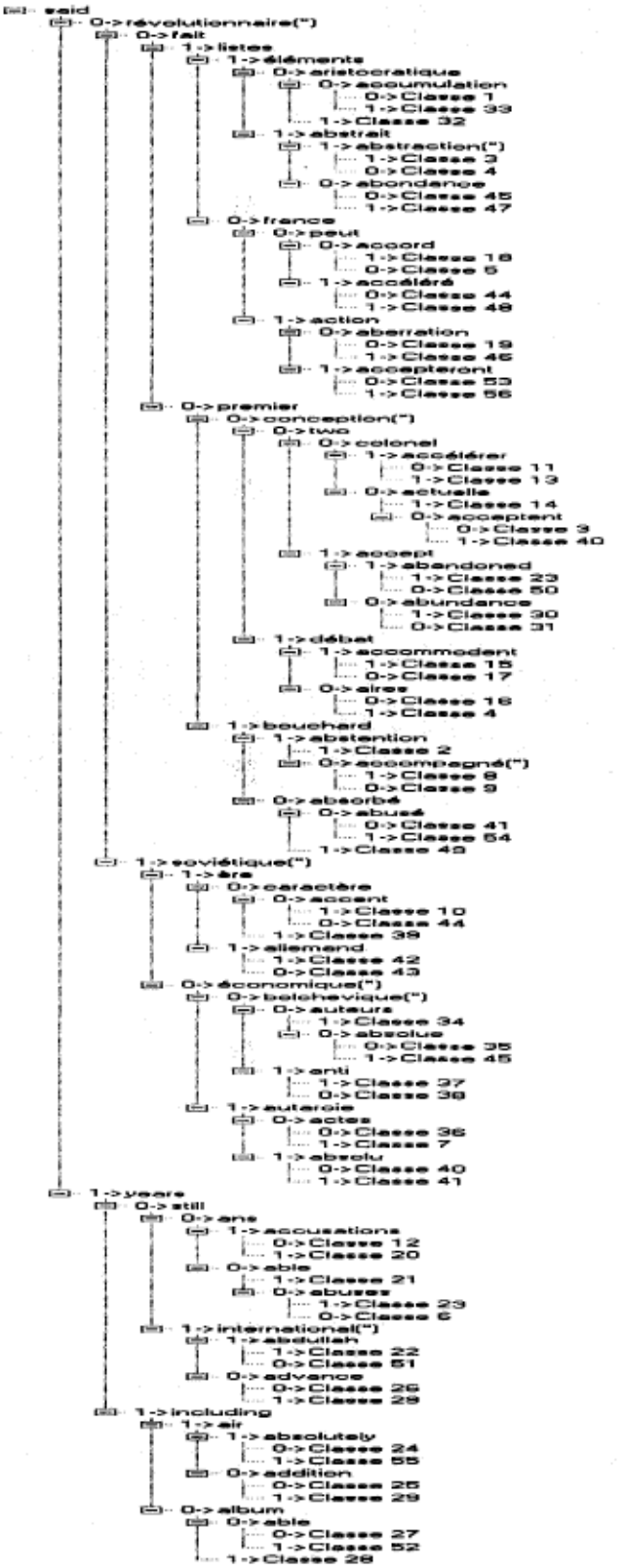

Figure 7.7: Arbre de décision C4.5 avec le Jeu D

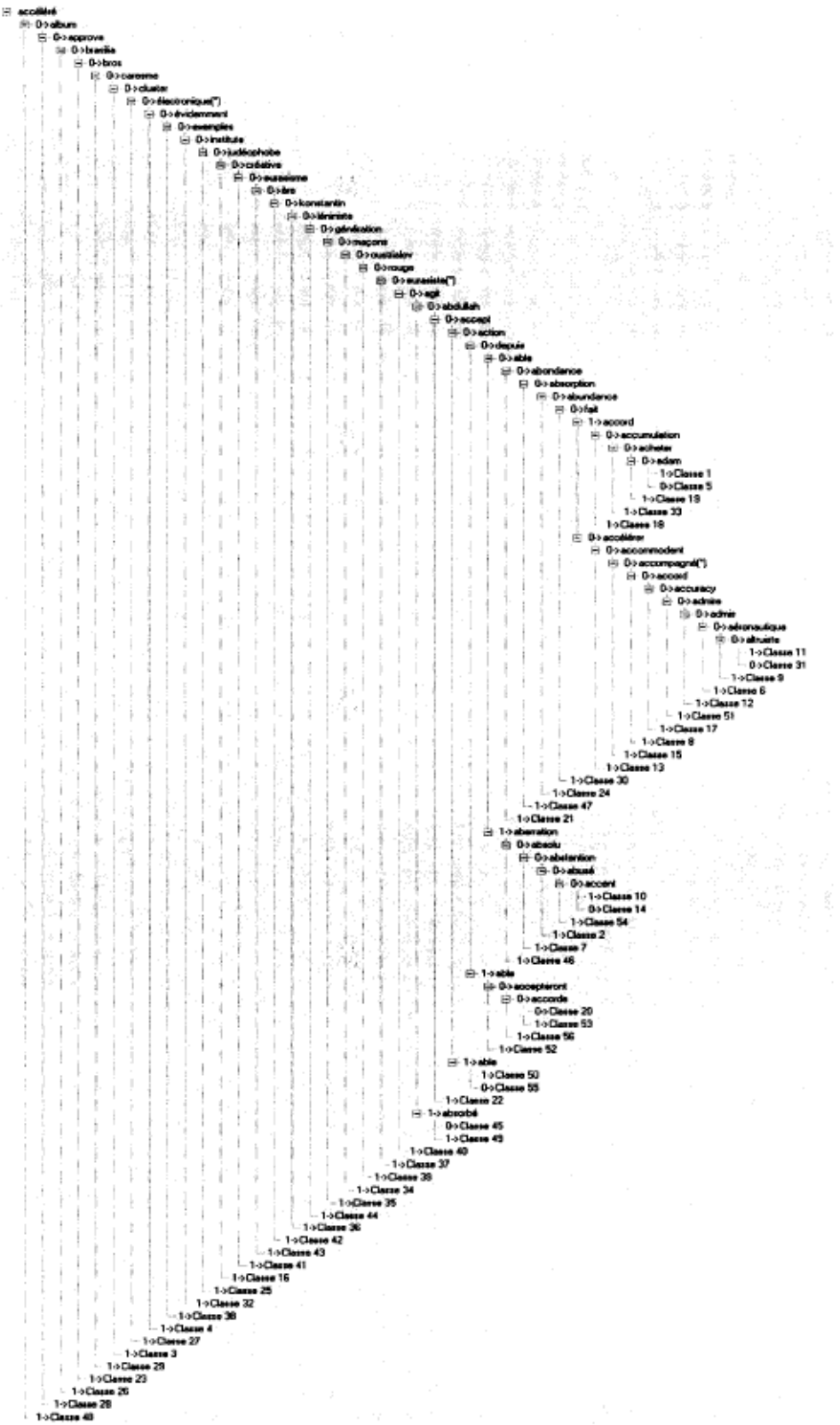

Figure 7.8: Arbre de décision CART avec le Jeu D

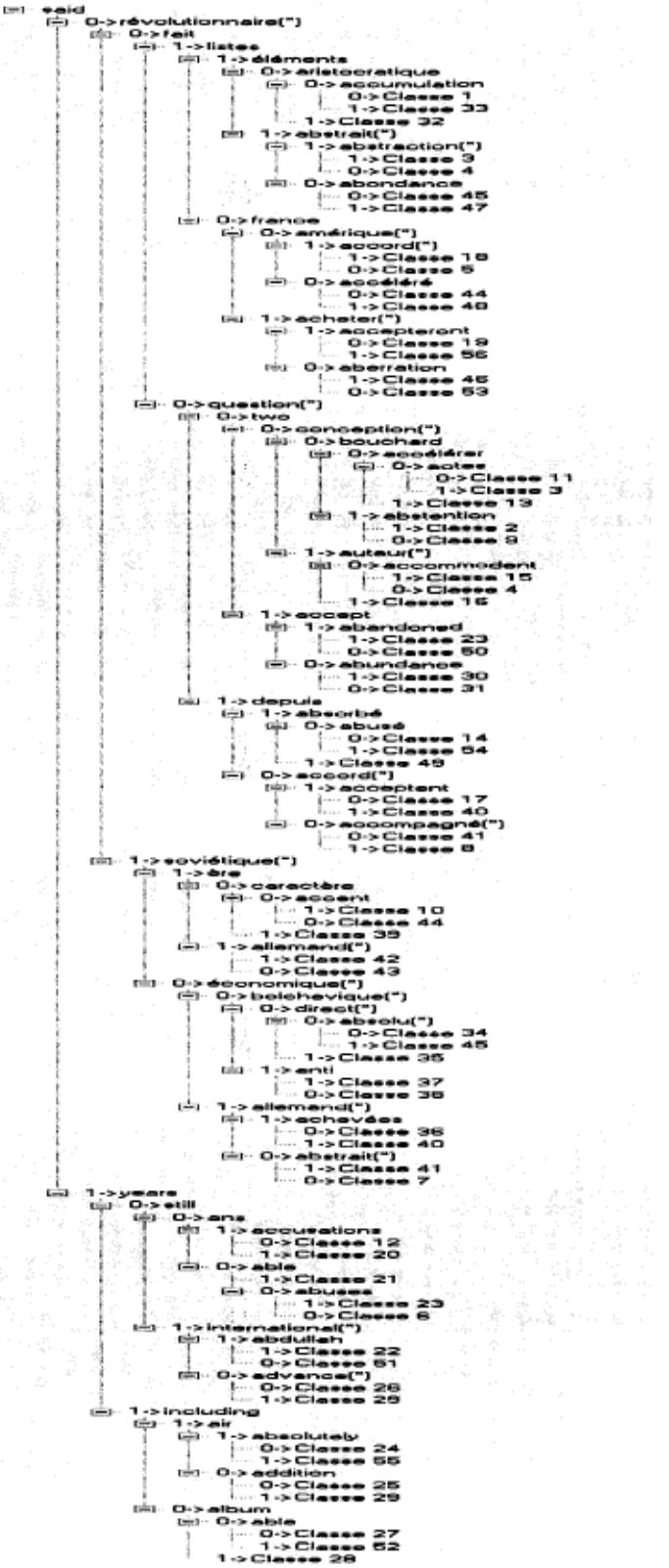

Figure 7.9: Arbre de décision C4.5 avec le Jeu E

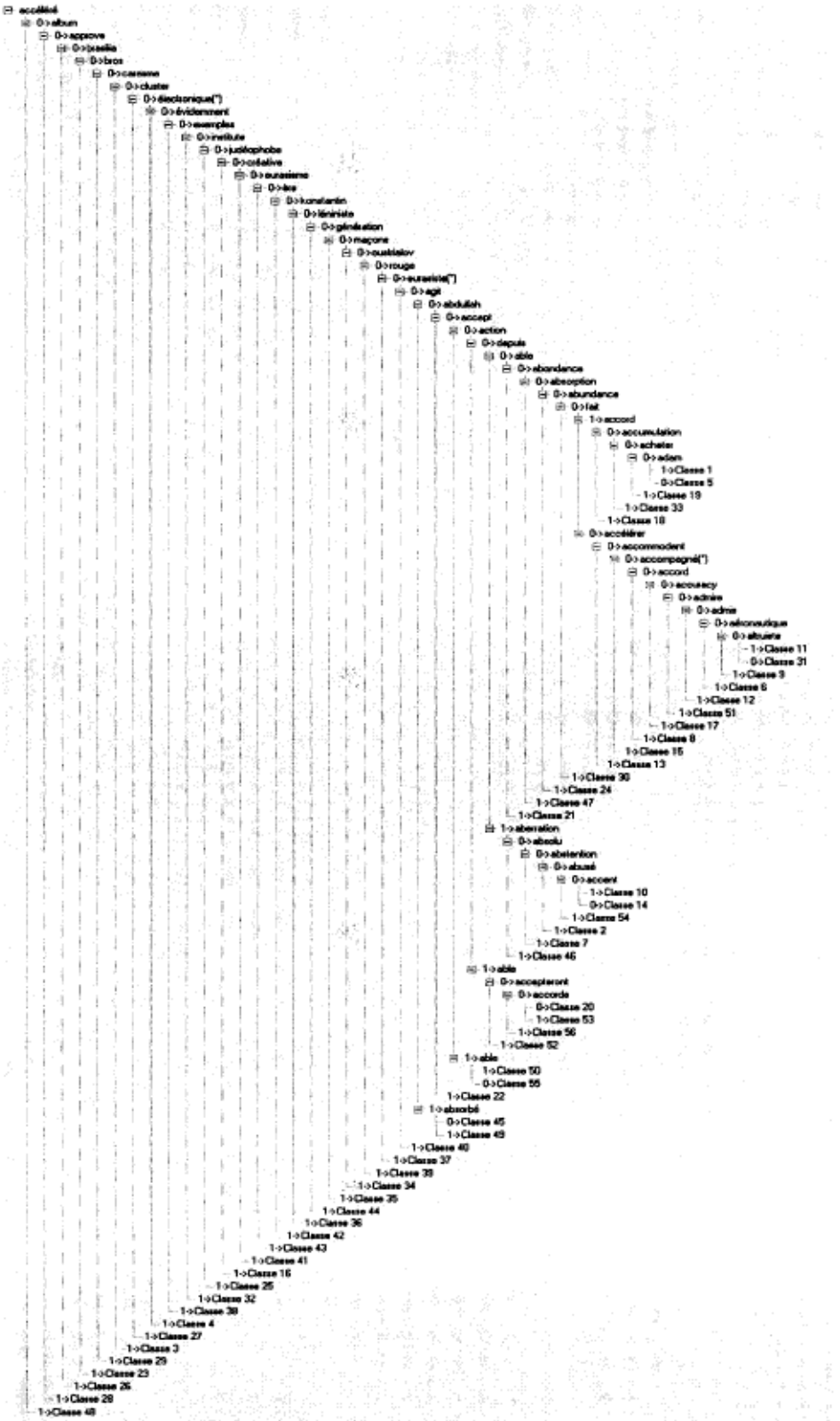

Figure 7.10: Arbre de décision CART avec le Jeu E

Annexe C

Exemple de règles de classification

L'annexe C contient des exemples de règles de classification, ces règles sont le reflet direct d'arbres de décision. Les mots du dictionnaire ont été traités avec les quadri-grammes avec un seuil de 75 %.

Règles de décision C4.5

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 1 AND éléments = 0 AND aristocratique = 0 AND accumulation = 0 THEN Classe 1

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 1 AND éléments = 0 AND aristocratique = 0 AND accumulation = 1 THEN Classe 33

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 1 AND éléments = 0 AND aristocratique = 1 THEN Classe 32

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 1 AND éléments = 1 AND abstrait(*) = 1 AND abstraction(*) = 1 THEN Classe 3

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 1 AND éléments = 1 AND abstrait(*) = 1 AND abstraction(*) = 0 THEN Classe 4

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 1 AND éléments = 1 AND abstrait(*) = 0 AND abondance = 0 THEN Classe 45

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 1 AND éléments = 1 AND abstrait(*) = 0 AND abondance = 1 THEN Classe 47

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 0 AND france = 0 AND amérique(*) = 1 AND accord(*) = 1 THEN Classe 18

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 0 AND france = 0 AND amérique(*) = 1 AND accord(*) = 0 THEN Classe 5

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 0 AND france = 0 AND amérique(*) = 0 AND accéléré = 0 THEN Classe 44

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 0 AND france = 0 AND amérique(*) = 0 AND accéléré = 1 THEN Classe 48

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 0 AND france = 1 AND acheter(*) = 1 AND accepteront = 0 THEN Classe 19

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 0 AND france = 1 AND acheter(*) = 1 AND accepteront = 1 THEN Classe 56

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 0 AND france = 1 AND acheter(*) = 0 AND aberration = 1 THEN Classe 46

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 1 AND listes = 0 AND france = 1 AND acheter(*) = 0 AND aberration = 0 THEN Classe 53

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 0 AND conception(*) = 0 AND bouchard = 0 AND accélérer = 0 AND actes = 0 THEN Classe 11

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 0 AND conception(*) = 0 AND bouchard = 0 AND accélérer = 0 AND actes = 1 THEN Classe 3

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 0 AND conception(*) = 0 AND bouchard = 0 AND accélérer = 1 THEN Classe 13

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 0 AND conception(*) = 0 AND bouchard = 1 AND abstention = 1 THEN Classe 2

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 0 AND conception(*) = 0 AND bouchard = 1 AND abstention = 0 THEN Classe 9

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 0 AND conception(*) = 1 AND auteur(*) = 0 AND accommodent = 1 THEN Classe 15

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 0 AND conception(*) = 1 AND auteur(*) = 0 AND accommodent = 0 THEN Classe 4

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 0 AND conception(*) = 1 AND auteur(*) = 1 THEN Classe 16

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 1 AND accept = 1 AND abandoned = 1 THEN Classe

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 1 AND accept = 1 AND abandoned = 0 THEN Classe 50

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 1 AND accept = 0 AND abundance = 1 THEN Classe 30

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 0 AND two = 1 AND accept = 0 AND abundance = 0 THEN Classe 31

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 1 AND depuis = 1 AND absorbé = 0 AND abusé = 0 THEN Classe 14

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 1 AND depuis = 1 AND absorbé = 0 AND abusé = 1 THEN Classe 54

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 1 AND depuis = 1 AND absorbé = 1 THEN Classe 49

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 1 AND depuis = 0 AND accord(*) = 1 AND acceptent = 0 THEN Classe 17

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 1 AND depuis = 0 AND accord(*) = 1 AND acceptent = 1 THEN Classe 40

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 1 AND depuis = 0 AND accord(*) = 0 AND accompagné(*) = 0 THEN Classe 41

IF said = 0 AND révolutionnaire(*) = 0 AND fait = 0 AND question(*) = 1 AND depuis = 0 AND accord(*) = 0 AND accompagné(*) = 1 THEN Classe 8

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 1 AND ère = 0 AND caractère = 0 AND accent = 1 THEN Classe 10

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 1 AND ère = 0 AND caractère = 0 AND accent = 0 THEN Classe 44

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 1 AND ère = 0 AND caractère = 1 THEN Classe 39

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 1 AND ère = 1 AND allemand(*) = 1 THEN Classe 42

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 1 AND ère = 1 AND allemand(*) = 0 THEN Classe 43

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 0 AND bolchevique(*) = 0 AND direct(*) = 0 AND absolu(*) = 0 THEN Classe 34

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 0 AND bolchevique(*) = 0 AND direct(*) = 0 AND absolu(*) = 1 THEN Classe 45

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 0 AND bolchevique(*) = 0 AND direct(*) = 1 THEN Classe 35

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 0 AND bolchevique(*) = 1 AND anti = 1 THEN Classe 37

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 0 AND bolchevique(*) = 1 AND anti = 0 THEN Classe 38

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 1 AND allemand(*) = 1 AND achevées = 0 THEN Classe 36

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 1 AND allemand(*) = 1 AND achevées = 1 THEN Classe 40

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 1 AND allemand(*) = 0 AND abstrait(*) = 1 THEN Classe 41

IF said = 0 AND révolutionnaire(*) = 1 AND soviétique(*) = 0 AND économique(*) = 1 AND allemand(*) = 0 AND abstrait(*) = 0 THEN Classe 7

IF said = 1 AND years = 0 AND still = 0 AND ans = 1 AND accusations = 0 THEN Classe 12

IF said = 1 AND years = 0 AND still = 0 AND ans = 1 AND accusations = 1 THEN Classe 20

IF said = 1 AND years = 0 AND still = 0 AND ans = 0 AND able = 1 THEN Classe 21

IF said = 1 AND years = 0 AND still = 0 AND ans = 0 AND able = 0 AND abuses = 1 THEN Classe 23

IF said = 1 AND years = 0 AND still = 0 AND ans = 0 AND able = 0 AND abuses = 0 THEN Classe 6

IF said = 1 AND years = 0 AND still = 1 AND international(*) = 1 AND abdullah = 1 THEN Classe 22

IF said = 1 AND years = 0 AND still = 1 AND international(*) = 1 AND abdullah = 0 THEN Classe 51

IF said = 1 AND years = 0 AND still = 1 AND international(*) = 0 AND advance(*) = 0 THEN Classe 26

IF said = 1 AND years = 0 AND still = 1 AND international(*) = 0 AND advance(*) = 1 THEN Classe 29

IF said = 1 AND years = 1 AND including = 1 AND air = 1 AND absolutely = 0 THEN Classe 24

IF said = 1 AND years = 1 AND including = 1 AND air = 1 AND absolutely = 1 THEN Classe 55

IF said = 1 AND years = 1 AND including = 1 AND air = 0 AND addition = 0 THEN Classe 25

IF said = 1 AND years = 1 AND including = 1 AND air = 0 AND addition = 1 THEN Classe 29

IF said = 1 AND years = 1 AND including = 0 AND album = 0 AND able = 0 THEN Classe 27

IF said = 1 AND years = 1 AND including = 0 AND album = 0 AND able = 1 THEN Classe 52

IF said = 1 AND years = 1 AND including = 0 AND album = 1 THEN Classe 28

Règles de decision CART

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 0 AND accommodent = 0 AND accompagné(*) = 0 AND accumulation = 0 AND accuracy = 0 AND acheter(*) = 0 AND adam = 1 THEN Classe 1

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 0 AND accommodent = 0 AND accompagné(*) = 0 AND accumulation = 0 AND accuracy = 0 AND acheter(*) = 0 AND adam = 0 AND admire(*) = 1 THEN Classe 12

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND

ancien(*) = 0 AND accélérer = 0 AND accommodent = 0 AND accompagné(*) = 0 AND accumulation = 0 AND accuracy = 0 AND acheter(*) = 0 AND adam = 0 AND admire(*) = 0 AND admis = 0 AND aéronautique = 0 THEN Classe 31

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 0 AND accommodent = 0 AND accompagné(*) = 0 AND accumulation = 0 AND accuracy = 0 AND acheter(*) = 0 AND adam = 0 AND admire(*) = 0 AND admis = 0 AND aéronautique = 1 THEN Classe 9

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 0 AND accommodent = 0 AND accompagné(*) = 0 AND accumulation = 0 AND accuracy = 0 AND acheter(*) = 0 AND adam = 0 AND admire(*) = 0 AND admis = 1 THEN Classe 6

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 0 AND accommodent = 0 AND accompagné(*) = 0 AND accumulation = 0 AND accuracy = 0 AND acheter(*) = 1 THEN Classe 19

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND

abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 0 AND accommodent = 0 AND accompagné(*) = 0 AND accumulation = 0 AND accuracy = 1 THEN Classe 51

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 0 AND accommodent = 0 AND accompagné(*) = 0 AND accumulation = 1 THEN Classe 33

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 0 AND accommodent = 0 AND accompagné(*) = 1 THEN Classe 8

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 0 AND accommodent = 1 THEN Classe 15

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 0 AND accélérer = 1 THEN Classe 13

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND

konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 1 AND abundance = 0 THEN Classe 20

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 0 AND ancien(*) = 1 AND abundance = 1 THEN Classe 30

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 1 AND altruiste = 1 THEN Classe 11

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 0 AND amérique(*) = 1 AND altruiste = 0 THEN Classe 5

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 0 AND absorption = 1 THEN Classe 24

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND

konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 0 AND abondance = 1 THEN Classe 47

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 1 AND aboard = 1 THEN Classe 21

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 0 AND able = 1 AND aboard = 0 THEN Classe 52

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 1 AND aberration = 0 AND absolu(*) = 0 AND abstention = 0 AND accent = 1 THEN Classe 10

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 1 AND aberration = 0 AND absolu(*) = 0 AND abstention = 0 AND accent = 0 AND accepteront = 0 THEN Classe 14

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND

abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 1 AND aberration = 0 AND absolu(*) = 0 AND abstention = 0 AND accent = 0 AND accepteront = 1 THEN Classe 56

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 1 AND aberration = 0 AND absolu(*) = 0 AND abstention = 1 THEN Classe 2

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 1 AND aberration = 0 AND absolu(*) = 1 THEN Classe 7

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 0 AND depuis = 1 AND aberration = 1 THEN Classe 46

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 1 AND abusé = 0 AND accueilli(*) = 0 AND action(*) = 0 THEN Classe 17

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 1 AND abusé = 0 AND accueilli(*) = 0 AND action(*) = 1 THEN Classe 53

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND

konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 1 AND abusé = 0 AND accueilli(*) = 1 THEN Classe 18

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 0 AND accord(*) = 1 AND abusé = 1 THEN Classe 54

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 1 AND able = 1 THEN Classe 50

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 0 AND accept = 1 AND able = 0 THEN Classe 55

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 0 AND abdullah = 1 THEN Classe 22

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 1 AND absorbé = 0 THEN Classe 45

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND

konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 0 AND agit = 1 AND absorbé = 1 THEN Classe 49 (Ok:2 Erreur:0)

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 0 AND eurasiste(*) = 1 THEN Classe 40

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 0 AND rouge = 1 THEN Classe 37

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 0 AND oustrialov = 1 THEN Classe 39

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 0 AND maçons = 1 THEN Classe 34

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 0 AND génération = 1 THEN Classe 35

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 0 AND léniniste = 1 THEN Classe 44

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 0 AND konstantin = 1 THEN Classe 36

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 0 AND aristocratique = 1 THEN Classe 32

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 0 AND ère = 1 THEN Classe 42 (Ok:3 Erreur:0)

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 0 AND eurasisme = 1 THEN Classe 43

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 0 AND créative = 1 THEN Classe 41

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 0 AND judéophobe = 1 THEN Classe 16

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 0 AND évidemment = 1 THEN Classe 38

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 0 AND électronique(*) = 1 THEN Classe 4 (Ok:3 Erreur:0)

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 0 AND cluster(*) = 1 THEN Classe 27

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 0 AND caresme = 1 THEN Classe 3

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 0 AND bros = 1 THEN Classe 29

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 0 AND brasilia = 1 THEN Classe 23

IF accéléré = 0 AND advocate(*) = 0 AND album = 0 AND benefit(*) = 1 THEN Classe 26

IF accéléré = 0 AND advocate(*) = 0 AND album = 1 THEN Classe 28

IF accéléré = 0 AND advocate(*) = 1 THEN Classe 25

IF accéléré = 1 THEN Classe 48

Annexe D

L'annexe B contient les résultats de la simulation effectuée avec un algorithme génétique, pour chacune des itérations, on a retenu uniquement les règles acceptables, c'est-à-dire que les règles correspondent à au moins un exemple du jeu de tests. Il a 3 catégories de règles acceptables, la première correspond aux règles qui sont associées à la bonne classe, la deuxième correspond aux règles qui sont associées à la mauvaise classe et la troisième correspond aux règles qui sont associées parfois à la bonne classe et dans d'autres cas, les règles sont associées à la mauvaise classe.

Jeu	Itération 0			Itération 20			Itération 40			Itération 60			Itération 80		
	Ok	Err	OK/Err	Ok	Err	OK/Err	Ok	Err	OK/Err	Ok	Err	OK/Err	Ok	Err	OK/Err
Aléatoire A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra A	77	0	0	26	0	0	13	0	0	9	0	0	20	12	0
Règles non extra A	123	0	0	14	77	39	0	120	43	0	104	77	38	32	20
Combiné A	200	0	0	23	0	0	16	1	0	30	2	0	5	7	0
Aléatoire B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra B	72	0	0	27	0	0	12	0	0	6	0	0	26	26	0
Règles non extra B	120	0	0	35	425	215	4	601	297	29	564	236	114	337	73
Combiné B	192	0	0	23	1	0	20	0	0	25	3	0	32	26	0
Aléatoire C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra C	72	0	0	27	0	0	12	0	0	13	0	0	27	16	0
Règles non extra C	120	0	0	38	486	274	133	299	471	424	197	279	215	233	62
Combiné C	192	0	0	21	0	0	14	0	0	17	0	0	4	5	0
Aléatoire D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra D	72	0	0	28	0	0	14	1	0	8	2	0	4	0	0
Règles non extra D	120	0	0	32	446	252	5	345	241	4	297	583	14	225	709
Combiné D	192	0	0	20	1	0	9	0	0	8	1	0	10	2	0
Aléatoire E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra E	72	0	0	32	1	0	11	1	0	16	0	0	58	32	0
Règles non extra E	120	0	0	34	413	198	15	526	339	23	514	163	1	435	61
Combiné E	192	0	0	27	0	0	19	1	0	28	3	0	30	33	0

Table 7-1: Tableau des fréquences des règles acceptables des 100 premières itérations d'un AG

Jeu	Itération 100			Itération 120			Itération 140			Itération 160			Itération 180			Itération 200		
	Ok	Err	OK/Err	Ok	Err	OK/Err	Ok	Err	OK/Err	Ok	Err	OK/Err	Ok	Err	OK/Err	Ok	Err	OK/Err
Aléatoire A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra A	10	11	0	10	11	0	5	5	0	3	2	0	0	0	0	0	0	0
Règles non extra A	10	6	1	12	12	2	10	9	3	11	12	2	6	7	3	9	14	2
Combiné A	1	1	0	1	0	0	2	2	0	2	2	0	1	0	0	1	0	0
Aléatoire B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra B	32	25	0	32	33	0	30	36	0	31	26	0	35	22	0	30	32	0
Règles non extra B	16	205	5	0	40	0	0	4	0	3	11	0	0	0	0	0	0	0
Combiné B	35	26	0	34	21	0	22	29	0	31	24	0	24	21	0	37	20	0
Aléatoire C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra C	23	2	0	29	31	0	26	27	0	27	34	0	27	30	0	25	24	0
Règles non extra C	130	67	8	91	18	1	39	17	0	1	5	0	0	0	0	0	0	0
Combiné C	2	0	0	5	0	0	21	7	0	43	33	0	40	27	0	52	37	0
Aléatoire D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra D	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles non extra D	35	154	702	178	100	112	78	66	32	61	57	23	67	43	13	473	72	15
Combiné D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Aléatoire E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Règles extra E	91	35	0	86	47	0	86	43	0	101	48	0	94	35	0	91	37	0
Règles non extra E	26	624	125	158	305	70	102	40	1	79	49	5	93	64	18	70	50	16
Combiné E	29	32	0	41	38	0	47	32	0	52	33	0	46	32	0	64	41	0

Table 7-2: Tableau des fréquences des règles acceptables des 100 dernières itérations d'un AG

Bibliographies

1. ***Induction of Decision Trees***, J.R. Quinlan, Kluwer Academic publishers, 1986

2. ***Data Mining: Practical Machine Learning Tools and Technique with JAVA implementation***, Ian H. Witten, Eibe Frank, Morgan Kaufmann 2000

3. ***C4.5 : Program for machine learning***, J.R. Quinlan, Morgan Kaufmann publisher 1993

4. ***Classification and Regression Tree***, Breiman et al , California: Wadsworth International, 1984

5. ***Intelligence Artificial: a modern approach***, Stuart Russell et Peter Norvig, Prentice Hall, 1995

6. ***Arbre de décision***, Ricco Rakotomala, Revue MODULAD 2005, numéro 33

7. ***Decision Tree's Induction Strategies Evaluate on a Hard Real World Problem***, Miland Zormarnd,Vili Podgorelec, Peter Kobol, Margaret Joseph Lane, 0-7695-0484-1/00, *IEEE 2000*

8. ***Classifiability Based Pruning of Decision Trees***, Ming Dog and Ravi Kothari, 0-7803-7944-9/01, *IEEE 2001*

9. ***Error-Based Pruning of Decision Trees Gown on Very Large Data Sets Can Work!*** , Lawrence O Hall, Richard Collins, Kevin W. Bowyer, Robert Banfield, 1082-3409/02, IEEE 2002

10. ***Knowledge Representation and Acquisition Approach Based on Decision Tree***, Jianshe Baie, Bo Fan and Junyi Xue, 0-7803-7902-0/03, IEEE 2003

11. ***Decision Tree Discovery***, *Handbook of Data Mining and Knowledge Discovery*, J.R Quinlan et R. Kohavi, Klosgën and Zytkow Editors, 267-276, 2002

12. ***Decision Tree Learning on Very Large Data Sets***, Lawrence O.Hall, Nitesh Chawla et Kevin W.Bowyet, 0-7803-4778-1/98, *IEEE 1998*

13. ***Efficient C4.5***, Salvatore Ruggieri, 1041-4347/02, *IEEE 2002*

14. ***C4.5 and Imbalanced Data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure***, Nitesh Chawla, Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, 2003

15. ***Decision Tree with Minimal Costs***, Charles X.Ling, Qiang Yang, Jianning Wang, Shichao Zhang, Appearing Proceeding of 21th International Conference on Machine Learning (ICML), Banff, Canada 2004

16. ***GATree: Genetically Evolved Decision Trees***, Athanassios Papagelis, Dimitrios Kalles, 1082-3409/00, *IEEE 2000*

17. ***Fuzzy Decision Tree, Linguistic Rules and Fuzzy Knowledge-Based Network: Generation and Evaluation***, Sushmita Mitra, Kishori M. Konwar, Sankar K. Pal, 1094-6977/02, IEEE 2002

18. ***An Fuzzy matching method of Fuzzy Decision Trees***, John W.T Lee, Juan Sun, Lan-Zhen Yang, 0-7803-7865-2/03, IEEE 2003

19. ***Look-Ahead Based Fuzzy Decision Tree Induction***, Ming Dong,mRavis Kothari, IEEE Transaction on FUZZY Systems Vol 9 no 3 June 2001, 1063-6706/01

20. **Apprentissage a partir d'exemple, François Denis et Remi Gilleron, http://www.grappa.univ-lille3.fr/polys/apprentissage/index.html**

21. **Genetic algorithm in Search, Optimization, and Machine Learning**, D. Goldberg, Addison Wesley , 1989

22. **Foundation of genetic algorithms**, G Rawling, Morgan Kaufmann Publisher, 1991

23. **Genetic Algorithms + Data Structures = Evolution Programs**, Z. Michalewicz, Springer, 1996

24. **Handbook of Data Mining and Knowledge Discovery**, W .Klögen et J Zytkow, Oxford University Press, 2002

25. **Data Mining: Opportunities and Challenges**, J Wang, IRM Press, 2003

26. **Algorithmes évolutionnaires**, S. Bernard, 2003, http://taxules.free.fr/cours_MP/tipe/algogene.html

27. **Darwinisme artificiel : une vue d'ensemble**, E. Lutton, Février 2003, http://fractales.inria.fr/html/Cours/cours_ea_aux/AE-RevueTSI.pdf

28. **Algorithmes génétiques**, T. Sabrina, 2003, http://sis.univ-tln.fr/~tollari/TER/AlgoGen1/

29. **Algorithme Génétique**, V. Magnin, Décembre 1999 , http://www.eudil.fr/~vmagnin/coursag/methodes_ag.html

30. **Les systèmes de classifieurs,** L. Mamlouk, http://www.limsi.fr/Individu/jps/enseignement/examsma/2003/MAMLOUK_BENOUIRANE/classifieurs.htm

31. **Synthèse de comportement animaux individuels et collectif par algorithmes génétiques,** R. Dumeur , 1995, http://www.ai.univ-paris8.fr/~renaud/publications/hthese/node49.html

32. CRISP-DM v1.0 (Cross Industry Standard Process for Data Mining) http://www.crisp-dm.org

33. **Les n-grams de caractères pour l'extraction de connaissances dans les bases de données textuelles multilingues**, Ismail Biskri et Sylvain Delisle, Juillet 2001

34. **Application de l'algorithme génétique à l'analyse terminologique**, Vincent Rialle et al, http://www.cavi.univ-paris3.fr/lexicometrica/jadt/jadt1998/rialle.htm

35. **An exploratory technique for investigating large quantities of categorical data**, Applied Statistic, G. Kass, 1980