

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Ahmed Draia - Adrar
Faculté des Sciences et de la Technologie
Département des Mathématiques et Informatique



Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en informatique

Option : Réseaux et Systèmes Intelligents

Thème

Sélection des modèles SVM pour la reconnaissance des caractères arabes manuscrits

Préparé par :

ABED Djemaa.

BAKHOUIA Roqiya.

Soutenu le : 2/06/2015 devant le jury composé de :

Dr. OMARI Mohammed

Président

Mr. MAMOUNI El mamoun

Encadreur

Mr. KOUHILI Mohammed

Examineur

Mr. OUAHAB Abd elouahab

Examineur

Année Universitaire 2014/2015.

Remerciement

On remercie notre grand dieu pour tous ce qu'il nous offre

Tout d'abord un grand merci à notre encadreur Mr Mamouni El Mamoun, pour son orientation et ses précieux conseils qui nous ont beaucoup aidés dans notre travail.

Nous remercions Mr Kohili Mohammed et Dr Omari Mohammed enseignants au département informatique de l'université d'Adrar qui nous a ouvert les porte du savoir et tous le corps enseignant qui à contribué à notre formation en lui exprimant notre profond respect.

Nous souhaitons remercier également tout les enseignants et toutes les personnes qui ont aidé, de près ou de loin à la réalisation de ce travail de fin d'étude.

Nos remerciements vont également à tous nos collègues de l'institut informatique pour leurs soutiens.

Roqiya+Djemaa

Dédicace

- *Grace à « الله » tous puissant et en signe de reconnaissance à tous les sacrifices consentis pour ma réussite et la volonté pour mener à bien ce modeste travail que je dédie*
- *Aux personnes les plus chères à mon cœur et qui ont attendu avec patience les fruits de leur bonne éducation, à mes chers parents.*
- *A mes sœurs que j'admire tant pour leurs parcours de formation que pour leurs réussites professionnelles qui m'ont soutenu et encouragé.*
- *A celle qui partage ma vie, qui a émis un rayon de soleil dans mon chemin, à ma très chère femme.*
- *A tous ceux que j'aime et qui m'aime, ma famille, mes amis et à tous ceux qui ont l'amour d'apprendre.*

ABED Djemaa

Dédicace

Grace à « الله » Je dédie ce modeste travail :

- ➡ *A ma très chère mère qui m'a toujours soutenue dans mes joies comme mes peines ;*
- ➡ *A mon très cher père qui s'est tant donné et sacrifié ;*
- ➡ *A mes très chères sœurs ;*
- ➡ *A mes très chers frères ;*
- ➡ *A toute la famille sans exception*
- ➡ *A mon ami et ma sœurs « Djemaa ABED » ;*
- ➡ *A tous mes amis ;*
- ➡ *A tous mes enseignements de première jusqu'à maintenant.*

Roqiya BAKHOUIA

Résumé

Le travail présenté dans ce mémoire s'intègre dans le cadre de la reconnaissance automatique des caractères arabe manuscrite, et répond à la nécessité d'expérimenter une nouvelle méthode d'apprentissage : les machines à vecteurs de support (SVM : Support Vectors Machines), appliqué à la reconnaissance .L'apprentissage est alors, une caractéristique fondamentale baser sur la sélection et l'optimisations des paramètre des noyaux et types des classifieurs (une contre une , une contre reste) de modèle à vecteurs de support . Le choix de ces paramètres a une grande influence sur les performances du classifieur final et aussi sur le temps de calcul.

L'une des techniques méta-heuristiques utilisés pour l'optimisation et la sélection sont les algorithmes génétiques (AG) qui sont favorables dans tel domaine à cause de leurs caractéristiques. Le système proposé combine le SVM avec cette technique de sélection (optimisation des paramètres), pour la reconnaissance des caractères arabe dans les quatre formes (isolé, début, milieu et fin) l'expérimentation menée sur une base de données de 3920 image.

Les contributions proposées ont été validées et appuyées par des analyses montrant leurs avantages et leurs limites. Les résultats sont encourageants et ouvrent de nouvelles perspectives de recherche.

Mots clés : reconnaissance, caractères arabes manuscrits, Machines à vecteurs de support, SVM, sélection de modèle, optimisation des paramètres, fonction noyau, méta-heuristique, algorithme génétique.

ملخص

العمل المقدم في هذه المذكرة يندرج في إطار التعرف التلقائي على الحروف العربية ، ومن أجل ذلك نحتاج لاختبار آلية جديدة للتعلم وهي : الآلة الحاملة للأشعة (SVM) وتطبيقها للتعرف على الحروف. التعلم هو سمة أساسية يقوم على الاختيار والتحسين من متغيرات النوى ونوع النموذج المصنف لآلة الأشعة الحاملة (واحد ضد واحد أو واحد ضد البقية). اختيار هذه المعايير لديه تأثير كبير على أداء المصنف النهائي وأيضا حساب الوقت.

واحدة من تقنيات الاستدلال الفوقي التي تستخدم للتحسين و الاختيار تدعى الخوارزميات الجينية (GA) وهي مناسبة لهذا المجال بسبب خصائصها المميزة . النظام المقترح يجمع بين SVM مع هذه التقنية (معلمات الأمثل)، للتعرف على الحروف العربية في أشكال الأربعة (معزولة، بداية، وسط ونهاية) التجربة أجريت على قاعدة بيانات تحتوي 3920 صورة. تم التحقق من صحة المساهمات المقترحة والمعتمدة من قبل تحليلات أظهرت مزاياها وحدودها. وكانت النتائج مشجعة وقد فتحت وجهات نظر جديدة للبحث. كلمات البحث: التعرف، الحروف العربية ، الآلة الحاملة للأشعة ، SVM ، اختيار النموذج، تحسين المتغيرات، النواة، الاستدلال الفوقي ، الخوارزميات الجينية

Abstract

The work presented in this memorial fits into the framework of the recognition of handwritten Arabic characters recognition, and meets the need to test a new method of learning: the support vector machine (SVM: Support Vectors Machines) applied to the recognition the training is a fundamental characteristic based on the selection and optimization of setting kernel and kinds of classifiers (one against one, one against rest) of model support vector. The choice of these parameters has a great influence on the final classifier performance and also the calculation time.

One of this meta-heuristic techniques used for optimization and selection called genetic algorithms (GA) are favorable in such a field because of their characteristics. The proposed system combines the SVM with this selection technology (optimization parameters), for recognition of handwritten Arabic characters recognition in the four forms (isolated, beginning, middle and end) the experiment conducted on a 3920 image database.

The proposed contributions have been validated and supported by analyzes showing their advantages and limitations. The results are encouraging and open new research perspectives.

Keywords: recognition, handwritten Arabic characters, support vector machines, SVM, model selection, parameter optimization, kernel function, meta-heuristics, genetic algorithm.

Sommaire

Remerciement	I
Dédicace	II
Résumer.....	IV
Abstract	V
Sommaire	VI
List des figures	XII
List des tableaux.....	XIV
Introduction générale.....	01
1. CHAPITRE I : Reconnaissance des caractères arabes manuscrits	
1.1.Introduction	03
1.2.Classes de reconnaissance des caractères.....	03
1.2.1. L'écriture en-ligne	03
1.2.2. L'écriture hors-ligne	04
1.3.Alphabet arabe : Données graphiques	04
1.3.1. Définition de la notion de fonte	04
1.3.2. Notion de la chasse	05
1.3.3. Notion du corps	05
1.4.Caractéristiques de l'écriture arabe	06
1.5.Problèmes liés à la reconnaissance de l'écriture arabe	08
1.5.1. Disposition spatiale du texte	08
1.5.2. Nombre de scripteurs	09
1.5.3. Taille du vocabulaire	09
1.5.4. Variations propres au scripteur	10
1.5.5. Variations propres à l'écriture manuscrite	10

Sommaire

1.6.Approches de reconnaissance	10
1.6.1. Approche globale	10
1.6.2. Approche analytique	11
1.7.Organisation générale d'un système de reconnaissance	11
1.7.1. Acquisition de l'image	12
1.7.2. Prétraitement	13
1.7.2.1.Binarisation de l'image	13
1.7.2.2.Elimination du bruit	13
1.7.2.3.Normalisation de l'image	14
1.7.2.4.Squelettisation (Thinning).....	14
1.7.3. Segmentation des caractères	14
1.7.3.1.Segmentation implicite.....	15
1.7.4. Segmentation explicite.....	15
1.7.5. Extractiondes caractéristique	16
1.7.6. apprentissage	17
1.7.6.1.Apprentissage supervisé	17
1.7.6.2.Apprentissage non supervisé	17
1.7.7. La décision	17
1.7.7.1.Poste traitement.....	17
1.8.Méthodes de classification	18
1.8.1. L'apprentissage statistique	19
1.8.2. Réseaux de neurones	19
1.8.3. Les arbres de décision	19
1.8.4. Régression	20
1.8.5. Clustering	20
1.8.6. Support Vecteur Machine (Machines à Vecteur de Support).....	20
1.9.Conclusion	21

Sommaire

2. CHAPITRE II : Machines à Vecteurs de Support

2.1.Introduction	22
2.2.Formulation	23
2.3.Pourquoi les Machines à Vecteurs de Support (SVM) ?.....	23
2.4.Apprentissage statistique et SVM	24
2.5.SVM principe de fonctionnement général	24
2.5.1. Notions de base: Hyperplan, marge et support vecteur	24
2.5.2. Linéarité et non-linéarité	25
2.5.3. Illustration de transformation de cas non linéaire : le cas XOR	26
2.6.Fondements mathématiques	27
2.6.1. Problème d'apprentissage	27
2.6.2. Classification à valeurs réelles	27
2.6.3. Complexité	27
2.6.4. La marge	28
2.6.4.1.Pourquoi maximiser la marge ?	28
2.6.4.2.Maximisation de la marge.....	28
2.6.4.3.Cas séparable	29
2.6.4.4.Cas non séparable	31
2.7.Les noyaux	33
2.7.1. Idée générale.....	33
2.7.2. Formulation (cas non linière)	33
2.7.3. Les fonctions noyaux.....	33
2.7.3.1.Propriétés des fonctions noyaux.....	34
2.7.3.2.L'astuce des fonctions noyaux	34
2.7.3.3.Relation entre les distances	34
2.7.3.4.Quelques fonctions noyaux.....	35
2.8.Architecture générale d'une machine à vecteur support	35
2.8.1. SVMs multiclasse	36
2.8.1.1.Une-contre-reste (1vsR).....	37
2.8.1.2.Une-contre-une (1vs1)	37

Sommaire

2.8.2. SVM monoclasse (Novelty detection).....	38
2.9. Les domaines d'applications	38
2.9.1. Exemple d'application.....	39
2.10. Systèmes de reconnaissance d'écriture à base de SVM	39
2.10.1. Système de Ayat, 2004	39
2.10.2. Système de Bellili, 2001	40
2.10.3. Système de Bahlman, 2002.....	40
2.11. Avantages et désavantages des SVM	41
2.12. Conclusion	41

3. CHAPITRE III : Sélection de modèle de machines à vecteurs de support

3.1. Introduction	43
3.2. Les classifieurs SVM et leurs hyper paramètres pour la sélection de modèle	43
3.3. Méthodes de sélection de modèle SVM	44
3.3.1. Technique de validation	44
3.3.2. Technique de validation généralisé approché (GACV)	44
3.3.3. Technique leave-one-out (LOO)	45
a) principe	45
b) algorithme	45
3.3.4. Erreur empirique	45
3.3.5. Grille de recherche	48
3.4. Stratégies d'optimisation	48
3.4.1. Optimisation des paramètres avec l'apprentissage incrémental.....	48
3.4.1.1. Technique de partition fixe.....	48
3.4.1.2. Technique basée sur l'erreur de classification.....	48
3.4.1.3. Technique basée sur le dépassement de la marge	49
3.4.2. Optimisation avec Méta-heuristiques.....	49
3.4.2.1. Définition	49
3.4.2.2. Notions et principes de fonctionnements.....	49
3.4.3. Présentation de quelque méta-heuristiques.....	50

Sommaire

3.4.3.1.Recherche avec tabous	50
a. principe de la méthode.....	50
b. Algorithme	51
3.4.3.2.Recuit simulé.....	51
a. Principe	51
b. Algorithme	52
3.4.3.3.Colonie de fourmi	52
3.4.3.4.Essaim de particulaire	53
3.4.3.5.Algorithme génétique.....	53
a. Principe	53
b. algorithme	54
c. Représentation d'une solution	54
d. Opérateur de base.....	54
3.4.3.6.Optimisation multi-objectif.....	59
3.5.Conclusion	59

4. CHAPITRE IV : Conception et implémentation

4.1.Introduction	60
4.2.Outils d'implémentation.....	60
4.2.1. Environnement logiciel.....	60
4.2.2. Bibliothèque SVM ^{multi classe} utilisée.....	61
4.3.La description de base des données utilisée	62
4.4.Description de notre système proposé LACDS.....	64
4.4.1. L'acquisition.....	65
4.4.2. Prétraitement.....	65
a. Binarisation.....	66
b. Encadrement	66
c. Normalisation (Adaptation)	67
4.4.3. Extraction de caractéristiques	67
a. Construction de la matrice de distribution.....	67
b. Vectorisation.....	69
c. Base de modèle	69
4.4.4. Apprentissage	71
4.4.5. Description d'algorithme génétique	71

Sommaire

4.4.5.1.Codage	72
4.4.5.2.Configuration d'AG	72
4.4.5.3.Algorithme génétique utilisé au système LACDS	73
4.5.Test, résultat et discussions	73
4.5.1. Résultat obtenue.....	73
4.5.2. Évaluation de système LACDS	76
4.6.Conclusion.....	79
Conclusion générale	80
Bibliographie	82

Liste des figures

Figure 1.1: Exemples des systèmes en-ligne.	04
Figure 1.2: Exemple de groupes de caractères ayant le même corps mais avec un nombre et/o emplacement de diacritiques différents.	07
Figure 1.3 : Exemples de caractères avec boucles.	07
Figure 1.4 : Différents styles et fontes pour l'écriture arabe.	08
Figure 1.5: Graphe de complexité des systèmes de reconnaissance.	09
Figure 1.6: Exemples d'allographes des caractères arabes.	10
Figure 1.7: Dichotomie des types d'écriture	12
Figure 1.8 : Exemple des caractères manuscrits normalisés.	14
Figure 1.9 : Exemple d'un mot arabe segmenté en caractères	15
Figure 1.10 : Apprentissage supervisé d'une machine.	18
Figure 2.1 : Arbre de classification des méthodes d'apprentissage à base de noyaux.	22
Figure 2.2 : Exemple d'un hyperplan séparateur.	24
Figure 2.3 : Exemple de vecteurs de support.	25
Figure 2.4 : Exemple de marge maximal (hyperplan valide).	25
Figure 2.5: a) Cas linéairement séparable, b) Cas non linéairement séparable	26
Figure 2.6: Exemple de changement de l'espace de données.	26
Figure 2.7: Illustration de cas non linéairement séparable (le cas XOR).	27
Figure 2.8: marge de la distance entre les 2 classes.	29
Figure 2.9: l'erreur en cas des données séparables.	31
Figure 2.10: données intrinsèquement non séparable.	31
Figure 2.11: données non séparable au plan de deux démentions	32
Figure 2.12. Architecture d'une machine à vecteur de support	36
Figure 2.13: Architecture du système en stratégie Un-contre-Tous	37
Figure 2.14 : Architecture du système en stratégie Une-contre-Une	38
Figure 2.15. Séparation des exemples d'une classe du reste de l'espace	38
Figure 3.1: Sélection de modèle basée sur l'erreur empirique.	47
Figure 3.2: cycle d'évolution.	53
Figure 3.3 : les étapes de sélection d'une Nouvelle population	55
Figure 3.4: Les couples à croiser	56

Liste des figures

Figure 3.5: Croisement entre deux parents avec 2 points de coupure pour la production de deux enfants par effet symétrique.	57
Figure 3.6: Exemple de Scramble Mutation	58
Figure 4.1 : Illustration de la relation entre notre application et SVM Multi classe ...	61
Figure 4.2 : Fenêtre principale de notre application LACDS.....	62
Figure 4.3: Décomposition de la base des caractères	62
Figure 4.4: Des exemples de notre base de données	63
Figure 4.5: Architecture du système LACDS	65
Figure 4.6 : Exemple de binarisation d'une image.....	66
Figure 4.7 : Exemple d'encadrement d'une lettre dans une image	66
Figure 4.8: Exemple de normalisation d'une lettre	67
Figure 4.9 : Exemple de matrice de distribution (5*5) de la lettre "Jim"	68
Figure 4.10: Exemple de vecteur des caractéristiques	69
Figure 4.11 : Modèle d'apprentissage	69
Figure 4.12: la création de corpus par le système LACDS	70
Figure 4.13 : Forme de base des données.....	70
Figure 4.14: relation entre SVM ^{multiclass} et notre application	71
Figure 4.15: structure d'individu	72
Figure 4.16: permet les interfaces de sélection des paramètres par AG (noyau RBF)	74
Figure 4.17: interface de test séparable de notre système LACDS	77
Figure 4.18: processus d'itération de l'AG pour l'optimisation simultanée (Génération/ taux de reconnaissance)	78

Liste des tableaux

Tableau.1.1 : Les 28 lettres de l’alphabet arabe avec leurs différentes formes.	05
Tableau.1.2 : Comparaison des caractéristiques de différentes langues	06
Tableau.1.3 : Les quatre formes des caractères "ain" et "ha" en fonction de leurs positions dans la chaîne de caractères	07
Tableau 4.1 : Description de la base d’apprentissage	63
Tableau 4.2 : Description de la base de test indépendant	64
Tableau 4.3 : Description des paramètres utilisés pour l'AG.....	72

Introduction générale

Depuis plusieurs années, de nombreux travaux de recherche ont été portés sur la reconnaissance des caractères manuscrits surtout les écritures latines. Dans ce mémoire, nous intéressons à la reconnaissance des caractères arabes manuscrits on - ligne qui reste aujourd'hui un thème de recherche ouvert. En effet, bien que le nombre de classes naturelles soit grand on trouve à l'intérieur de chacune d'entre elles, une très grande variabilité de l'écriture. De plus, les conditions souvent relativement précaires dans les quelles sont écrits les caractères et la variabilité du matériel utilisé (utilisation de divers stylos, de différentes qualités de papier) tendent à complexifier la reconnaissance.

A l'heure actuelle, les problèmes de l'écriture latine manuscrite contrainte sont partiellement résolus, et la lecture automatique a fait une grande avancée dans beaucoup de domaines. Le retard de l'écriture arabe par rapport à l'écriture latine peut être attribué à la complexité morphologique de l'alphabet arabe. De plus, le manque de protocoles communs de validation et de tests contribue énormément à ce retard. Depuis, les recherches se sont multipliées dans ce domaine. Durant ces dernières décennies, plusieurs approches et méthodes ont été proposées par les chercheurs, dans le but d'améliorer les taux de reconnaissance. C'est dans ce cadre que s'inscrit notre travail.

La variabilité de l'écriture manuscrite permet de confronter les algorithmes de classification et d'apprentissage à des problèmes difficiles et réalistes. La plupart des classifieurs classiques ont donné des résultats remarquables dans ce domaine, notamment les réseaux de neurones. Mais la nécessité de performances élevées dans des applications réelles, a poussé la recherche vers des modèles de classifications de plus en plus complexes.

De nombreux travaux ont démontré la supériorité du SVM sur les méthodes discriminantes classiques. Sa robustesse vis-à-vis de la dimensionnalité des données et son pouvoir accru de généralisation, font que le SVM est nettement plus avantageux. Les SVM représentent une méthode de classification bien adaptée pour traiter des données de très hautes dimensions telles que les textes et les images. La formulation des SVM laisse très peu de place aux paramètres fixés par l'utilisateur. De plus, il s'agit d'un problème d'optimisation, ce qui est assez complexe (en termes algorithmiques) mais donne des garanties sur la convergence à un optimum global,

Introduction générale

L'usage des algorithmes génétiques convient parfaitement pour la réalisation d'un système d'optimisation car il est un concept basé sur la théorie de l'évolution des espèces et comprend par le fait même des aptitudes d'adaptation très fortes. Leur principe consiste à faire évoluer une population des individus parents pour obtenir des fils, à l'aide des opérateurs génétiques ; croisement et mutation. Chaque individu code une solution au problème abordé, sa pertinence à la résolution est mesurée par une fonction d'adaptabilité dite fitness. Cette procédure permet de passer d'une génération à une autre plus évoluées jusqu'à arriver à une solution optimale. L'algorithme génétique assure à la fois l'exploration d'un espace de recherche et l'exploitation des solutions déjà explorées.

L'objectif le plus important visé par ce travail est la sélection méta-heuristique du modèle SVM pour la reconnaissance des caractères arabes manuscrits. Nous avons utilisé une approche basée sur les algorithmes génétiques afin d'avoir un taux de reconnaissance appréciable.. Il est organisé en quatre chapitres. Le premier chapitre présente les différents blocs, schéma standard de reconnaissance des caractères manuscrits. Le deuxième chapitre décrit en détail le concept ainsi que la formulation mathématique des SVMs. Le troisième chapitre décrit la sélection des paramètres SVM aussi nous avons présenté dans ce chapitre la stratégie utilisée pour la sélection et l'optimisation des paramètres du modèle SVM. Et le dernier décrit l'implémentation et l'évaluation de système. Dans la conclusion, nous rappelons l'objectif ainsi que les perspectives de ce travail.

Chapitre I

Reconnaissance des caractères

Manuscrits

1.1. Introduction

La variabilité de l'écriture manuscrite permet de confronter les algorithmes de classification et d'apprentissage à des problèmes difficiles et réalistes. Un système de reconnaissance de l'écriture doit idéalement, localiser, reconnaître et interpréter n'importe quel texte ou nombre écrit sur un support de qualité arbitrairement variable tels que les cartes, les formulaires, les agendas, les vieux manuscrits, etc.

Ce type de système aide, dans ce cas, à convertir l'information contenue dans les vieux manuscrits pour être enregistrée dans des bases de données que tout le monde peut interroger et consulter à travers l'internet par la suite, on s'est vite aperçu de la difficulté de réaliser des systèmes capables de reconnaître tout type de texte et on s'est tourné vers des systèmes dédiés, où l'on connaît à priori le style d'écriture que le système doit traiter. Par exemple, il existe des systèmes pour la lecture automatique des chèques bancaires, la reconnaissance du code de l'adresse postale, la lecture des formulaires d'impôts,...etc.

1.2. Classes de Reconnaissance des Caractères

On classe souvent les méthodes de reconnaissance en fonction du mode d'acquisition de l'écriture :

1.2.1. L'écriture en-ligne

Est obtenue en saisie continue et se présente sous la forme d'une séquence de points ordonnée dans le temps avec un tracé est sans épaisseur [11]. Les systèmes en ligne prennent en compte l'information chronologique des mouvements du bras du scripteur. Cette information additionnelle augmente la précision bien que souvent coûteuse en temps de calcul.

Dans ce cas, la donnée est de type signal où la reconnaissance est effectuée sur des données à une seule dimension ; l'approche doit tirer profit du lever du stylo et de la représentation temporelle. De plus, la réponse en continu du système permet à l'utilisateur de corriger et de modifier son écriture de manière interactive. L'analogie avec la reconnaissance de la parole est très fréquente et il n'est pas rare de voir des chercheurs appliquer des techniques issues de ce domaine. L'acquisition du tracé est assurée généralement par une tablette graphique munie d'un stylo électronique.



Figure 1.1 : Exemples des systèmes en-ligne.

1.1.2. L'écriture hors-ligne

Est obtenue par la saisie d'un texte déjà existant, obtenue par un scanner ou une caméra. Dans ce cas, on dispose d'une image binaire ou en niveaux de gris, ayant perdu toute information temporelle sur l'ordre des points. De plus, ce mode introduit une difficulté supplémentaire relative à la variabilité du tracé en épaisseur et en connectivité, nécessitant l'application de techniques de prétraitement [10].

1.3. Alphabet arabe : Données graphiques

L'alphabet arabe n'a qu'un système d'écriture dans lequel les lettres sont liées ou ne sont pas liées entre elles selon des règles précises. Il existe différents styles d'écriture, mais dans aucun d'eux il est possible de juxtaposer des lettres totalement isolées les unes des autres. Il n'y a que des caractères typographiques copiés de l'écriture manuscrite. Le caractère arabe est en effet dessiné non pas en fonction des contraintes géométriques des procédés de composition pour imprimerie, mais en fonction de la main et d'une esthétique visuelle héritée de la calligraphie. La fonctionnalité et la lisibilité sont sacrifiées à l'esthétique calligraphique qui substitue l'élégance à la clarté.

1.3.1. Définition de la notion de fonte

Une police (fonte) est un ensemble de caractères d'une même famille, d'une même graisse et pour un corps donné. Ces caractéristiques typographiques sont normalisées dans l'imprimerie, tant au niveau du symbole (dimensions et dessin qui représente la forme et l'épaisseur du caractère), qu'à celui de la chaîne (mot : suite de symboles appartenant à la même fonte ou des fontes compatibles) dans chaque ligne du texte[1].

1.3.2. Notion de la chasse

La chasse est l'espace qu'occupe un caractère d'imprimerie. Elle dépend du dessin, du style et de la grosseur du caractère, et comprend en plus de la largeur, l'espace entre caractères [1].

1.3.3. Notion du corps

Le corps désigne la hauteur d'un caractère typographique comprenant le blanc de séparation horizontale avec la ligne au dessus. Le corps varie en fonction de l'usage prévu pour le caractère

Caractère	Début	Milieu	Fin	Isolé
Alif			ا	ا
Ba	بـ	بـ	بـ	بـ
Tha	ثـ	ثـ	ثـ ou تـ	تـ ou ةـ
Nun	نـ	نـ	نـ	نـ
Ya	يـ	يـ	يـ	يـ
Jim	جـ	جـ	جـ	جـ
Ha	حـ	حـ	حـ	حـ
Kha	خـ	خـ	خـ	خـ
Dal			دـ	دـ
Thé			ذـ	ذـ
Ra			رـ	رـ
Za			زـ	زـ
Waw			وـ	وـ
Sin	سـ	سـ	سـ	سـ
Chin	شـ	شـ	شـ	شـ
Sad	صـ	صـ	صـ	صـ
Dhad	ضـ	ضـ	ضـ	ضـ
Tad	طـ	طـ	طـ	طـ
Dha	ظـ	ظـ	ظـ	ظـ
Ayn	عـ	عـ	عـ	عـ
Ghayn	غـ	غـ	غـ	غـ
Fa	فـ	فـ	فـ	فـ
Qaf	قـ	قـ	قـ	قـ
Kaf	كـ	كـ	كـ	كـ
Lam	لـ	لـ	لـ	لـ
Mim	مـ	مـ	مـ	مـ
He	هـ	هـ	هـ	هـ

Tableau 1.1: Les 28 lettres de l'alphabet arabe avec leurs différentes formes.

1.4. Caractéristiques de l'écriture arabe

L'écriture arabe imprimée ou manuscrite possède des caractéristiques différentes d'autres langues en structure et en mode de liaison entre les caractères formant un mot, ce qui rend l'application directe des techniques de reconnaissance développées pour les caractères chinois ou latins, par exemple, une tâche délicate pour leur reconnaissance[6].

<i>Caractéristiques</i>	<i>Arabe</i>	<i>Latine</i>	<i>Hebrue</i>	<i>Hindou</i>
<i>justification</i>	D à G	G à D	D à G	G à D
<i>Cursivité</i>	Oui	non	non	Oui
<i>diacritiques</i>	Oui	non	non	Oui
<i>Nombre de voyelles</i>	3	5	11	-
<i>Nombre de lettres</i>	28	26	22	40
<i>Formes des lettres</i>	1-4	2	1	1

Tableau 1.2: Comparaison des caractéristiques de différentes langues.

L'écriture arabe est à la base de nombreuses autres langues telles que le Perse et certains dialectes d'Afrique, d'Inde, d'Indonésie, de Chine et de Turquie [12]. L'écriture arabe utilise 28 lettres (Tableau 1.2) auxquelles il faut ajouter « El hamza ء » qui est le plus souvent considéré comme un caractère à part entière.

El hamza a une orthographe spéciale qui dépend de règles grammaticales, ce qui multiplie les formes nécessaires à sa représentation puisqu'elle peut s'écrire seule ou sur le support de trois lettres "alif, waw, ya". De plus, l'alphabet arabe comprend d'autres caractères additionnels tels que « ؤ » et « ڤ ». La considération du symbole « ~ » qui s'écrit uniquement sur le support du caractère « ٲ », fait apparaître d'autres graphismes. L'écriture arabe a ainsi plusieurs spécificités, nous citons :

- Un trait caractéristique de l'écriture arabe est la présence d'une ligne de base horizontale dite ligne de *référence* ou *d'écriture*. C'est le lieu de ligatures horizontales des caractères d'une même chaîne ;
- Les caractères arabes s'écrivent cursivement de la droite vers la gauche, aussi bien dans le cas de l'imprimé que du manuscrit ;

- L'écriture arabe est une écriture calligraphique, elle varie selon les milieux et les régions, d'une extrême simplicité formelle à la complexité exhaustive de l'arabesque. Il existe une centaine de styles dont seulement quelques uns sont couramment utilisés dans le monde arabo-musulman, nous citons par exemple : Tholothi, Neskhi, Requeh, Dewani, Farci et Koufique. La figure (1.4) suivante montre un exemple de différents styles graphiques de l'écriture arabe.

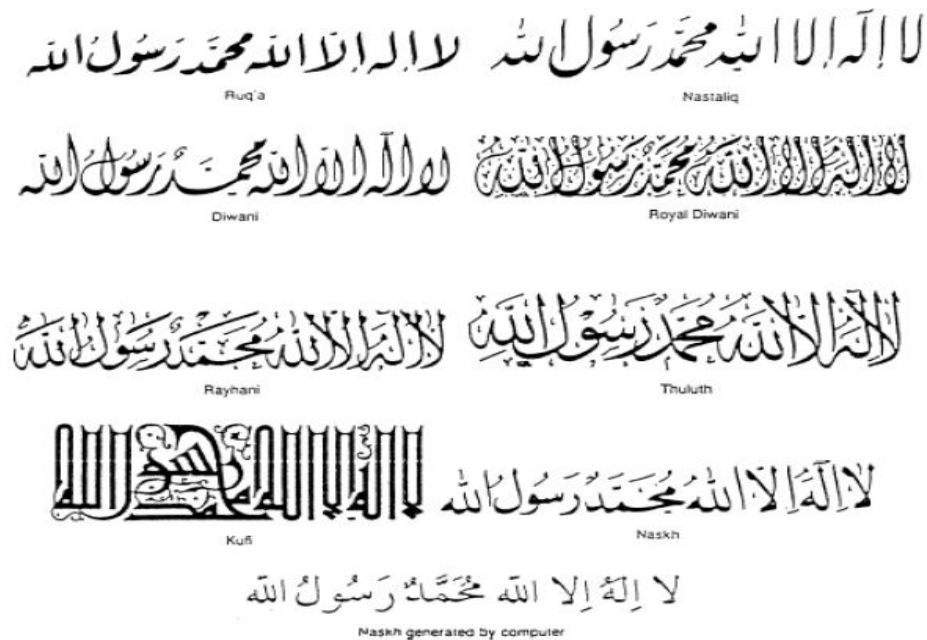


Figure 1.4: Différents styles et fontes pour l'écriture arabe.

1.5. Problèmes liés à la reconnaissance d'écriture arabe

Divers problèmes compliquent le processus de reconnaissance, ce qui rend la tâche de l'écriture très délicate, parmi ces problèmes nous pouvons citer :

- Disposition spatiale du texte,
- Nombre de scripteurs,
- Taille du vocabulaire.

1.5.1. Disposition spatiale du texte

La classification de Tappert [7] indique que la présentation du texte peut subir deux types de contraintes : externes conduisant à une écriture pré casée, zonée, guidée ou générale; et internes provenant des habitudes propres à chaque scripteur et conduisant à une écriture détachée, groupée, script (bâton), purement cursive ou mixte. Il est évident que l'écriture

détachée reste la plus facile à réaliser du fait de la séparation quasi immédiate des lettres ; Au contraire, l'écriture cursive nécessite plus d'efforts du fait de l'ambiguïté des limites entre les lettres.

1.5.2. Nombre de scripteurs

La difficulté de reconnaissance croît avec ce nombre, divisant l'échelle en trois : mono, multi et omni scripteurs. En multi scripteur, le système doit s'adapter à l'écriture de plusieurs scripteurs, tandis qu'en omni scripteur, le système doit être capable de généraliser son apprentissage à n'importe quel type d'écriture [7].

1.5.3. Taille du vocabulaire

On fait la différence entre les applications à vocabulaire limité (< 100 mots) et celles à vocabulaire très étendu (> 10 000 mots). Il est évidente que dans le premier cas, la complexité est moindre, car la réduction du nombre limite l'encombrement mémoire et favorise l'utilisation de méthodes de reconnaissance directes et donc rapides, par balayage systématique de l'ensemble des mots du lexique [7]. La figure 1.5 montre la complexité des systèmes de reconnaissance de l'écriture.

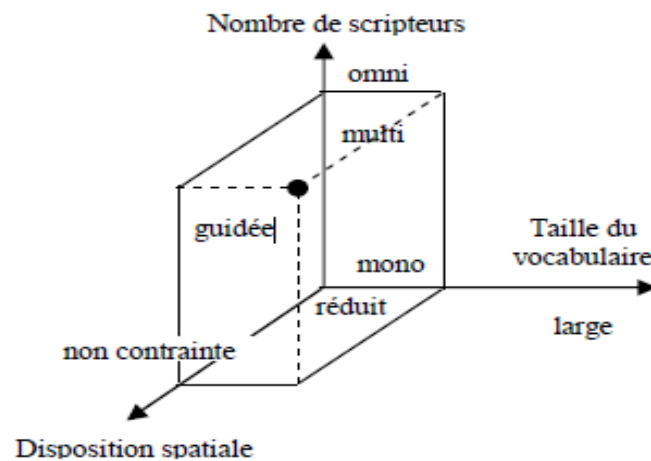


Figure 1.5: Graphe de complexité des systèmes de reconnaissance.

D'autres types de critères peuvent influencer la complexité des systèmes de reconnaissance. Ils sont relatifs aux variations intrinsèques de l'écriture dans un contexte d'écriture cursive. Parmi ces variations nous pouvons noter celles :

- propres au scripteur,
- propres à l'écriture manuscrite,

1.5.4. Variations propres au scripteur

Les variations propres au scripteur traduisent le style personnel en termes de rapidité, de continuité et de régularité. Tous ces éléments influent sur la forme des lettres (écriture penchée, bouclée, arrondie, linéaire, etc.) et bien sûr sur la forme des ligatures, compromettant parfois le repérage des limites entre lettres [7].

1.5.5. Variations propres à l'écriture manuscrite

La forme d'une lettre dépend de sa position dans le mot (début, milieu, fin) ainsi que des lettres voisines. Toutes ces variations vont conduire à des formes morphologiques (dessins) différentes d'une même lettre, appelées allographes [8]. La figure 1.6 donne des exemples d'allographes.

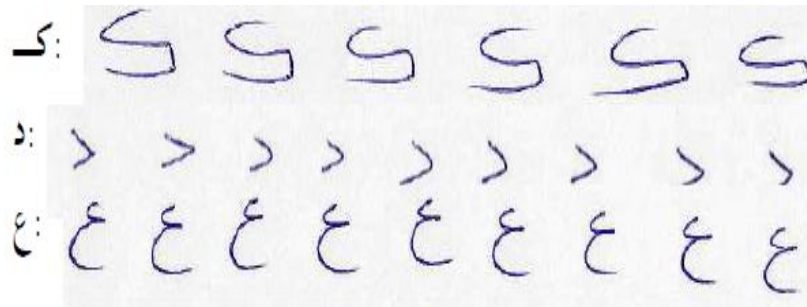


Figure 1.6: Exemples d'allographes des caractères arabes.

1.6. Approches de reconnaissance

Il existe deux approches pour la reconnaissance des mots : la reconnaissance globale et la reconnaissance analytique.

1.6.1. Approche globale

Dite encore holistique, a une vision générale du mot ; elle se base sur une description générale de l'image du mot. Elle considère le mot comme une seule entité indivisible et le décrit indépendamment des caractères qui le constituent. Cette approche présente l'avantage de garder le caractère dans son contexte avoisinant, ce qui permet une modélisation plus efficace des variations de l'écriture et des dégradations qui peuvent l'entacher. Cependant, cette méthode est pénalisée par la taille mémoire, le temps de calcul et la complexité du traitement, qui croissent linéairement avec la taille du lexique considéré, d'où une limitation du vocabulaire. Cette approche est souvent appliquée pour réduire la liste de mots candidats dans

le contexte d'une reconnaissance à grand vocabulaire. Il est nécessaire, dans ce cas, d'utiliser des primitives très robustes, comme dans les travaux de Govindaraju [13], pour ne pas manquer le mot réel parmi les candidats. Le mot est ensuite trouvé à l'aide de primitives de plus en plus précises (ou d'un classifieur de plus en plus fin).

1.6.2. Approche analytique

Permet de s'affranchir des limites de l'approche globale, mais nécessite une interprétation locale basée sur une segmentation du mot. La reconnaissance consiste à identifier les entités segmentées puis tendre vers une reconnaissance du mot, ce qui constitue une tâche assez délicate pouvant générer différents types d'erreurs [12]. La difficulté d'une telle approche a été clairement évoquée par Sayre en 1973 et peut être résumée par le dilemme suivant « *pour reconnaître les lettres, il faut segmenter le tracé et pour segmenter le tracé, il faut reconnaître les lettres* ».

1.7. Organisation générale d'un système de reconnaissance

La reconnaissance de l'écrite manuscrite s'intéresse à identifier correctement l'entrée d'une image du texte écrit sur papier scannée ou photographié, en la convertissant en un texte sous forme d'un fichier informatique en format d'édition telle HTML ou Latex . Typiquement, quelque soit le système de reconnaissance du manuscrit, il fait appel des phases suivantes :

- Acquisition (scanning, Numérisation),
- Prétraitement,
- Segmentation à des caractères séparés ou segments reliés à un caractère,
- Extraction des caractéristiques,
- Classification, suivis éventuellement d'une phase de post-traitement.

Les applications de la reconnaissance de caractères peuvent varier substantiellement selon la nature de l'écriture et son support de saisi. Dans cette étude, nous nous intéressons à la reconnaissance on-ligne des caractères manuscrits. Les différentes applications de la reconnaissance de caractères ne peuvent être dissociées de façon exclusive tant elles partagent plusieurs caractéristiques.

La reconnaissance de caractères, en effet, signifie le décodage de n'importe quel texte imprimé ou manuscrit, ou information symbolique incluant par exemple les formules mathématiques, le dessin technique, etc. De ce point de vue, le moyen utilisé pour transcrire l'information et la traiter est sans importance. Donc, la représentation des caractères n'est pas

restreinte aux seules images comme c'est le cas dans la reconnaissance en-ligne. La figure 1.7 présente une classification des différents types d'écriture faite selon le média, le support de saisi et l'application considérés. La distinction entre le texte imprimé et le texte manuscrit bien qu'évidente n'est pas toutefois exclusive. Un système qui reconnaît du manuscrit doit être capable de reconnaître de l'imprimé.

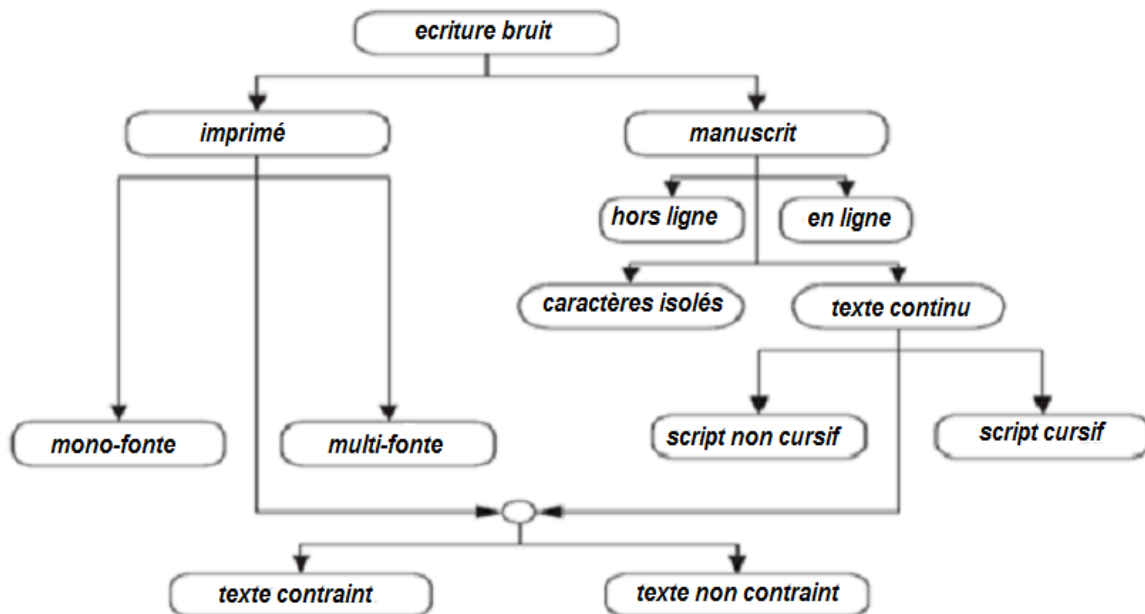


Figure 1.7: Dichotomie des types d'écriture [1].

La démarche classique suivie en reconnaissance de caractère manuscrit consiste à opérer selon les étapes suivantes :

1.7.1. Acquisition de l'image

L'écriture est digitalisée par un scanner pour être transformée en une image. Ceci correspond à l'entrée du système. Cette étape est assez simple mais très importante car elle influence sérieusement les étapes suivantes [6]. Deux paramètres importants doivent être pris en compte :

- Résolution : la résolution normale est de 300 dpi (dot per inch : points par pouce). Cependant, dans certains cas, lorsque la taille de l'écriture est petite, il est nécessaire d'augmenter la résolution pour tenir compte de la finesse de l'écriture.

- Niveau d'éclairage : l'éclairage influe également sur la qualité de la reconnaissance. Trop faible, le bruit devient important. Trop fort, le bruit est réduit mais les traits fins disparaissent.

1.7.2. Prétraitement

Lorsque l'acquisition est réalisée, la plupart des systèmes comportent une étape de prétraitement. Généralement, ces prétraitements ne sont pas spécifiques à la reconnaissance de texte, mais sont des prétraitements classiques en traitement d'image. Le prétraitement a pour but de préparer l'image du tracé à la phase suivante d'analyse.

Il s'agit essentiellement de réduire le bruit superposé aux données et ne garder, autant que possible, que l'information significative de la forme présentée. Le bruit peut être dû au dispositif d'acquisition, aux conditions d'acquisition (éclairage, mise incorrecte du document...), ou encore à la qualité du document d'origine. Parmi les opérations de prétraitements généralement utilisées, citons : la binarisation, le redressement de l'écriture, le lissage, la squelettisation et la normalisation.

1.7.2.1. Binarisation de l'image

Elle permet de passer d'une image de niveaux de gris à une image binaire composée de deux valeurs 0 et 1, plus simple à traiter. En général, on utilise un seuil de binarisation approprié qui traduit la limite des contrastes fort et faible dans l'image.

1.7.2.2. Elimination du bruit

L'image des caractères peut être entachée de bruit introduit durant l'acquisition et au cours des différentes transformations. Ce bruit correspond soit à des absences de points (trous), soit à des empâtements et donc à une surcharge de points. Le lissage consiste à examiner le voisinage d'un pixel et éliminer les pixels isolés d'une part (*nettoyage*).

Une méthode fréquemment utilisée pour le lissage, consiste à parcourir l'image pixel par pixel, en utilisant une fenêtre de taille 3×3 et à changer la valeur du pixel en se basant sur les valeurs de ses 8 voisins. Un point 1 est mis à 0 s'il n'y a pas assez de points noirs autour de lui et vice versa. Une méthode statistique similaire consiste à inverser le pixel si la somme des pixels voisins est inférieure à un seuil déterminé. Cependant, le choix du seuil est critique : un faible seuil peut éliminer aussi bien les imperfections que les discontinuités naturelles et discriminantes dans une lettre ; par exemple ; un « μ » peut facilement se transformer en un « ϵ » à la suite d'un mauvais choix du seuil.

1.7.2.3. Normalisation de l'image

Pour faciliter l'étape de reconnaissance, les caractères segmentés doivent être normalisés à une taille fixée. La contrainte principale réside dans le choix de cette taille. Trop petite, il y a un risque d'une perte d'information. Trop grande, l'étape de reconnaissance va opérer lentement [1].

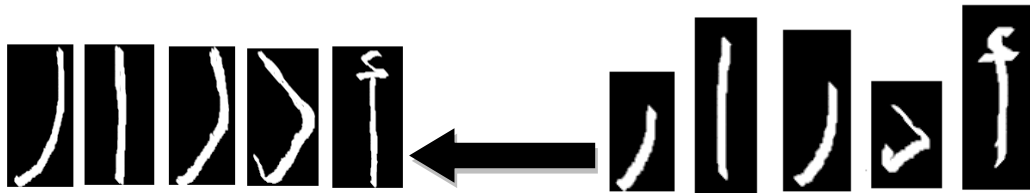


Figure 1.8: Exemple des caractères manuscrits normalisés.

1.7.2.4. Squelettisation (Thinning)

Comme l'image binaire se présente comme une succession de traits d'épaisseurs variables, mal définis, et généralement avec bruit, l'opération de squelettisation est appliquée pour simplifier l'image du caractère en une image à « ligne » plus facile à traiter en la réduisant au tracé du caractère [1].

Les points du tracé résultant constituent généralement les lignes centrales des points initiaux. Elle est définie comme étant l'équivalent de la forme, mais avec une épaisseur très réduite qui peut varier entre 1 et 2 pixels. La squelettisation (amincissement). Cependant, la squelettisation des caractères arabes peut induire en erreur : deux points diacritiques sont souvent confondus avec un seul point. Selon la qualité du document à traiter, le type d'écriture (manuscrite ou imprimée) et la méthode d'analyse adoptée, une ou plusieurs techniques de prétraitements sont appliquées.

1.7.3. Segmentation des caractères

La segmentation consiste à séparer les caractères les uns des autres. La difficulté de cette étape réside dans le fait que les caractères ne sont pas alignés et ne présentent pas la même taille, de plus, dans certains cas, les caractères se chevauchent. Par conséquent, il faut utiliser des techniques de séparation permettant d'isoler les caractères [1].



Figure 1.9:Exemple d'un mot arabe segmenté en caractères.

Généralement, la performance de la segmentation affecte directement la fiabilité du système global : outre des erreurs de confusion, certaines erreurs de segmentation peuvent engendrer même des rejets. Deux techniques ont été utilisées dans la reconnaissance des mots

1.7.3.1. Segmentation implicite

Consiste à segmenter le mot en parties inférieures aux lettres appelées *graphèmes* et à retrouver les lettres puis les mots par combinaison de ces graphèmes.

1.7.3.2. Segmentation explicite

Consiste à segmenter le mot exactement en caractères en utilisant des propriétés générales de l'écriture d'un pseudo mot. Dans cette technique, la segmentation devient l'étape la plus critique dans le processus de reconnaissance, car une erreur à ce niveau conduira automatiquement à une classification erronée. Ce type de segmentation est fréquemment illustré par des règles identifiant les points de segmentation des caractères[1].

Le problème de segmentation des mots latins qu'ils soient imprimés ou manuscrits a été largement étudié malgré qu'il soit possible parfois, d'appliquer les résultats de ces études pour l'arabe, en général ils ne sont pas suffisants pour segmenter les mots arabes. Un caractère arabe peut être constitué de plus d'une partie et la nature cursive de l'écriture arabe rend l'utilisation de ces méthodes insuffisante.

Plusieurs méthodologies de segmentation des mots arabes, explicite ou implicite, ont été développées utilisant des techniques variées (histogrammes de projection, contour, squelette, fenêtre glissante...). Cependant la segmentation reste un problème ouvert.

1.7.4. Extraction des caractéristiques

Pour la prise de décision, un système de reconnaissance n'a besoin que de l'information pertinente pour différencier un objet d'un autre. Dans ce but, une étape d'extraction de caractéristiques est réalisée. C'est une phase critique lors de la construction d'un système de reconnaissance. Dans la littérature les caractéristiques employées pour la reconnaissance de l'écriture arabe comprennent :

- le nombre de traits (strokes), leurs tailles, directions et pentes;
- les points extrémaux (end points);
- la hauteur et la largeur du caractère;
- la catégorie de la forme (partie primaire, point diacritique, etc);
- le nombre de points diacritiques et leurs positions par rapport à la ligne de base, et les zigzags (Hamza's);
- les points de rebroussement et d'inflexion ;
- le nombre d'occlusions (boucles);
- les concavités et les convexités dans les quatre principales directions (ouest, nord, est, sud) ;
- les jambages (descendants) et les hampes (ascendants);
- la taille du rectangle englobant le tracé ou le caractère;
- le nombre de composantes connexes ... ;

1.7.5. Apprentissage

La décision nécessite de définir clairement la connaissance que nous avons sur les formes à traiter. Cette définition repose sur l'apprentissage qui se charge d'acquérir la connaissance et de l'organiser en classes ou modèles de références. La tendance consiste à remplacer ce nombre par une meilleure qualité des traits caractéristiques. Les procédés d'apprentissage sont différents selon qu'il s'agit de reconnaître des caractères imprimés ou manuscrits, ou de reconnaître un texte monofonte¹ ou multifonte² [14]. D'une manière générale, nous distinguons deux types de techniques d'apprentissage : *supervisé* et *non supervisé*.

¹ monofonte, ne peut reconnaître qu'une seule fonte à la fois.

² capable de reconnaître divers types de fontes parmi un ensemble de fontes préalablement apprises.

1.7.5.1. Apprentissage supervisé

Les différentes familles des formes sont connues à priori et la tâche d'apprentissage est guidée par un superviseur ou professeur. L'apprentissage est réalisé lors d'une étape préliminaire à la reconnaissance en introduisant un grand nombre d'échantillons de référence.

1.7.5.2. Apprentissage non supervisé

Appelé encore, suivant l'approche utilisée, classification automatique. Il consiste à doter le système d'un mécanisme automatique qui s'appuie sur des règles précises de regroupement pour trouver les classes de références.

1.7.6. La décision

Le processus de reconnaissance peut toujours se résumer à une décision de la classification. Pour cela, il faut choisir une représentation qui permette une description de l'objet à analyser (caractère traité), puis une règle de décision qui s'appuie sur cette description.

A partir de la description du caractère traité, le module de reconnaissance cherche, parmi les modèles de références en présence, ceux qui lui sont les plus "proches". La notion de proximité a un sens différent en fonction de la nature de la représentation et du type de la méthode utilisée.

Différentes approches de décision ont été proposées dans la littérature, mais, de façon classique, on peut distinguer cinq catégories :

- Approche structurelle ;
- Approche statistique ;
- Approche stochastique ;
- Approche connexionniste ;
- Approches hybrides.

1.7.7. Post-traitement

L'objectif du post-traitement est d'améliorer le taux de la reconnaissance. Il est responsable de sélectionner une solution parmi un ensemble en ayant recours à des informations de haut niveau (lexicales, syntaxiques, sémantiques, pragmatiques,...) qui ne sont pas disponibles au niveau du classifieur. Cette étape du processus est souvent implémentée comme un ensemble de techniques qui dépendent de plusieurs facteurs entre autres : les fréquences d'apparition des mots, les lexiques, et autres informations sur le contexte. Des vérifications contextuelles classiques, telles que la recherche dans un

dictionnaire, les probabilités d'occurrences de bigramme et de trigramme..., sont appliqués dans les différents travaux qui prévoient un post-traitement[2].

La méthode basée sur un dictionnaire est traditionnellement simplifiée pour accélérer la recherche et réduire la complexité de calcul. Le dictionnaire est construit à partir des mots réduits à leurs racines, les suffixes et les préfixes sont ainsi éliminés.

1.8. Méthodes de classification

Généralement, un système de reconnaissance est construit à partir des connaissances a priori sur le problème. Le fait de classer un objet correspond donc à prendre une décision sur base d'une ou plusieurs règles. Dans les années 90, l'approche *Machine Learning (ML)* ou apprentissage machine devient très populaire. Il s'agit d'apprendre automatiquement les règles de décision sur base d'un ensemble d'exemples déjà classés.

L'apprentissage machine inclut entre autres les méthodes supervisées et non supervisées. Dans les techniques non-supervisées, les objets sont présentés sans leur catégorie. Un exemple type de méthodes non-supervisées est le clustering dont le but est de créer des groupes (clusters) d'objets présentant des caractéristiques semblables. Dans le cas des méthodes supervisées, on cherche à estimer une fonction $f(x)$ à partir des exemples x , comme l'indique la figure ci-dessous[2].



Figure 1.10: Apprentissage supervisé d'une machine.

- $x, f(x)$ Une donnée et son étiquette.
- \hat{f} est une estimation de $f(x)$.

Les données préparées sont stockées dans des entrepôts de données qui sont directement accessibles par la phase d'extraction des connaissances. Dans cette phase, plusieurs techniques et outils sont utilisés selon les objectifs et les informations recherchées. Cependant, deux grandes tendances partagent ces techniques : la prédiction et la description.

La prédiction englobe une famille de méthodes qui visent à utiliser les données disponibles pour prédire le comportement des données futures. La description englobe des méthodes qui visent à découvrir les modèles (patterns) compréhensibles par les humains et qui décrivent les données analysées. La prédiction est un processus à deux étapes : une étape d'apprentissage (entraînement) et une étape de classification (utilisation). Dans l'étape d'apprentissage, un classifieur (une fonction, un ensemble de règles, ...) est construit en analysant (ou en apprenant de) une base de données d'exemples d'entraînement avec leurs classes respectives.

1.8.1. L'apprentissage statistique

Les techniques d'apprentissage statistique se basent sur les lois statistiques, ce sont des méthodes qui reposent sur la théorie de Bayes représentant une référence théorique pour les approches statistiques de résolution des problèmes de classification. Les réseaux Bayésiens (ou réseaux de croyance) représentent l'application la plus connue de la théorie de Bayes dans la classification. Ils sont beaucoup plus précis que d'autres techniques d'apprentissage, puisqu'ils, d'un côté, prennent en considération les dépendances entre les attributs, et d'un autre côté, peuvent intégrer des connaissances humaines au préalable. Malheureusement, ces réseaux sont très coûteux en temps de calcul pour l'apprentissage et l'espace nécessaire pour les stocker est aussi exhaustif [15].

1.8.2. Réseaux de neurones

Un réseau de neurones est un assemblage de neurones connectés entre eux. Un réseau réalise une ou plusieurs fonctions algébriques de ses entrées, par composition des fonctions réalisées par chacun des neurones. La capacité de traitement de ce réseau est stockée sous forme de poids d'interconnexions obtenus par un processus d'apprentissage à partir d'un ensemble d'exemples d'apprentissage. Il arrive souvent que les exemples de la base d'apprentissage comportent des valeurs approximatives ou bruitées. Si on oblige le réseau à répondre de façon quasi parfaite relativement à ces exemples, on peut obtenir un réseau qui est biaisé par des valeurs erronées.

1.8.3. Les arbres de décision

Les arbres de décision représentent une méthode très efficace d'apprentissage supervisé. Il s'agit de partitionner un ensemble de données en des groupes les plus homogènes possible du point de vue de la variable à prédire. On prend en entrée un ensemble de données classées, et on fournit en sortie un arbre qui ressemble beaucoup à un diagramme d'orientation où chaque

nœud final (feuille) représente une décision (une classe) et chaque nœud non final (interne) représente un test. Chaque feuille représente la décision d'appartenance à une classe des données vérifiant tous les tests du chemin menant de la racine à cette feuille [16].

Les arbres de décisions sont très répandus, à cause de la simplicité de lecture de leurs résultats et leur traitement naturels des cas multiclasse. Néanmoins, ils posent beaucoup de problèmes tel que :

- La difficulté de manipulation des attributs numériques,
- L'espace nécessaire pour leur déduction,
- Leur non scalabilité.

1.8.4. Régression

La régression est la méthode utilisée pour la prédiction des valeurs continues. Son objectif est de trouver le meilleur modèle qui décrit la relation entre une variable continue de sortie et une ou plusieurs variables d'entrée. Dans l'analyse des bases de données, la régression est utilisée pour prédire des informations continues telles que la valeur d'une action d'une entreprise dans une bourse [4].

1.8.5. Clustering

Le clustering regroupe un ensemble de techniques qui visent à regrouper les enregistrements d'une base de données en des groupes selon leur rapprochement les uns des autres en ne se basant sur aucune information antérieure, c'est un apprentissage non supervisé. Un système d'analyse en clusters prend en entrée un ensemble de données et une mesure de similarité entre ces données, et produit en sortie un ensemble de partitions décrivant la structure générale de l'ensemble de données [4].

La similarité entre les exemples peut être mesurées par la distance euclidienne, si les attributs sont numériques. Si les attributs sont symboliques, la seule mesure qui existe, dans l'absence de toute information sur la signification des valeurs, c'est l'égalité ou l'inégalité.

1.8.6. Support Vecteur Machine (Machines à Vecteur de Support)

Le SVM est un modèle discriminant qui tente de minimiser les erreurs d'apprentissage tout en maximisant la marge séparant les données des classes. La maximisation de la marge est une méthode de régularisation qui réduit la complexité du classifieur. Elle sert à pénaliser les

paramètres du modèle de la même façon que la méthode du «weight decay³» qui altère les poids de grande amplitude dans un PMC⁴. Le noyau d'un SVM est une fonction symétrique défini-positive qui permet de projeter les données dans un espace transformé de grande dimension dans lequel s'opère plus facilement la séparation des classes [17].

1.9. Conclusion

Dans ce chapitre, nous avons exposé les principales propriétés morphologiques de l'écriture arabe, ensuite nous avons présenté les différentes étapes de reconnaissance des caractères manuscrits hors-ligne en générale et les méthodes de classification fondés sur les machines.

De plus, nous avons vus les problèmes majeurs dans ce domaine qui se ramènent à la cursivité de l'écriture et à la sensibilité de certaines caractéristiques topologiques de l'arabe à la dégradation, en l'occurrence les points diacritiques et les boucles. Donc, la reconnaissance optique de caractères arabe reste une tâche encore non résolue.

Dans le système de reconnaissance, nous nous intéressons plus précisément au classifieur fondé sur les SVMs. Aussi, nous abordons dans le prochain chapitre la description détaillée de cette méthode de classification appliquées à la reconnaissance des caractères manuscrits.

Cette technique est appelé machines à vecteurs de support (SVM) qui appliquée avec une efficacité remarquable à la reconnaissance des caractères manuscrits, au traitement d'images, à la prédiction de séries temporelles, au diagnostic médical, au contrôle qualité, etc.

³ les réseaux de neurones multicouches

⁴ La méthode de dégradation des pondérations

Chapitre II

Machines à Vecteurs de

Support

2.1. Introduction

L'origine des machines à vecteurs de support (SVM) remonte à 1975 lorsque Vapnik et Chervonenkis ont proposé le principe du risque structurel et la dimension VC pour caractériser la capacité d'une machine d'apprentissage. A cette époque, ce principe n'a pas trouvé place et il n'existait pas encore un modèle de classification solidement appréhendé pour être utilisable. Il a fallu attendre jusqu'à l'an 1982 pour que Vapnik propose le SVM un premier classifieur basé sur la minimisation du risque structurel.

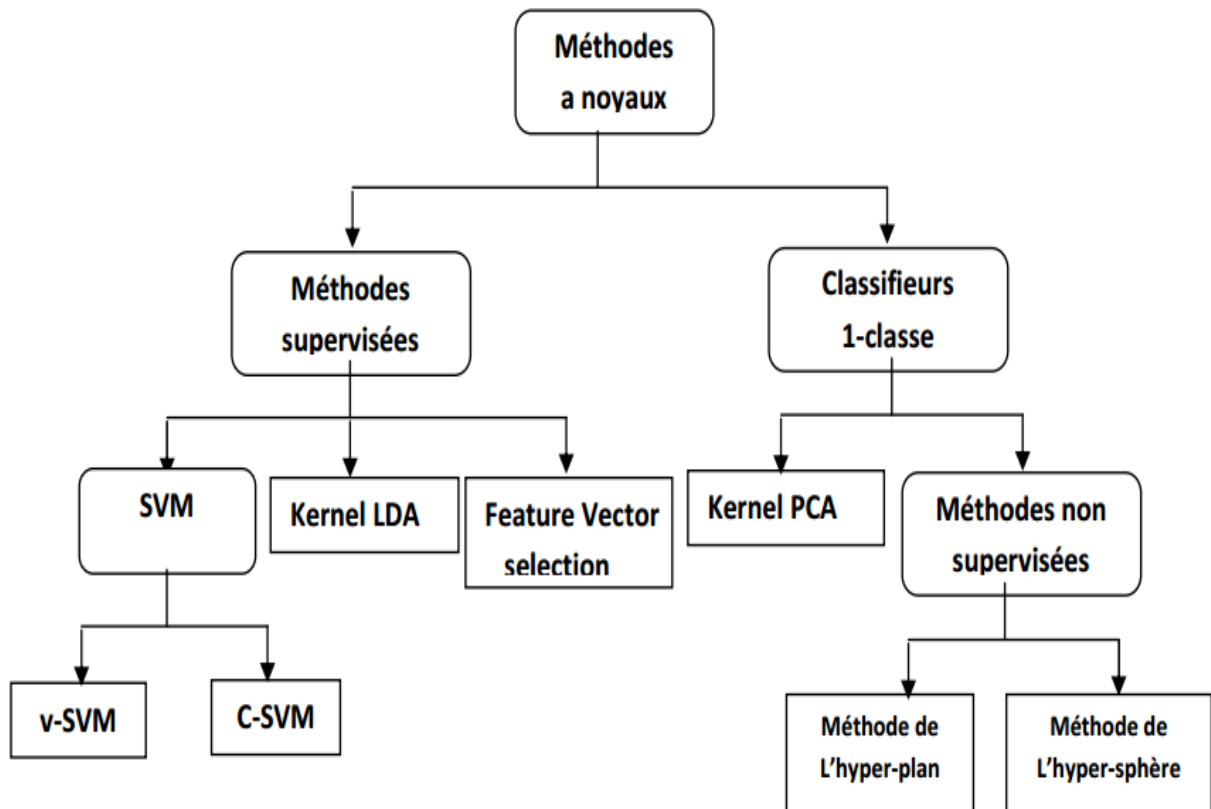


Figure2.1 : Arbre de classification des méthodes d'apprentissage à base de noyaux.

Le classifieur SVMs a été conçu pour une séparation de deux ensembles de données. Il est considéré donc comme un classifieur binaire. Le but de SVM est de trouver un hyperplan qui va séparer et maximiser la marge de séparation entre deux classes.

Le problème de recherche de l'hyperplan séparateur possède une formulation duale. Ceci est particulièrement intéressant car, sous cette formulation duale, le problème peut être résolu au moyen de méthode d'optimisation quadratique. Depuis, les SVMs (le pluriel est utilisé pour désigner les différentes variantes du SVM) n'ont cessé de susciter l'intérêt de plusieurs communautés de chercheurs de différents domaines d'expertise. Par exemple, Cortes et al.

Dans [19], et Burges et al [20]. ont appliqué les SVM à la reconnaissance des chiffres manuscrits isolés, Blanz et al. [21] ont expérimenté le SVM sur des objets 2D de vues différentes. Osuna et al. ont traité la détection d'images de visages [22]. Dans la plupart [23] ainsi que Wahba [24] ont mis en évidence la ressemblance entre le SVM et la théorie de régularisation. Ils ont démontré qu'associer un noyau particulier à un SVM revient à considérer une pénalisation différente de l'erreur d'apprentissage en maximisant la marge.

Ce qui permet de dire que la maximisation de la marge dans l'espace augmenté est une régularisation de l'apprentissage. Dès lors, le SVM permet de répondre à deux problèmes centraux de la théorie de l'apprentissage statistique :

- Le contrôle de la capacité du classifieur ;
- Le sur-apprentissage.

2.2. Formulation

On considère la formulation suivante :

Soit l'ensemble D tel que

$$D_l = \{(x_i, y_i) \in X \times Y \text{ ou } i = 1, \dots, l \text{ et } Y = \{-1, +1\}\} \quad (2.1)$$

Soit un problème de classification binaire défini par les observations $(x_1, y_1) \dots (x_l, y_l)$ tirées de manières indépendantes d'une même distribution. Chacune des données $x_i \forall i=1 \dots l$ représente un vecteur de caractéristique dans \mathbb{R}^n . les variables $y_i = \{+1, -1\} \forall i=1 \dots l$ représente les classes d'appartenance correspondantes aux données x_i .

Il s'agit d'estimer la fonction de décision $f(x)$ qui approxime au mieux les échantillons de l'ensemble d'apprentissage, telle que $y_i = f(x_i)$ pour toute donnée x_i

2.3. Pourquoi les Machines à Vecteurs de Support (SVM) ?

L'algorithme des machines à vecteurs de support a été développé dans les années 90 par le russe Vladimir Vapnik. Initialement, les SVM ont été développés comme un algorithme de classification binaire supervisée. Il s'avère particulièrement efficace de par le fait qu'il peut traiter des problèmes mettant en jeu de grands nombres de descripteurs, qu'il assure une solution unique (pas de problèmes de minimum local comme pour les réseaux de neurones) et il a fourni de bons résultats sur des problèmes réels [18].

L'algorithme sous sa forme initiale revient à chercher une frontière de décision linéaire entre deux classes, mais ce modèle peut considérablement être enrichi en se projetant dans un autre

espace permettant d'augmenter la séparabilité des données. On peut alors appliquer le même algorithme dans ce nouvel espace, ce qui se traduit par une frontière de décision non linéaire dans l'espace initial.

2.4. Apprentissage statistique et SVM

La notion d'apprentissage étant importante, nous allons commencer par effectuer un rappel. L'apprentissage par induction permet d'arriver à des conclusions par l'examen d'exemples particuliers. Il se divise en apprentissage supervisé et non supervisé. Le cas qui concerne les SVM est l'apprentissage supervisé.

Les exemples particuliers sont représentés par un ensemble de couples d'entrée/sortie. Le but est d'apprendre une fonction qui correspond aux exemples vus et qui prédit les sorties pour les entrées qui n'ont pas encore été vues. Les entrées peuvent être des descriptions d'objets et les sorties la classe des objets donnés en entrée [6].

2.5. SVM principe de fonctionnement général

2.5.1. Notions de base: Hyperplan, marge et support vecteur

Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan. Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points [25].

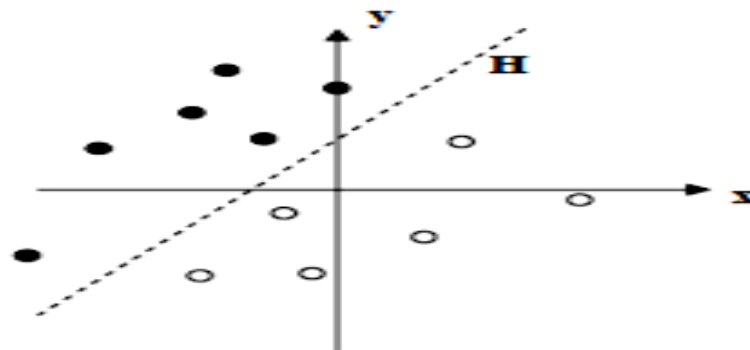


Figure 2.2 : Exemple d'un hyperplan séparateur.

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

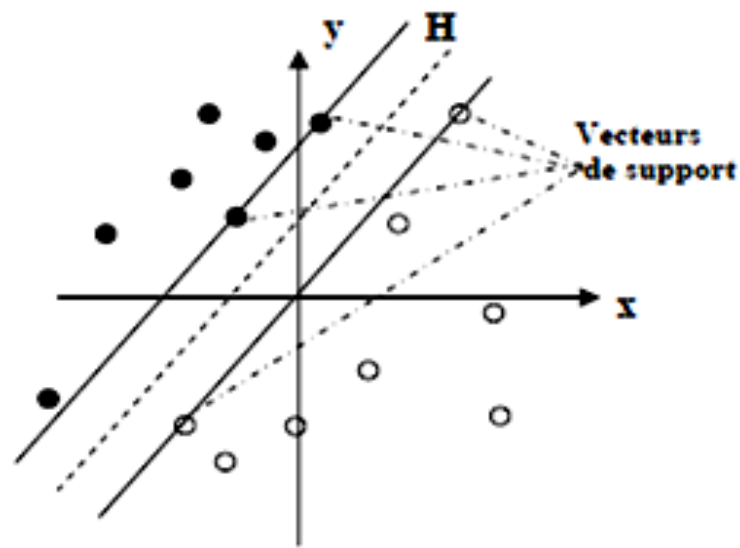


Figure 2.3 : Exemple de vecteurs de support.

On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge [25].

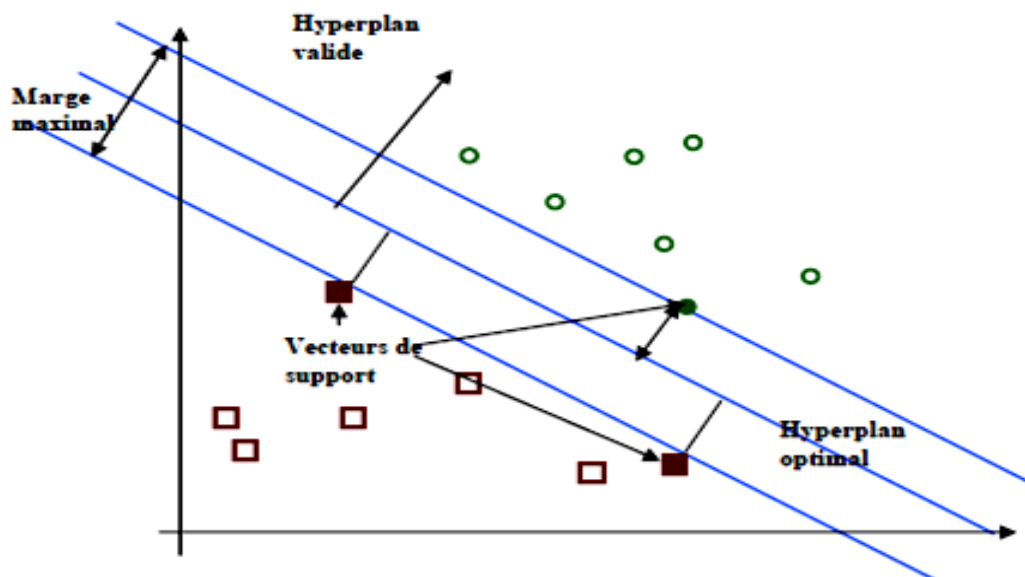


Figure 2.4 : Exemple de marge maximale (hyperplan valide).

2.5.2. Linéarité et non-linéarité

Parmi les modèles des SVM, on constate les cas linéairement séparable et les cas non linéairement séparable. Les premiers sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de

séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables [25].

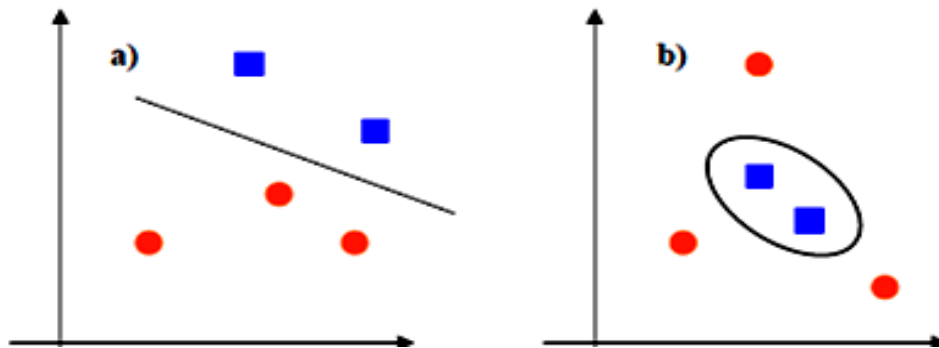


Figure 2.5: a) Cas linéairement séparable, b) Cas non linéairement séparable.

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

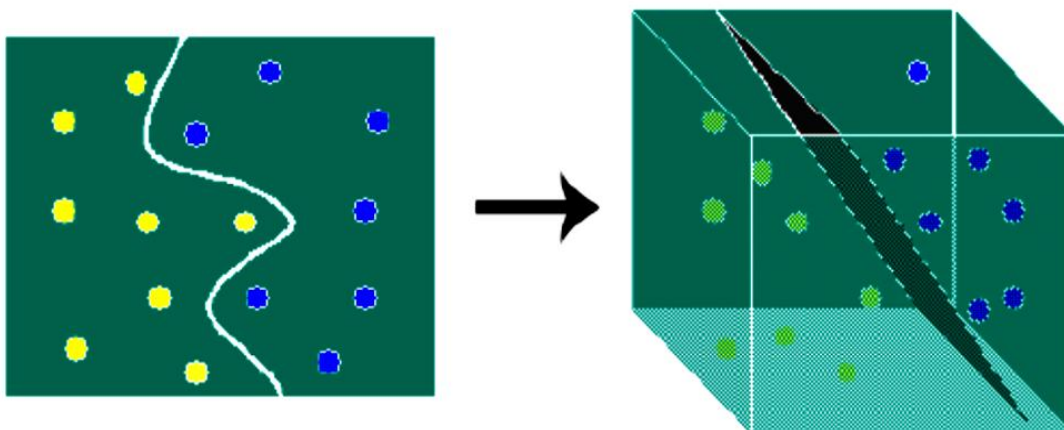


Figure 2.6: Exemple de changement de l'espace de données.

2.5.3. Illustration de transformation de cas non linéaire : le cas XOR

Le cas de XOR n'est pas linéairement séparable, si on place les points dans un plan à deux dimensions [2], on obtient la figure suivante :

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

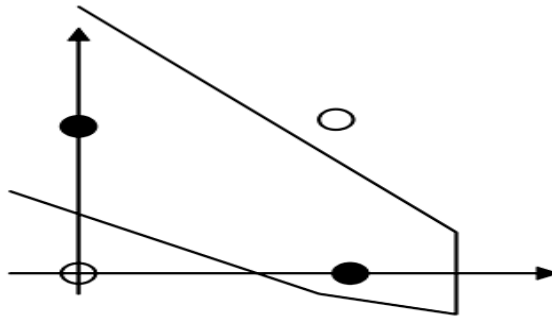


Figure 2.7: Illustration de cas non linéairement séparable (le cas XOR).

2.6. Fondements mathématiques

2.6.1. Problème d'apprentissage

On s'intéresse à un phénomène f (éventuellement non déterministe) qui, à partir d'un certain jeu d'entrées x , produit une sortie $y = f(x)$. Le but est de retrouver cette fonction f à partir de la seule observation d'un certain nombre de couples entrée-sortie $\{(x_i; y_i) : i = 1, \dots, n\}$ afin de « prédire » d'autres évènements. On considère un couple (X, Y) de variables aléatoires à valeurs dans X, Y . Seul le cas $Y = \{-1, 1\}$ (classification) nous intéresse ici (on peut facilement étendre au cas $\text{card}(Y) = m > 2$ et au cas $Y = \mathcal{R}$). La distribution jointe de (X, Y) est inconnue. Sachant qu'on observe un échantillon $S = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ de n copies indépendantes de (X, Y) , on veut: construire une fonction $h : X \rightarrow Y$ telle que $P(h(X) \neq Y)$ soit minimale [34].

2.6.2. Classification à valeurs réelles

Plutôt que de construire directement $h : X \rightarrow \{-1, 1\}$, on construit : $f : X \rightarrow \mathcal{R}$ (ensemble des réels). La classe est donnée par le signe de f ; $h = \text{signe}(f)$. L'erreur se calcule avec $P(h(X) \neq Y) = P(Y f(X) \leq 0)$. Ceci donne une certaine idée de la confiance dans la classification. Idéalement, $|Y f(X)|$ est proportionnel à $P(Y|X)$. $Y f(X)$ représente la marge de f en (X, Y) . Le but à atteindre est la construction de f et donc h . Nous allons voir comment y parvenir [7].

2.6.3. Complexité

Nous allons évaluer la complexité (temps de calcul) de l'algorithme SVM. Elle ne dépend que du nombre des entrées à classer (d) et du nombre de données d'apprentissage (N). On montre que cette complexité est **polynomiale** en N . $dN^2 \leq \text{Complexité} \leq dN^3$

Taille de la matrice hessienne = N^2

En effet, on doit au moins parcourir tous les éléments de la matrice ainsi que toutes les entrées. Pour un très grand nombre de données d'apprentissage, le temps de calcul explose. C'est Pourquoi les SVMs sont pratiques pour des « petits » problèmes de classification [26].

2.6.4. La marge

Supposons pour l'instant que les échantillons de l'ensemble d'apprentissage sont séparables par un hyperplan (Figure 2.2), i.e. on choisit des fonctions de décision de la forme :

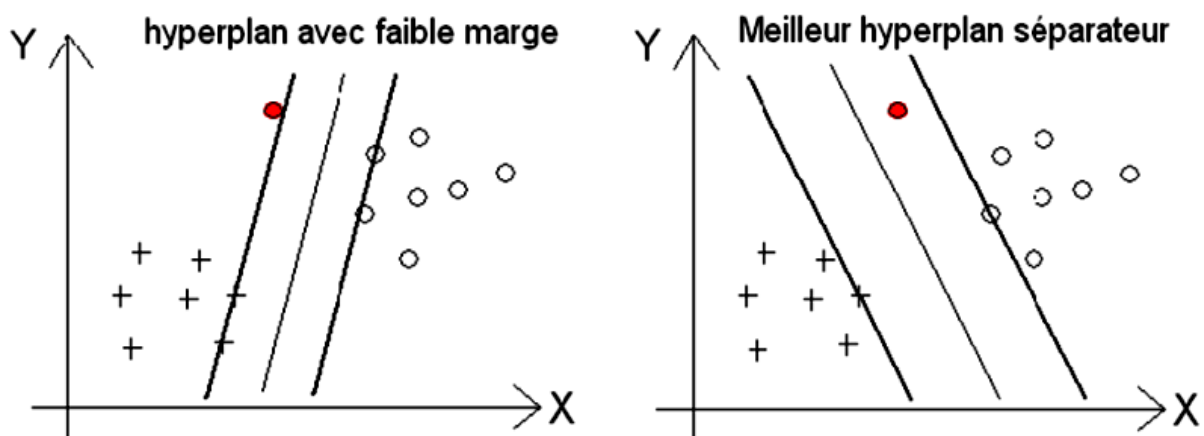
$$f(x) = w \cdot x^* + b \quad (2.2)$$

La marge est la distance minimale entre les échantillons de l'ensemble d'apprentissage et la frontière de décision

2.6.4.1. Pourquoi maximiser la marge ?

Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsque l'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples.

Dans le schéma qui suit, la partie droite nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé.



2.6.4.2. Maximisation de la marge

Dans le cas séparable, on va considérer les points les plus près de l'hyperplan séparateur : vecteurs supports (support vectors). Pour tout point de l'espace des exemples, la distance a

l'hyperplan séparateur est donnée par : $r = \frac{|w \cdot x + b|}{\|w\|} \quad (2.3)$

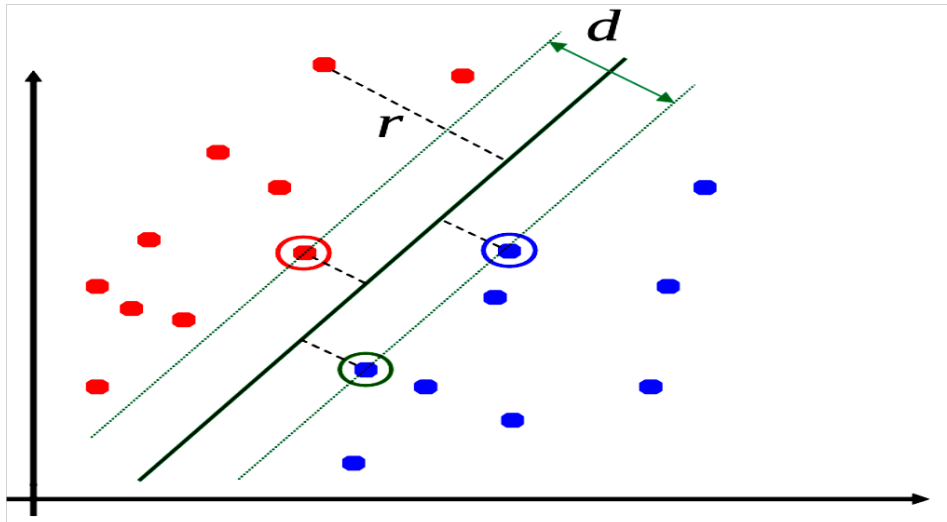


Figure 2.8: marge de la distance entre les 2 classes.

2.6.4.3. Cas séparable

Problème primal

Un point (x, y) est bien classé si et seulement si $yf(x) > 0$. Comme le couple (w, b) est défini à un coefficient multiplicatif près, on s'impose $yf(x) \geq 1$.

Le problème revient alors à trouver w et b tels que $d = \frac{2}{\|w\|}$ est maximale (x_i, y_i) sous les contraintes :

$$\begin{cases} w \cdot x + b \geq 1, & \text{si } y_i = 1 \\ w \cdot x + b < 1, & \text{si } y_i = -1 \end{cases} \quad (2.4)$$

De manière équivalente, le problème peut s'écrire plus simplement comme la minimisation de :

$$\frac{1}{2} \|w\|^2 \quad (2.5)$$

$$\text{Sous les contraintes : } y_i (w \cdot x + b) \geq 1, \forall i \in [1, N] \quad (2.6)$$

Il peut être en effet plus aisé de minimiser $\|w\|^2$ plutôt que directement $\|w\|$.

Les points qui vérifient $y_i (w \cdot x_i + b) = 1$, sont appelés "vecteurs supports". Ce sont les points les plus près de la marge. Ils sont sensés être peu nombreux par rapport à l'ensemble des exemples.

Le problème dual

On passe du problème primal au problème dual en introduisant des multiplicateurs de Lagrange (α_i) pour chaque contrainte

$$\begin{cases} \max \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ \forall i, 0 \leq \alpha_i \leq c \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (2.7)$$

c : est une constante.

C'est un problème de programmation quadratique de dimension n (nombre d'exemples). On définit ainsi la matrice suivante appelée « matrice hessienne » : $(x_i \cdot x_j)_{i,j}$ qui représente la matrice des produits des entrées X (La notation matricielle permettant de résoudre plus facilement le problème en informatique).

On montre que si les α_i^* sont solutions de ce problème alors on a :

$$w^* = \sum_{i=1}^N \alpha_i^* y_i x_i \quad (2.8)$$

Seuls les α_i correspondant aux points les plus proches sont non nuls. On parle de vecteurs de support.

La fonction de décision associée est donc :

$$f(x) = \sum_{i=1}^N \alpha_i^* y_i x_i \cdot x + b \quad (2.9)$$

Il existe néanmoins des cas où on ne peut pas classer les entrées de façon linéaire.

Un problème s'exprime sous forme duale comme la minimisation de :

$$w(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (2.10)$$

Fait partie des problèmes d'optimisation quadratique pour lesquels il existe de nombreux algorithmes de résolution à savoir: SMO¹, simpleSVM, LASVM² ...

¹ SMO : « Optimisation Séquentielle par modification Minimale » (SMO) développé par John Platt.

² LASVM : est un résolveur SVM approximative qui utilise approximation en ligne proposé par Bordes et Bottou, 2005.

2.6.4.4. Cas non séparable

Classification à marge souple

L'idée est d'ajouter des variables d'ajustement ξ_i dans la formulation pour prendre en compte les erreurs de classification ou le bruit

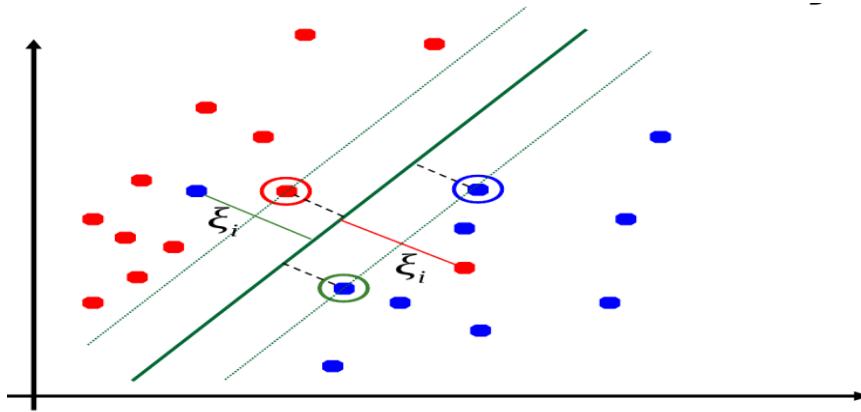


Figure 2.9: l'erreur en cas des données séparables.

Formulation:

$$\text{Minimiser } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i ; \quad (2.11)$$

C est une constante permettant de contrôler le compromis entre nombre d'erreurs de classement, et la largeur de la marge.

Que se passe-t-il si l'ensemble d'apprentissage est intrinsèquement non séparable ?



Figure 2.10: données intrinsèquement non séparable.

Pourquoi ne pas plonger le problème dans un espace de plus grande dimensionnalité ?

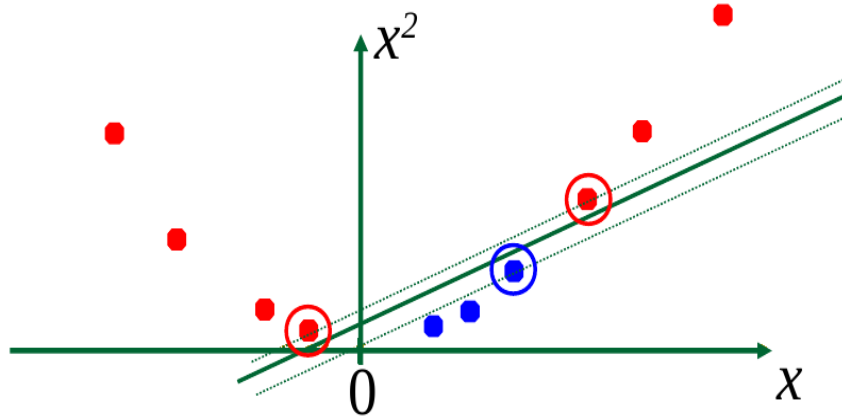
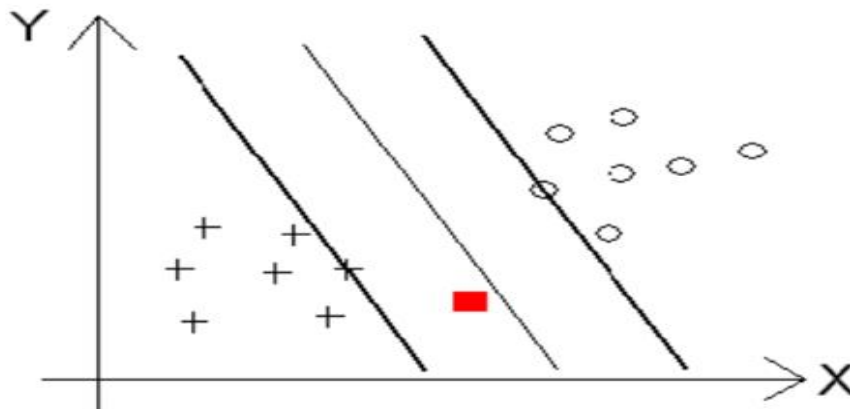


Figure 2.11: données non séparable au plan de deux démentions.

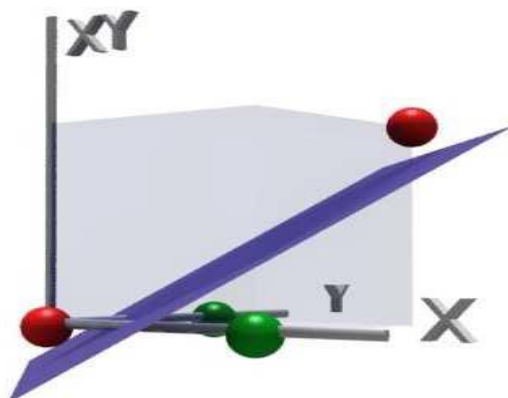
En général, la classification d'un nouvel exemple inconnu est donnée par sa position par rapport à l'hyperplan optimal. Dans le schéma suivant, le nouvel élément sera classé dans la catégorie des « + ».



2.7. Les noyaux

2.7.1. Idée générale

L'espace des données peut toujours être plongé dans un espace de plus grande dimension dans lequel les données peuvent être séparées linéairement [9].



Exemple le cas XOR : On effectue la transformation :

$$(x, y) \rightarrow (x, y, x \cdot y)$$

Dans ce cas le problème peut être séparé linéairement.

La résolution des SVM ne s'appuie que sur le produit scalaire $\langle x_i, x_j \rangle$ entre les vecteurs d'entrée. Si les données d'apprentissage sont plongées dans un espace de plus grande dimension via la transformation $\varphi: x \rightarrow \varphi(x)$, le produit scalaire devient :

$$K(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle \quad (2.12)$$

$K(x_i, x_j)$ est appelée fonction noyau.

2.7.2. Formulation (cas non linéaire)

- Minimiser: $w(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j \gamma_i \gamma_j (x_i \cdot x_j)$ (2.13)

- Sous les contraintes $0 \leq \alpha_i \leq C$

- La solution est de la forme:

$$w = \sum_{i \in sv} \alpha_i \gamma_i \varphi(x_i) \quad (2.14)$$

Avec sv est les supports des vecteurs

$$f(x) = w \cdot \varphi(x) + b = \sum_{i \in sv} \alpha_i \gamma_i K(x_i, x) + b \quad (2.15)$$

2.7.3. Les fonctions noyaux

Il est en général beaucoup plus naturel de définir une fonction noyau $K(x_i, x_j)$ qui vérifie la condition $K(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$ plutôt que de créer une fonction Φ de projection qui permette d'obtenir un espace H suffisamment "flexible" et ensuite de rechercher si une formulation simple de K existe pour que la relation $\langle \Phi(x_i), \Phi(x_j) \rangle \equiv K(x_i, x_j)$ soit vérifiée. De plus, des phénomènes de sur-apprentissage peuvent apparaître si la marge correspondante est faible pour les données prises dans cet espace de redescription préconstruit, s'il est trop bien adapté aux données d'entraînement.

On ne peut pas pour autant définir une fonction noyau K sans prendre certaines précautions. En particulier, il est indispensable que l'espace de redescription existe et donc que la fonction Φ existe, même si elle n'est pas facilement exprimable. Dans le cas contraire, le problème de recherche d'un hyperplan optimal n'a pas lieu d'exister.

2.7.3.1. Propriétés des fonctions noyaux

Une fonction noyau K est une fonction symétrique,

$$\forall x_1, x_2 \in X : K(x_1, x_2) = K(x_2, x_1)$$

qui vérifie l'inégalité de Cauchy-Schwarz,

$$\forall x_1, x_2 \in X : K(x_1, x_2)^2 \leq K(x_1, x_1)K(x_2, x_2)$$

Cependant ces conditions ne sont pas suffisantes pour garantir l'existence d'un espace de redescription dans lequel K correspond à un produit scalaire.

Une des conséquences de ce théorème est que toute matrice de Gram³

$$K = (K(x_i, x_j))_{i,j}^N = 1 \quad (2.16)$$

est définie semi-positive (elle n'a pas de valeur propre négative).

N : est le nombre total d'exemples.

2.7.3.2. L'astuce des fonctions noyaux

C'est la possibilité de transformer une fonction noyau dans un problème en une autre fonction noyau tout en gardant les propriétés de la méthode les utilisant. La formulation duale des SVM utilise cette possibilité [9].

2.7.3.3. Relation entre les distances

Il est possible d'utiliser l'astuce des fonctions noyaux pour les méthodes basés sur la distance euclidienne en utilisant le fait que:

$$d(x_i, x_j)^2 = \langle x_i - x_j, x_i - x_j \rangle = \langle x_i, x_i \rangle + \langle x_j, x_j \rangle - 2 \langle x_i, x_j \rangle \quad (2.17)$$

En particulier la distance entre deux points dans l'espace de redescription est facile à déterminer:

$$d(\Phi(x_i), \Phi(x_j))^2 = K(x_i, x_i) + K(x_j, x_j) - 2K(x_i, x_j) \quad (2.18)$$

Cette possibilité est par exemple utilisée dans le cadre de l'application de la règle du plus proche voisin dans un espace de redescription donné.

2.7.3.4. Quelques fonctions noyaux

Une liste non exhaustive de quelques fonctions noyaux couramment utilisées avec les SVM:

- **Noyau linéaire** $K(x_i, x_j) = x_i \cdot x_j$; (2.19)

- **Noyau polynomial de degré p** $K(x_i, x_j) = 1 + x_i \cdot x_j^p$; (2.20)

- **Noyau Gaussien** $K(x_i, x_j) = \exp \frac{-\|x_i - x_j\|^2}{2\delta^2}$; (2.21)

³Matrice de Gram : Soit $K(s; t)$ un noyau positif sur Ω et $(x_i)_{i=1;N}$ une suite de points de Ω . On appelle matrice de Gram la matrice carrée K de dimension N et de terme général $K_{ij} = K(x_i; x_j)$.

- Cette formulation est équivalente aux réseaux de neurones à bases radiales avec l'avantage supplémentaire que les centres des fonctions à base radiale (qui sont les vecteurs supports) sont optimisés
- **Sigmoïde** $K(x_i, x_j) = \tanh(\alpha x_i \cdot x_j + \beta)$; (2.22)

Les paramètres P , β , α et δ sont les paramètres libres de ces différentes fonctions noyaux et ils ont une grande influence sur les capacités en généralisation des SVM.

Il est aussi possible de définir de nouvelles fonctions noyaux en combinant des fonctions noyaux de bases. La somme ou le produit de deux fonctions noyaux valides est également une fonction noyau valide.

Les noyaux gaussiens sont des noyaux dits de type radial (fonction à base radiale abrégée RBF indiquant qu'ils dépendent de la distance entre deux exemples).

2.8. Architecture générale d'une machine à vecteur support

Une machine à vecteur support, recherche à l'aide d'une méthode d'optimisation, dans un ensemble d'exemples d'entraînement, des exemples, appelés vecteurs support, qui caractérisent la fonction de séparation. La machine calcule également des multiplicateurs associés à ces vecteurs.

Les vecteurs supports et leurs multiplicateurs sont utilisés pour calculer la fonction de décision pour un nouvel exemple [7]. Le schéma (figure 2.12) résume l'architecture générale d'une SVM dans le cas de la reconnaissance des caractères manuscrits.

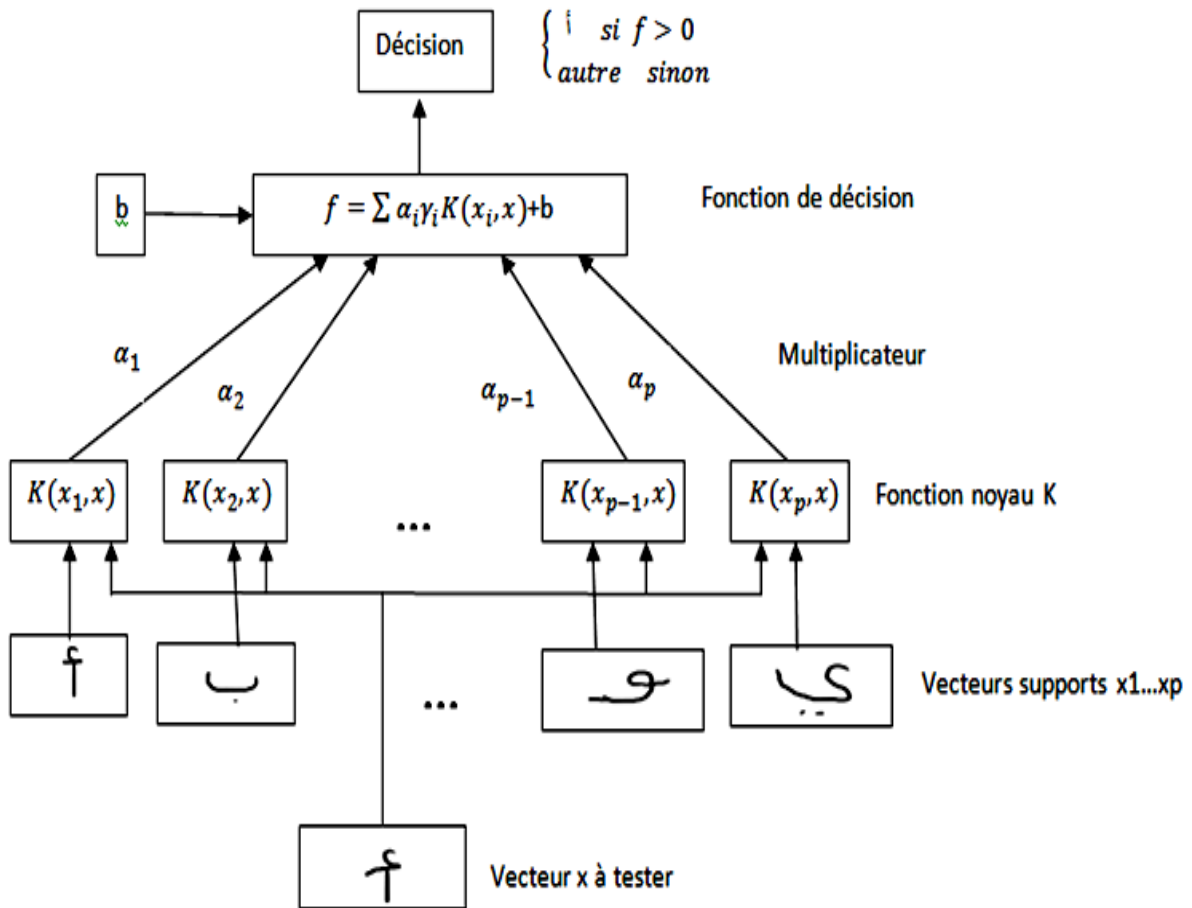


Figure 2.12. Architecture d'une machine à vecteur de support.

La fonction noyau K est utilisée pour calculer la distance entre le vecteur à tester x et chaque vecteur support dans l'espace de caractéristique. Les résultats sont ensuite linéairement combinés en utilisant les multiplicateurs de Lagrange α_i et ajoutés au biais b . Le résultat final f permet de décider à propos du nouveau vecteur : si $f(x)$ est positive, il s'agit du caractère "f", sinon, il s'agit d'un autre caractère.

2.8.1. SVMs multiclasse

Les méthodes des machines à vecteur support multiclasse, réduisent le problème multiclasse à une composition de plusieurs hyperplans biclasses permettant de tracer les frontières de décision entre les différentes classes [27]. Ces méthodes décomposent l'ensemble d'exemples en plusieurs sous ensembles représentant chacun un problème de classification binaire. Pour chaque problème un hyperplan de séparation est déterminé par la méthode SVM binaire.

On trouve dans la littérature plusieurs méthodes de décomposition :

2.8.1.1. Une-contre-reste (1vsR)

Elle consiste à déterminer pour chaque classe k un hyperplan $H_k(w_k, b_k)$ la séparant de toutes les autres classes. Cette classe k est considéré comme étant la classe positive (+1) et les autres classes comme étant la classe négative (-1).

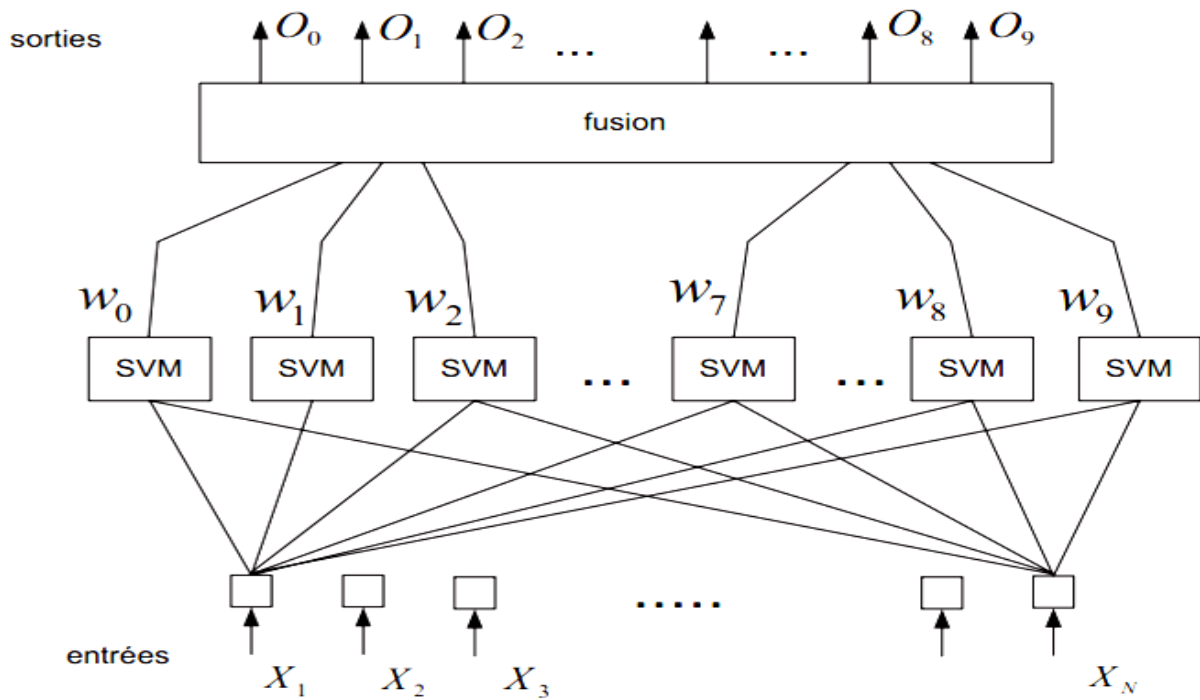


Figure 2.13: Architecture du système en stratégie Un-contre-Tous

2.8.1.2. Une-contre-une (1vs1)

Elle consiste à utiliser un classifieur pour chaque paire de classes. Au lieu d'apprendre K fonctions de décisions, la méthode 1vs1 discrimine chaque classe de chaque autre classe, ainsi $K(K - 1)/2$ fonctions de décisions sont apprises.

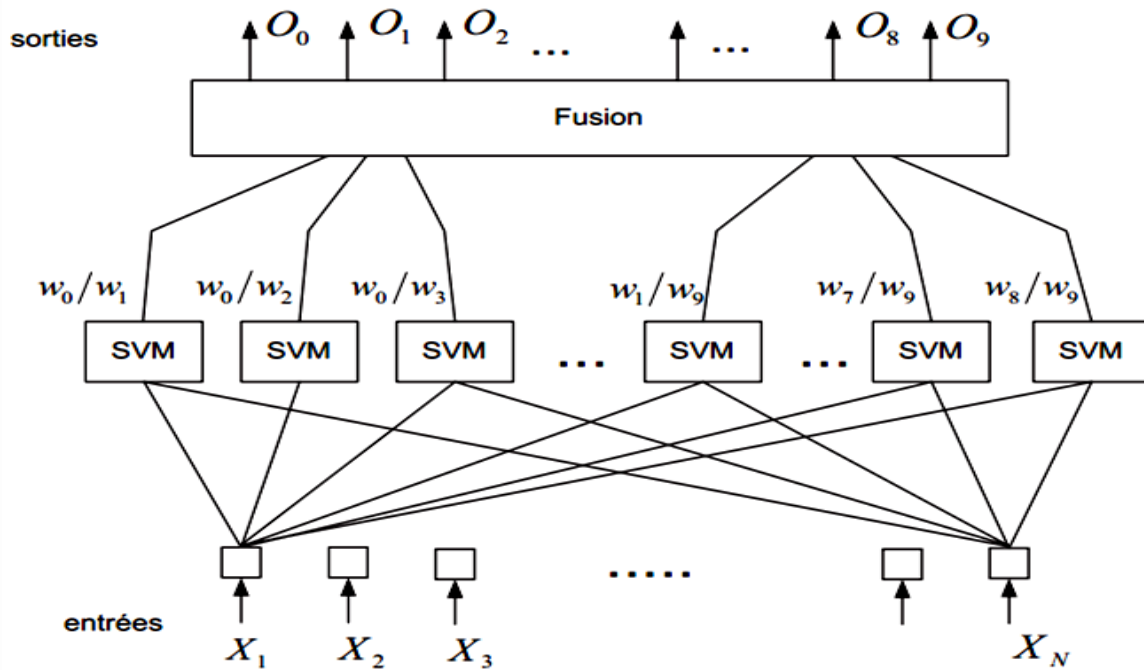


Figure 2.14 : Architecture du système en stratégie Une-contre-Une

2.8.2. SVM monoclasse (Novelty detections)

Pour la classification SVM monoclasse, il est supposé que seules les données de la classe cible sont disponibles. L'objectif est de trouver une frontière qui sépare les exemples de la classe cible du reste de l'espace, autrement dit, une frontière autour de la classe cible qui accepte autant d'exemples cibles que possible [28]. Cette frontière est représentée par une fonction de décision positive à l'intérieur de la classe et négative en dehors. La figure 2.15 représente, en deux dimensions, un cas de séparation d'une classe de toute autre classe.

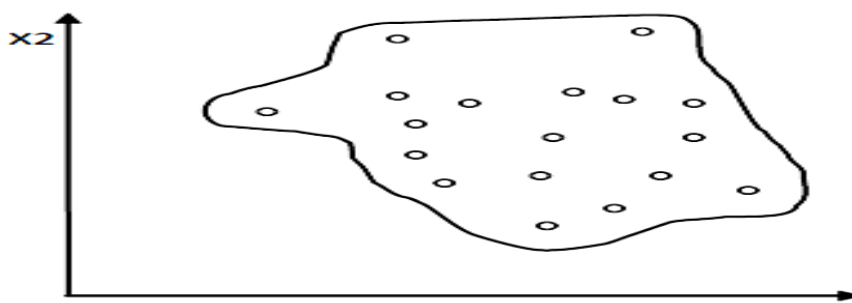


Figure 2.15. Séparation des exemples d'une classe du reste de l'espace

2.9. Les domaines d'applications

La réalisation d'un programme d'apprentissage par SVM se ramène à résoudre un problème d'optimisation impliquant un système de résolution dans un espace de dimension conséquente. L'utilisation de ces programmes revient surtout à sélectionner une bonne famille de fonctions

noyau et à régler les paramètres de ces fonctions. Ces choix sont le plus souvent faits par une technique de validation croisée, dans laquelle on estime la performance du système en la mesurant sur des exemples n'ayant pas été utilisés en cours d'apprentissage.

2.9.1. Exemple d'application :

- Classification de données biologiques/physiques
- Classification de documents numériques
- Classification d'expressions faciales
- Classification de textures
- E-learning
- Détection d'intrusion
- Reconnaissance de la parole
- CBIR : Content Based Image Retrieval

2.10. Systèmes de reconnaissance d'écriture à base de SVM

2.10.1. Système de Ayat, 2004

Le travail présenté dans [29] présente une nouvelle méthodologie de sélection de modèle automatique des machines à vecteurs de support pour la reconnaissance de chiffres manuscrits. L'approche permet d'optimiser les paramètres de noyaux et de réduire efficacement la complexité du classifieur en minimisant le nombre de vecteurs de support.

Ceci s'accompagne d'une réduction de l'erreur de généralisation. L'algorithme proposé a été testé sur la base USPS (US Postal Service), et la base INDCENPARMI. La base de données USPS est un benchmark bien connu au sein de la communauté de reconnaissance de formes. Cet ensemble contient 9298 images de chiffres manuscrits dont 7291 images de test.

Ces images ont été saisies à partir d'images d'enveloppes collectées au centre CEDAR à Buffalo (Etats-Unis). Chaque image de chiffre est représentée par 16*16 pixels de niveau de gris allant de 0 à 255. La base de données de chiffres indiens de CENPARMI a été collectée à partir d'images de chèques saoudiens fournis par la compagnie "Al Rajhi Banking Corporation". La base de chiffres indiens a été constituée à partir des montants numériques et contient deux corpus de données. La taille du corpus d'apprentissage est 4682, celle de l'ensemble de test est 1939.

Pour décrire les images de chiffres indiens, Ayat a considéré un premier ensemble de caractéristiques structurelles calculé par rapport aux régions pouvant exister dans les images

de chiffres. Les régions détectées sont : région en vallée, région en montagne, région en boucle. Ces primitives structurelles sont combinées avec des primitives statistiques. Les primitives statistiques considérées sont basées sur la technique de zonage et la distribution des directions de Freeman et les courbures des pixels composant les contours du tracé. Le taux de reconnaissance obtenu est 85%. Le taux de reconnaissance obtenu sur la base USPS après l'entraînement du système avec les valeurs des hyper-paramètres est 95,8%, 59,6% et 94,7% respectivement avec les noyaux KMOD (noyau proposé par Ayat), RBF et polynomial (d=2). Le taux de reconnaissance est amélioré 0,2% par rapport à la sélection manuelle.

2.10.2. Système de Bellili, 2001

Dans [30], les auteurs proposent un système de reconnaissance de chiffres manuscrits par combinaison hybride de réseau de neurones de type MLP et de machines à vecteurs de support. Les SVM sont utilisés pour améliorer significativement les performances de reconnaissance d'un réseau MLP dans le voisinage des hyperplans de séparation entre chaque paire de classes numériques, dans l'espace augmenté des formes à reconnaître.

Cette architecture de combinaison est fondée sur la constatation que les deux sorties maximales de la couche de sortie du MLP contiennent presque systématiquement la bonne classe de la forme à classifier et que certaines paires de classes constituent la majorité des confusions générées par le MLP (le 3 et 9 par exemple). Des SVM locaux spécialisés sont introduit pour détecter la bonne classe parmi les deux meilleures hypothèses de classification fournies par le réseau MLP. Le système hybride MLP-SVM réalise un taux de reconnaissance de 98,01% sur des chiffres manuscrits issus de codes postaux.

2.10.3. Système de Bahlman, 2002

Le travail présenté dans [31] décrit une nouvelle approche de classification pour la reconnaissance de l'écriture manuscrite "en ligne" des caractères latins. L'approche proposée combine les DTW (*Dynamic Time Warping*) et les SVM pour établir une nouvelle fonction noyau appelée : "*Gaussian Dynamic Warping Kernel*". Le système proposé a été testé sur la base "UNIPEN" de caractères manuscrits latins en utilisant la méthode de minimisation séquentielle (SMO). D'après les auteurs, cette approche a donné des résultats meilleurs que ceux obtenus par les HMM⁴.

⁴ Hidden Marcov Model

2.11. Avantages et désavantages des SVM

L'avantage de la création d'une fonction de décision avec l'algorithme des SVM est que la solution produite correspond à l'optimum d'une fonction convexe. Elle ne possède donc pas plusieurs optima locaux comme pour les réseaux de neurones (dans leur formulation classique), mais un optimum global. Cet optimum correspond à la minimisation du risque structurel et donc à la recherche d'une hypothèse possédant de bonnes capacités de généralisation à partir d'un espace d'hypothèses donné. L'espace d'hypothèses dépend du choix de la fonction noyau, ce qui est à la fois un avantage et un inconvénient [33].

Un avantage car l'algorithme d'apprentissage reste identique. Le problème est que la fonction noyau a souvent des paramètres libres, comme la largeur de bande avec un noyau gaussien, et que la recherche de la valeur optimale de ces paramètres libres ne correspond plus à la recherche du minimum d'une fonction convexe [33].

La recherche des valeurs idéales de ces paramètres, appelée communément sélection de modèles, est critique. Illustrons cet état critique avec l'utilisation d'un noyau gaussien et une largeur de bande σ très faible. La fonction de décision produite, pour ces conditions, sera cohérente avec les données et elle aura autant de vecteurs de support que d'exemples. L'hyperplan sélectionné, bien que de marge géométrique maximum, a une marge faible.

Les bornes supérieures vues précédemment sont très élevées et elles ne garantissent plus rien sur le pouvoir de généralisation. Donc la sélection d'une mauvaise fonction noyau (ou de ces paramètres) peut conduire à produire un effet de sur-apprentissage et c'est le principal désavantage des SVM [33].

Un autre désavantage des SVM est la durée importante de la phase d'entraînement, en particulier si une sélection de modèles doit être réalisée, bien que dans les dix dernières années plusieurs optimisations de l'algorithme initial aient été proposées.

Un dernier désavantage est la complexité de la fonction de décision produite lorsque la base d'apprentissage est de taille importante, même si, à la différence de la règle PPV, l'ensemble des exemples n'est pas utilisé grâce à la notion de vecteur de support.

2.12. Conclusion

Dans ce chapitre, on a tenté de présenter de manière simple et complète le concept de système d'apprentissage introduit par Vladimir Vapnik, les Machines à Vecteurs de Support. On a donné une vision générale et une vision purement mathématiques des SVM. Cette méthode de classification est basée sur la recherche d'un hyperplan qui permet de séparer au mieux des ensembles de données. La formulation élégante des SVM laisse très peu de place aux paramètres et leur nombre est linéaire en fonction de l'ensemble d'apprentissage.

Le SVM est donc une méthode de classification particulièrement bien adaptée pour traiter des données de très haute dimension telles que les textes et les images. Dans le chapitre suivant, on définit les paramètres fixés par l'utilisateur pour l'algorithme d'apprentissage des SVM.

Chapitre III

Sélection de modèles

de machines à vecteurs de

support

3.1. Introduction

Plusieurs recherches se sont intéressées très tôt à la problématique de la sélection de modèle pour les machines à vecteurs de support et aussi l'optimisation de ces modèles. Plus récemment, la sélection de modèle a été vue comme une tâche d'optimisation. Dans ce contexte, un algorithme d'optimisation est mis en œuvre afin de trouver l'ensemble d'hyper-paramètres qui permettra d'obtenir les meilleures performances en classification.

Les méthodes permettant de réaliser l'optimisation seront très variables suivant la nature du problème. Il existe des méthodes classiques et d'autres heuristiques. Dans le cas plus général où seule la fonction noyau a été fixée (un noyau gaussien par exemple), mais pas les paramètres du noyau (σ pour un noyau gaussien) et la constante de régularisation C , le choix des valeurs optimales de σ et C correspond à la recherche d'un modèle θ optimal ($\theta \equiv (C, \sigma)$) [42].

Le choix de α^* (multiplicateurs de lagrange) étant considéré comme implicite à un modèle μ particulier, même si le calcul effectif de α^* n'est pas trivial. Le problème de la recherche du modèle μ correspond à un problème qui n'est plus convexe, il y a donc en général plusieurs minima locaux. Parmi les méthodes proposées pour résoudre ce type de problème, des méthodes désignées sous le terme de méta-heuristiques. Dans ce chapitre en va voir quelque détaille sur ces méthode de sélection et optimisation de modèle SVM.

3.2. Les classifieurs SVM et leurs hyper paramètres

Les problèmes de classification avec des coûts de mauvaise classification asymétriques et inconnus peuvent être pris en charge par les SVM en introduisant le paramètre de régularisation C . Dans ce cas, étant donné un ensemble de N exemples d'apprentissage $x \in R^n$ appartenant à la classe labellisée y_i :

$$(x_1, y_1) \dots (x_N, y_N), x \in R^n, y_i \in \{-1, +1\} \quad (3.1)$$

α ensemble des multiplicateurs de lagrange, paramètres du SVM .la maximisation du lagrangien dual par rapport aux α_i devient :

$$Max_{\alpha} \left\{ \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j k(x_i, x_j) \right\} \quad (3.2)$$

$$\text{Sous les contraintes } \left\{ \begin{array}{l} \max \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j x_i x_j \\ \forall i, 0 \leq \alpha_i \leq c \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{array} \right.$$

3.3. Méthodes de sélection de modèle SVM

3.3.1. Technique de la validation croisée

La validation croisée est une bonne technique d'évaluation de la performance en généralisation d'un classifieur a été introduite par Allen(1971) dans le but d'optimiser les choix des variable explicatives dans le modèle linéaire. L'idée derrière cette technique est de tester le modèle du classifieur obtenu sur des données qui n'ont pas participé à l'apprentissage, afin de permettre de prédire le comportement du classifieur face aux nouvelles données.

En pratique, on parle de «*K-fold cross validation*» qui consiste à diviser l'ensemble de validation en k sous-ensembles. On utilise alors un sous-ensemble pour tester le modèle issu de l'apprentissage effectué avec les $k-1$ autres sous-ensembles. Et on répète k fois cette opération à travers tous les k sous ensembles formés. La moyenne des erreurs déterminées lors des k opérations permet de donner une estimation de la capacité de généralisation du classifieur. La variance des résultats obtenus est réduite lorsque k croît, mais l'inconvénient, c'est le temps de calcul qui devient très long [42]

3.3.2. Validation croisée généralisée approchée (GACV)

Wahba et al [49]. Ont développé une approximation de la validation croisée. Malheureusement, si le critère de la validation croisée donne une estimation forte précise des paramètres optimaux (au sens statistique), sa mise en œuvre reste limitée en pratique par un coût numérique élevé. En effet, sous sa forme originale, elle requiert autant d'analyses qu'il y a de données pour calculer une valeur de l'estimateur V_0 . Un tel calcul devrait être répété un certain nombre de fois pour atteindre itérativement le minimum, qui correspond aux paramètres optimaux. Telle quelle, cette méthode est donc clairement inapplicable de façon courante; elle ne peut servir que pour quelques tests isolés, lorsque la base de données n'est pas trop grande et l'optimum facilement localisable.

3.3.3. Technique leave-one-out (LOO)

a) Principe

L'idée derrière cette méthode consiste à prédire la valeur de la propriété pour composé l'ensemble des données, et aussi une meilleure méthode qui est destiné à éviter le biais possible introduit en se fondant sur une division particulière en test et composants ferroviaires, est de partitionner l'ensemble original de plusieurs manières différentes et de calculer un score moyen sur les différentes partitions. L'avantage est que toutes les données peuvent être utilisés pour la formation - ne doit être retenu dans un ensemble de test séparé [85].

b) Algorithme:

- Subdiviser l'échantillon en $K=n$ blocs
- Pour chaque individu k :
- Construire sur le modèle $M(X, n-1)$
- Calculer l'erreur e_k en test sur le k -ième : $\begin{cases} e_k = 1 & \text{erreur} \\ e_k = 0 & \text{bon classement} \end{cases}$
- Calculer la moyenne e_{moy} des erreurs en test (moy proportion des erreurs).

3.3.4. Erreur empirique

La sélection de modèle pour les machines à vecteurs de support en utilisant l'erreur empirique. L'idée derrière cette technique est la minimisation de l'estimation de l'erreur de généralisation à travers une base de validation.

Considérons un problème bi-classe où les observations sont: $\{(x_1, y_1), \dots, (x_N, y_N)\}$ et soit t_i l'étiquette bipolaire des observations devient unipolaire; c'est à dire, $t_i = 0$ pour les observations de la classe négative et $t_i = 1$ pour celles de la classe positive. L'erreur empirique de cette observation est définie par l'expression suivante:

$$E_i = |t_i - P_i| \quad (3.3)$$

Avec P_i L'estimation de la probabilité associée à une observation donnée à partir de la sortie brute d'un SVM est réalisée en utilisant la fonction logistique proposée par Platt [51]. Cette fonction possède deux paramètres A et B et s'écrit:

$$P_i = \frac{1}{1 + \exp(A \cdot f_i + B)} \quad (3.4)$$

Avec $f_i = f(x_i)$ la sortie brute du SVM.

L'utilisation du modèle développé par Platt pour estimer la probabilité permet de quantifier la distance d'une observation à l'hyperplan déterminé par la machine à vecteurs de support à l'aide d'une fonction continue et dérivable. En effet, l'estimation de probabilité permet de réaliser un calibrage de la distance évaluée par $f(x_i)$ entre 0 et 1 de manière que :

- les observations de la classe positive qui sont bien classées et situées en dehors de la marge aient des probabilités estimées très proches de 1 ;
- les observations de la classe négative qui sont bien classées et situées en dehors de la marge aient des probabilités estimées très proches de 0;
- les observations situées dans la marge aient des probabilités estimées proportionnelles à $f(x_i)$

Ainsi, avec l'erreur empirique définie, seules les observations mal classées et celles se trouvant dans la marge déterminée par la machine à vecteurs de support sont très importantes puisque les autres observations donnent des erreurs presque nulles. Par conséquent, la minimisation de l'erreur empirique entraîne la réduction des vecteurs de support (les observations se trouvant dans la marge), la minimisation de l'erreur empirique permet de bien classer les données en dehors de la marge. Ce critère de sélection de modèle est donc utile pour la régularisation de la maximisation de la marge. Entre autre, la réduction de la complexité de la machine avec moins de vecteurs de support permet de réduire la borne de l'erreur en généralisation. L'erreur attendue en généralisation pour une machine à vecteur de support est bornée par :

$$EP_{error} \leq \min\left(\frac{N}{l}, \frac{[R^2|w|^2]}{l}, \frac{d}{l}\right) \quad (3.5)$$

Où :

N : nombre d'exemples de validation;

l : nombre d'exemples d'apprentissage;

R : rayon de la boule englobant les points d'apprentissage;

w : vecteur orthogonal à l'hyperplan optimal;

d : diamètre de la boule englobant les points d'apprentissage.

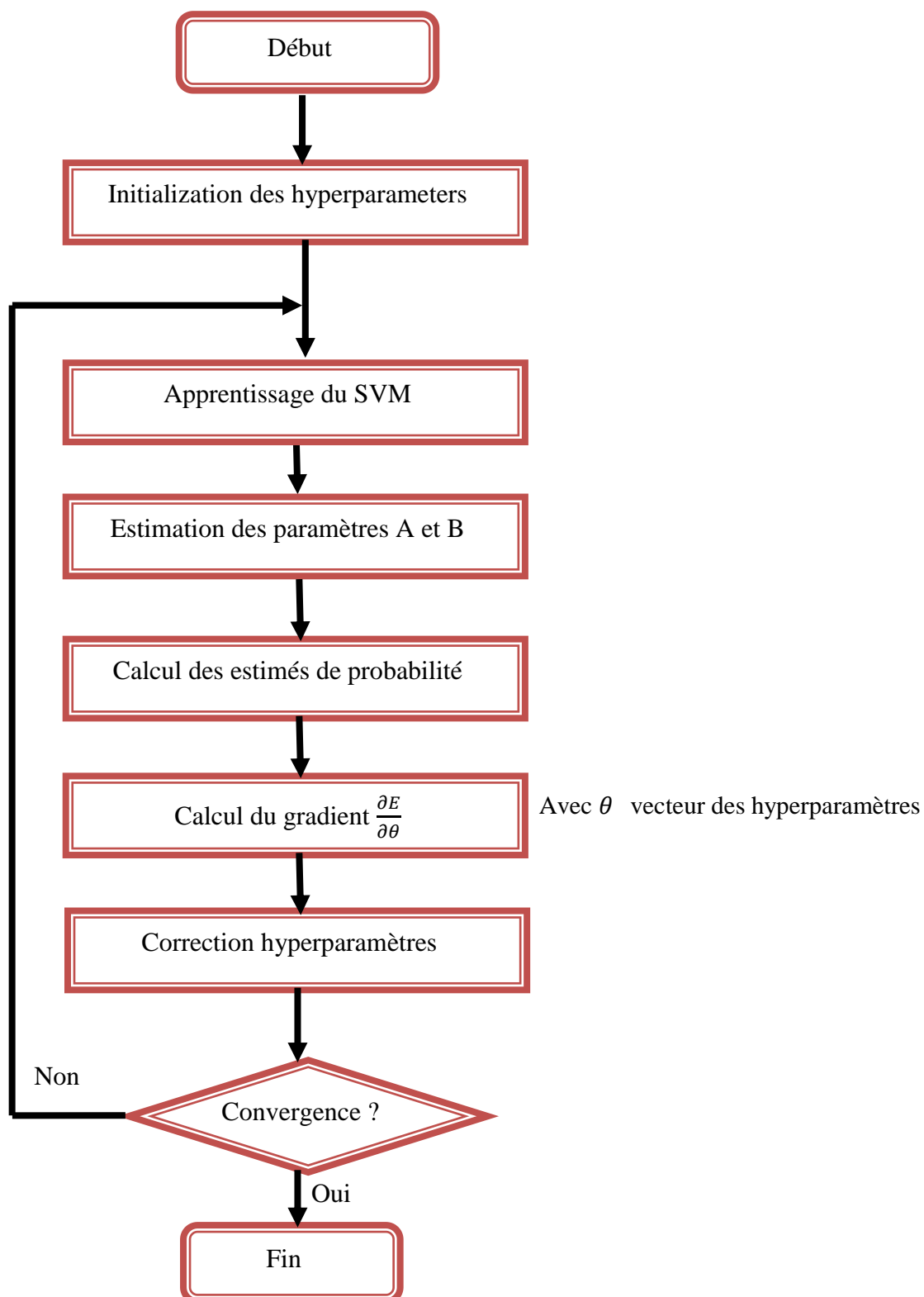


Figure 3.1: Sélection de modèle basée sur l'erreur empirique [42].

3.3.5. Grille de recherche

C'est une méthode classique qui discrétise l'espace des modèles. Leur limite est la sélection de modèles qui exploitent peu de paramètres (≤ 2). Malgré ce défaut, elle permet de tracer des surfaces correspondant à l'évolution des capacités de généralisation d'un SVM en fonction des valeurs de ces deux hyper-paramètres. Cela permet de mieux appréhender ces évolutions, en plus d'une classique sélection de modèle.

Un grand nombre d'expérimentations dans montrent que le paysage de l'erreur en généralisation, à travers l'utilisation de cette "grille", comporte des minima locaux, ce qui illustre que la seule sélection des hyper-paramètres des SVM est un problème difficile en soi. Ces surfaces montrent également que l'évolution de l'erreur en généralisation a des variations peu prononcées lorsque les variations des hyper-paramètres sont faibles. L'optimisation des capacités en généralisation ne nécessite donc pas la recherche de valeurs qui soient très précises pour ces hyper-paramètres, mais d'un ordre de grandeur qui ne soit pas trop grossier. Cela permet de réaliser une recherche efficace avec des pas de discrétisation relativement grands.

3.4. Stratégies d'optimisation

3.4.1. Optimisation des paramètres avec l'apprentissage incrémental

L'apprentissage incrémental permet de construire un classifieur de façon séquentielle. C'est une technique qui consiste à caractériser le modèle du classifieur à partir d'une partie des exemples d'apprentissage tout en conservant la capacité de l'améliorer au cours des étapes ultérieures lorsque de nouvelles données d'apprentissage seront disponibles. Ainsi, lors des étapes d'incrémental, les nouvelles informations contenues dans les nouvelles données sont ajoutées à celles qui étaient mémorisées par le modèle précédemment construit, ce qui permet d'accroître la qualité du classifieur en construction. Ayant plusieurs techniques à savoir :

3.4.1.1. Technique de partition fixe

L'ensemble initial d'apprentissage est découpé en sous-ensembles de taille fixe. À chaque itération, les éléments d'un sous ensemble encore non utilisé sont ajoutés aux vecteurs de support issus de la précédente [52].

3.4.1.2. Technique basée sur l'erreur de classification

À chaque itération, les nouveaux éléments sont d'abord testés avec le modèle courant. Si un élément est mal classé, il est retenu pour participer au nouvel apprentissage avec les vecteurs de

support du modèle courant. Sinon, l'élément est supprimé car on considère que ce dernier n'a pas d'information pertinente pour modifier la frontière de décision [53].

3.4.1.3. Technique basée sur le dépassement de la marge

Dans ce cas-ci, les nouveaux éléments sont testés pour retenir ceux qui sont à l'intérieur de la marge; c'est-à-dire que tous les nouveaux éléments qui sont en dehors de la marge du modèle de SVM courant sont écartés. Ainsi, le nouvel ensemble d'apprentissage est formé des vecteurs de support en plus des nouvelles données qui sont à l'intérieur de la marge provisoire (mal classés ou pas). Avec cette considération, les éléments se trouvant en dehors de la marge et qui sont mal classés sont considérés comme des données aberrantes et sont ignorées dans les prochaines étapes [42].

3.4.2. Optimisation avec Méta-heuristiques

3.4.2.1. Définition :

Les méta-heuristiques sont généralement des stratégies itératives de haut niveau qui contrôlent, guident et modifient une heuristique pour un problème spécifique, afin d'accroître ses performances et produire une solution de haute qualité. L'heuristique peut être une simple procédure, une procédure complexe, une procédure de recherche locale ou bien une méthode de construction. L'objectif principal est d'éviter les inconvénients d'évolution itérative et, en particulier, la descente multiples en permettant la recherche locale pour échapper à des optima locaux. Ceci est réalisé soit en permettant les déplacements détériorés ou de générer de nouvelles solutions de départ pour la recherche locale d'une manière plus "intelligente" plutôt que de les générer aléatoirement. Les méta-heuristiques peuvent manipuler à chaque itération une ou plusieurs solutions [86].

3.4.2.2. Notions et principes de fonctionnements

Pour bien appliquer une méta-heuristique dans un problème d'optimisation, il est nécessaire de bien modéliser les différents paramètres tels que :

- **L'ensemble des solutions** : cet ensemble définit le domaine de recherche d'un optimum.

- **La fonction objective** : qui sert de critère pour déterminer la meilleure solution à un problème d'optimisation. Elle associe une valeur à une instance d'un problème d'optimisation.
- **Voisinage d'une solution** : l'exploration de l'ensemble de solutions consiste à passer d'une solution à une solution voisine. Généralement le voisinage est utilisé dans le cadre d'une recherche locale.
- **L'opérateur de combinaison** : utilisé dans le cadre des méthodes évolutives, il permet de générer de nouvelles solutions à partir de l'ensemble de solutions en suivant une certaine stratégie.
- **L'intensification** : C'est le fait de scruter minutieusement une région prometteuse pour être sûr que l'on a trouvé la meilleure solution. L'intensification n'est pas tout le temps nécessaire, elle est effectuée seulement sur les régions ayant des solutions encourageantes, ou si une recherche normale n'est pas suffisante pour trouver une assez bonne solution.
- **La diversification** : Le principal défaut que l'on reproche aux méta-heuristiques se basant sur la recherche locale, c'est d'être trop locale, c'est-à-dire quelle passent la quasi-totalité de leur temps de recherches sur des partitions restreintes de l'espace de recherche. L'inconvénient est que la recherche peut « rater » des régions, ou les solutions sont très bonnes et se focalise sur des solutions assez lointaines de la solution optimale.
- **Autoriser les solutions infaisables** : C'est le fait d'élargir l'espace de recherche à des solutions qui seraient exclues par certaines contraintes, mais qu'on autorise du fait que leur voisinage soit prometteur est contient potentiellement des solutions de bonne qualité. Cela permet aussi de s'extraire d'optima locaux.

3.4.3. Présentation de quelque méta-heuristique

3.4.3.1 Recherche avec tabous

a) principe de la méthode

La recherche avec tabous est une méthode méta-heuristique pour l'optimisation difficile qui a été développée dans les années 80 par Fred Glover [54, 55]. Elle s'est révélée particulièrement efficace et a été appliquée avec succès à de nombreux problèmes [56, 57]. Elle est basée sur une recherche itérative qui choisit dans un voisinage restreint la meilleure solution

même si elle est plus mauvaise que celle de l'itération précédente. Une mémoire à court terme est utilisée pour éviter tout cycle visitant périodiquement le même optimum local. A partir des derniers mouvements mémorisés, un ensemble de solutions est considéré comme tabou et un mouvement n'est réalisable que vers l'une des solutions voisines qui ne sont pas taboues. Le choix des solutions taboues peut être tout simplement les dernières solutions visitées, mais en général le critère qui détermine les solutions taboues à un moment précis de la recherche dépend du problème à optimiser et de la représentation d'une solution. Lorsque des caractéristiques de modification définissent l'appartenance d'une solution à l'ensemble tabou, l'ensemble des solutions interdites à chaque itération peut contenir des solutions meilleures que toutes celles déjà visitées. Un mécanisme particulier, appelé l'aspiration, permet de pallier l'effet de ne pas pouvoir les sélectionner et de lever ainsi leur statut de tabou [60].

b) Algorithme

1. Choisir solution initiale $s \in S$
2. Initialiser Tabou M
3. répéter
 - Choisir $s' \in V(s)$ telle que: ($f(s')$ meilleure solution de $V(s)$)
 - ET Critère d'aspiration vérifié)
 - OU $f(s')$ meilleure solution de $V(s)$ non taboue
 - $s \leftarrow s'$
 - Mise à jour le Tabou M
4. Jusqu'à Critère d'arrêt vérifié

3.4.3.2. Recuit simulé

a) principe

[65] Kirkpatrick et al (Kirkpatrick *et al.*, 1983) et Cerny (Cerny, 1985) ont été les premiers à s'inspirer d'une telle technique pour résoudre des problèmes d'optimisation combinatoire. Le voisinage $N(s)$ d'une solution $s \in X$ s'apparente à l'ensemble des états atteignables depuis l'état courant en faisant subir des déplacements infinitésimaux aux atomes du système physique. A chaque itération, une seule solution voisine s'est générée. Celle-ci est acceptée si elle est meilleure que la solution courante. Dans le cas contraire, on procède comme dans l'algorithme de Metropolis et al. Et la nouvelle solution s'est acceptée avec une certaine probabilité $\text{prob}(\Delta f, t)$ qui dépend de l'importance de la détérioration $\Delta f = f(s') - f(s)$ et d'un paramètre

t correspondant à la température. Les changements de température sont effectués sur la base d'un schéma de refroidissement précis. En règle générale, la température est diminuée par paliers à chaque fois qu'un certain nombre d'itérations est effectué. La meilleure solution trouvée est mémorisée dans la variable s^* . L'algorithme est interrompu lorsqu'aucune solution voisine n'a été acceptée pendant un cycle complet d'itérations à température constante.

b) Algorithme

1. Engendrer une configuration initiale S_0 de $S : S \leftarrow S'$
2. Initialiser la température T en fonction du schéma de refroidissement
3. Répéter
4. Engendrer un voisin aléatoire S' de S
5. Calculer $\Delta E = f(S') - f(S)$
6. Si $\Delta E \leq 0$ alors $S \leftarrow S'$
7. Sinon accepter S_0 comme la nouvelle solution avec la probabilité $P(E, T) = \exp^{-\frac{\Delta E}{T}}$
8. Fin si
9. Mettre T à jour en fonction du schéma de refroidissement
10. Jusqu'à la condition d'arrêt
11. Retourner la meilleure configuration trouvée le

3.4.3.3. Colonie de fourmi

Depuis quelques années, les ingénieurs s'intéressent au comportement des insectes sociaux afin de créer une nouvelle forme de "résolution collective" de problèmes. L'algorithme de la fourmi, développé initialement par Colorni et al. (Colorni *et al*, 1991) est une méthode évolutive dont les mécanismes de recherche s'inspirent fortement du comportement collectif d'une colonie de fourmis. Dans la phase de coopération, chaque solution de la population courante est examinée à tour de rôle dans le but de mettre à jour une mémoire globale [65]. La phase d'adaptation individuelle fait intervenir ensuite une méthode constructive qui utilise l'information contenue dans la mémoire globale pour créer des nouvelles solutions admissibles. Une telle approche utilise de manière répétée une méthode constructive en faisant intervenir à chaque fois l'expérience accumulée lors des précédentes applications de la méthode. Chaque application de la méthode constructive correspond au travail réalisé par une fourmi isolée.

3.4.3.4. Essaim de particulaire

Par J. Kennedy & R. Eberhart en 1995 le principe est de mettre en jeu des groupes de particules sous forme de vecteurs se déplaçant dans l'espace de recherche. Chaque particule p est caractérisée par deux variables d'état (sa position courante $x(t)$ et sa vitesse courante $v(t)$). Cette technique repose sur deux règles :

- Chaque particule se souvient du meilleur point par lequel elle est passée au cours de ses évolutions et tend à y retourner,
- Chaque particule est informée du meilleur point connu au sein de la population et tend à s'y rendre.

3.4.3.5. Algorithmes génétiques

a) principe

Les algorithmes génétiques ont été conçus par J. Holland, sont de plus en plus considérés comme un cas particulier des algorithmes évolutionnaires. Ils ont été appliqués avec succès dans des problèmes d'optimisation et d'apprentissage automatique. Ceux-ci sont particulièrement efficaces pour parcourir des espaces de recherche à la fois vastes et complexes. Ils ne nécessitent pas notamment de connaissances spécifiques du domaine pour converger vers une bonne solution, mais au contraire mettent en œuvre un cadre général permettant d'évaluer et de recombiner des solutions partielles.

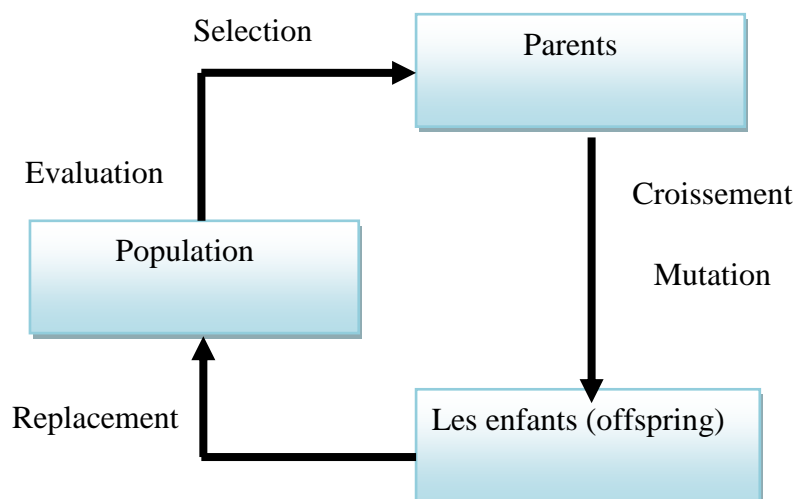


Figure 3.2: cycle d'évolution.

b) Algorithme:

1. initialisation d'une population d'individus ;
2. évaluation des performances de chaque individu ;
3. sélection d'un ensemble d'individus (*parents*) parmi les plus performants pour la reproduction ;
4. génération des individus *enfants* par croisement entre les *parents* ;
5. mutation éventuelle de certains individus (*enfants* et *parents*) ;
6. évaluation des performances des *enfants* (et des *parents* qui ont muté) ;
7. regroupement de tous les individus: *parents* et *enfants*, puis sélection d'un sous ensemble d'individus performants pour la prochaine itération ;
8. Tant qu'une condition d'arrêt n'est pas vérifiée, répéter les étapes 3 à 8 ;
9. Retourner le meilleur individu trouvé pendant l'ensemble de ces itérations.

c) Représentation d'une solution

Comme évoqué précédemment, il est usuel de coder les solutions comme des chaînes binaires, même si la solution d'un problème ne s'exprime pas naturellement sous forme binaire. Dans le cas où une solution ne se représente pas naturellement sous forme de chaîne binaire, il est alors nécessaire de définir une méthode de transformation d'une représentation à l'autre. Mais on peut coder la solution par d'autre code comme les codes décimale et hexadécimale. Par exemple, si on a le problème de voyageur de commerce il est mieux d'utiliser le code décimale pour coder les valeurs de chemins.

d) Opérateurs de base

Dans les méthodes évolutionnaires, les quatre opérateurs de base sont ceux de sélection, de croisement, de mutation et de remplacement. Nous donnons ici une brève description de ces trois types d'opérateurs en n'abordant que les versions les plus courantes.

– Opérateurs de sélection

Dans l'algorithme précédent, l'opérateur de sélection est utilisé aux étapes 3 et 7. C'est un opérateur qui prend en entrée une population et produit en sortie la liste des individus sélectionnés. Dans la majorité des cas, c'est le même type d'opérateur qui est utilisé à ces deux étapes, même si d'un point de vue sémantique ces deux opérations ont une signification

différente. A l'étape 3, la phase de sélection détermine quels sont les individus qui participeront aux reproductions. A l'étape 7, la phase de sélection détermine quels sont les individus qui vont survivre.

On trouve essentiellement quatre types de méthodes de sélection différentes:

- La méthode de la "loterie biaisée"(roulette wheel) de Goldberg,
- La méthode "élitiste",
- La sélection par tournois,
- La sélection universelle stochastique.

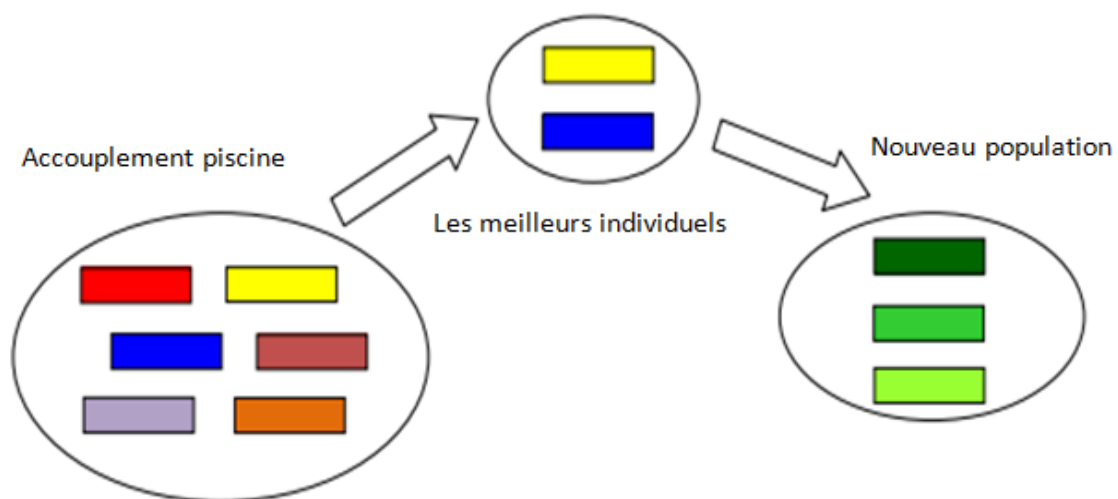


Figure 3.3 : les étapes de sélection de nouvelle population.

La sélection par tournois: la sélection par tournoi est l'une des sélections les plus utilisées dans les algorithmes évolutionnaires. Le principe consiste à choisir un sous-ensemble d'individus (S individus) aléatoirement dans la population, puis à sélectionner le meilleur individu dans ce groupe en fonction de sa fitness. Ce processus est répété jusqu'à l'obtention du nombre d'individus requis (Miller et Goldberg, 1995). Le nombre de participants à un tournoi (S), appelé la taille de tournoi, est utilisé pour faire varier la pression de cette sélection. Si ce nombre est grand, alors la pression sera forte et les faibles individus auront une petite chance d'être choisis. Ce gagnant peut être choisi d'une façon déterministe ou probabiliste. Dans le cas déterministe, qui est pratiquement le plus utilisé, le gagnant est l'individu de meilleure qualité (meilleure fitness). Dans le cas probabiliste, chacun des participants peut être choisi en tant que

gagnant avec une probabilité proportionnelle à sa fitness. Cela consiste à utiliser une procédure de sélection proportionnelle comme la (roulette) [87].

– **Opérateurs de croisement (crossover)**

Cet opérateur a pour but d'enrichir la diversité de la population en manipulant la structure des chromosomes. Classiquement, les croisements sont envisagé individus parents et génèrent deux nouveaux enfants. Pour éviter le rassemblement des individus autour d'un dominant, ils existe plusieurs méthode de croisement permet les on a croisement proposée par Angel Kuri.M [61] qui consiste à faire croiser non plus les deux meilleurs parents, mais plutôt de recombinaison deux chromosomes l'un est le meilleur, l'autre est le plus mauvais (cible).

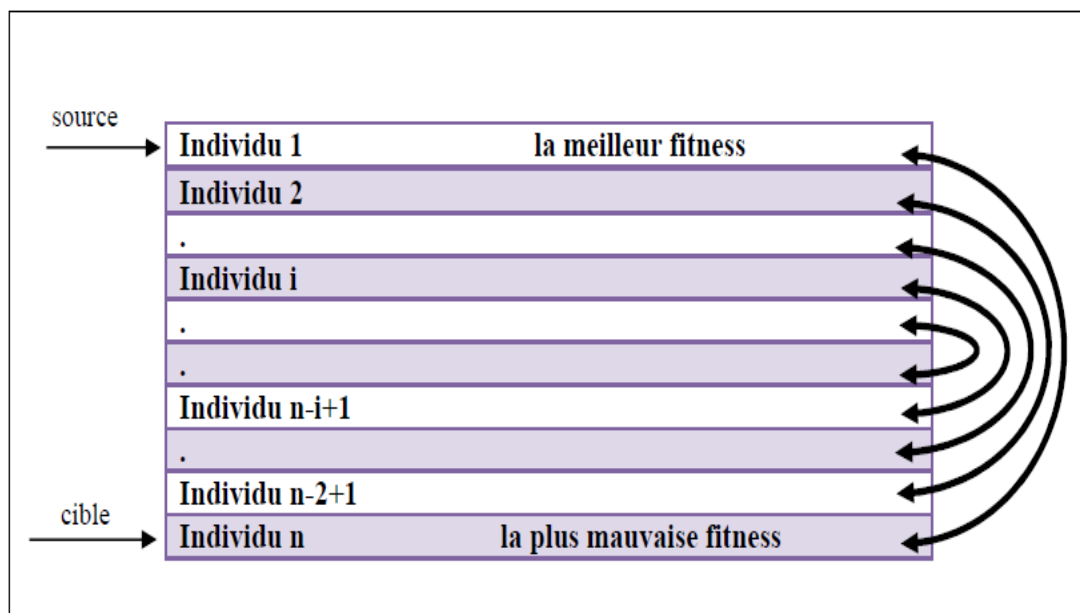


Figure 3.4: Les couples à croiser.

Il reste à définir comment favoriser cet effet. Partons de la considération que toute solution représentée par un individu est un vecteur correspondant à un ensemble de variables dont on recherche globalement les valeurs optimales. L'idée est que chaque solution (individu), même si elle n'est pas optimale, doit contenir une partie des valeurs de la solution optimale (ou proche de l'optimal) recherchée. L'opérateur de croisement a donc pour objectif d'extraire de chaque individu les bonnes bribes d'informations qu'il contient pour les transmettre aux enfants. Pour les algorithmes évolutionnaires, il est courant de ne produire qu'un seul enfant pour chaque croisement. Initialement, il était courant de ne choisir qu'un point de coupure dont la position était choisie par tirage aléatoire, ce qui consiste à échanger deux sous chaînes binaires. D'un

point de vue plus général, le nombre de points de coupure peut être plus important. Ce nombre peut être fixé ou déterminé aléatoirement pour chaque croisement [64].

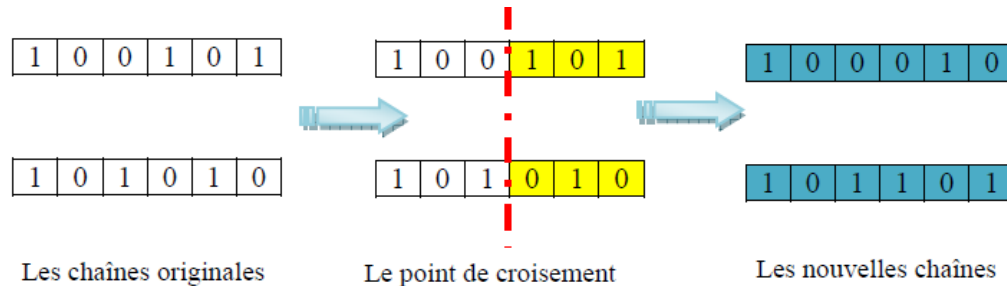


Figure 3.5: Croisement entre deux parents avec 2 points de coupure pour la production de deux enfants par effet symétrique.

Il existe plusieurs techniques de croisement à savoir:

- Order Crossover (OX): choisit deux points aléatoires de croisement;
- One point Crossover (1X);
- N Point Crossover;
- Partially Mapped Crossover (PMX);
- Uniform Crossover (UX);
- Order-Based Crossover (OBX);
- ... etc.

One point Crossover (1X): L'opérateur de croisement le plus simple est le croisement en un point. Initialement, cet opérateur a été proposé pour la représentation binaire. La première étape consiste à choisir aléatoirement un point de coupure pour partager chaque parent en deux parties. Puis le premier enfant est construit en utilisant la première partie du premier parent et la deuxième partie du deuxième parent. A l'inverse, le deuxième enfant est une concaténation de la seconde partie du premier parent et de la première partie du second parent.

– Opérateur de mutation

La mutation traditionnelle consiste généralement, à tirer aléatoirement un gène dans le chromosome et le remplacer par une valeur aléatoire. Ce changement est une étape critique ; il peut rapprocher l'individu vers la solution optimale, comme il peut l'écartier largement et la reprise peut être difficile par la suite. Une des tendances sert à ajuster le taux de mutation, tandis que le taux reste une mesure probable et indéterminée, varie selon le problème rencontré. Vu l'influence de cet opérateur sur la convergence d'un AG vers la solution optimale, et ainsi

d'améliorer le rendement de processus d'apprentissage. Il est spontané de penser à implémenter la mutation d'une manière permettant au système d'éviter le problème de convergence prématuré ou d'écarter de la bonne solution. Plusieurs opérateurs de mutation existent:

- **Insertion:** l'opérateur d'insertion choisit un gène aléatoirement et l'insère dans une autre position du chromosome.
- **Échange (1-Opt ou Swap mutation):** l'opérateur d'échange consiste à prendre au hasard deux gènes (2 clients) du chromosome et à les permuter.
- **Inversion:** l'inversion inverse l'ordre de visite des individus entre deux points de coupure choisis aléatoirement.
- **Mélange (Scramble Mutation ou SM):** Elle permute aléatoirement les individus entre deux points de coupure choisis au hasard (figure 3.6)

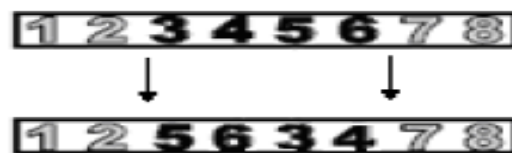


Figure 3.6: Exemple de Scramble Mutation.

– L'évaluation et la sélection pour remplacement

Après l'application des opérateurs de reproduction (variation), chaque individu dans la population des enfants est évalué. Puis, la sélection pour le remplacement ou plus simplement le remplacement détermine quels individus devront disparaître de la population à chaque génération et quels individus forts survivent dans la population de la prochaine génération [87].

a) Remplacement générationnel

Cette stratégie est proposée par (Holland, 1975). Ici, la population des enfants remplace systématiquement la population courante. Ainsi, la population de prochaine génération est égale à la population des enfants.

b) Le remplacement élitisme

L'élitisme est une façon de protéger la rémanence de bonnes solutions et d'assurer leur survie tout au long de la recherche. Ici, la population de la prochaine génération est choisie à partir de la population des enfants et de la population courante ensemble. Cette sélection a l'avantage de permettre une convergence (plus) rapide des solutions, mais au détriment de la diversité des individus.

c) Remplacement des stratégies d'évolution

Dans les stratégies d'évolution, la taille de la population de la prochaine génération est plus petite que la taille de la population des enfants. Soit on choisit les meilleurs individus de la population des enfants, soit on les choisit de la population des enfants et la population courante conjointes

– Critère d'arrêt

Généralement, le cycle de génération et remplacement est répète jusqu'à ce qu'un critère d'arrêt soit satisfait. Ce critère peut être notamment un nombre d'itérations (générations), un temps maximal de calcul, ou/et une solution satisfaisante. L'algorithme évolutionnaire retourne alors la (ou les) meilleure(s) solution(s) qu'il a identifié(s) de génération en génération.

3.4.3.6. Optimisation multi-objectif

Il est courant de devoir optimiser plusieurs objectifs en même temps. Le problème le plus courant est de trouver des solutions à des problèmes qui soit à la fois de la meilleure qualité et du coût le plus réduit [60]. Ce qui correspond généralement à choisir un bon compromis entre les deux. La solution la plus couramment utilisée pour réaliser cette optimisation est de définir une nouvelle fonction objective à partir de l'ensemble des autres. Par exemple, par une combinaison linéaire de ces objectifs. Les valeurs des coefficients de cette combinaison correspondent alors à la caractérisation du compromis choisi.

3.5. Conclusion

Nous avons présenté dans ce chapitre les différentes méthodes de sélection de modèle SVM, ainsi que les différentes méthodes méta-heuristiques. Nous avons indiquons qu'un grand nombre de problèmes d'apprentissage correspondaient à l'optimisation d'un problème difficile.

A travers quelques références, nous montrons que des méthodes à base de méta-heuristiques ont déjà été appliquées avec succès à des problèmes d'optimisation.

Nous avons détaillons plus particulièrement dans ce chapitre les méta-heuristiques correspondant aux algorithmes évolutionnaires (algorithmes génétiques).

Chapitre IV

Résultats expérimentaux

et discussions

4.1. Introduction

Dans ce dernier chapitre, nous présentons les différentes étapes réalisées durant l'implémentation de notre application, nous commençons à décrire la création de la base de données, ensuite nous définissons les fonctionnalités de notre application et enfin nous discutons les résultats obtenus.

Nous avons utilisé un modèle de sélection basé sur les algorithmes génétiques, en analysant les résultats aboutis. Nous illustrant notamment l'effet des paramètres choisis sur la qualité des solutions trouvées.

4.2. Outils d'implémentation

Notre domaine d'application exige la réalisation d'un système de sélection et de reconnaissance. Dans ce que suit nous détailleront les outils nécessaires pour accomplir cette tâche.

4.2.1. Environnement logiciel

JAVA est un langage orienté objet qui apporte un support efficace et élégant en définissant de manière très stricte la façon dont les objets communiquent entre eux. Le principal avantage est que chaque objet puisse être mis au point séparément ainsi que :

- JAVA est extensible sans aucune limitation car pour étendre le langage il suffit de définir de nouvelles classes [81].
- JAVA est un langage fiable, sécuritaire et polyvalent.
- Un autre avantage de ce langage de programmation réside dans le fait que la syntaxe de Java est analogue à celle de C++ ce qui le rend économique et professionnel [81].
- le développement d'applications en Java est beaucoup plus rapide que la majorité des langages de programmation

Notre choix s'est porté sur NetBeans parce qu'il est écrit en Java, il est relativement simple à manipuler et possède des outils intégrés permettant un traitement aisé des résultats comme les illustrations graphiques.

4.2.2. Bibliothèque SVM ^{multi classe} utilisée

L'algorithme réalisé dans cette application a été développé en se basant sur SVM multi classes. C'est une librairie de SVM écrite en langage C et conçue par Thorsten Joachims, chercheur à l'université de *Dortmund*. Elle est parmi les implémentations les plus connues de SVM. Elle présente plusieurs avantages :

- Extensible;
- Il supporte les ensembles d'apprentissage de grande taille ;
- Paramétrable et fournie les fonctions noyaux connues dans SVM.
- Il donne la main à l'utilisateur de définir son propre noyau suivant ces besoins ;
- Est un algorithme optimisé en temps d'exécution et en utilisation de mémoire.
- permet de résoudre trois types de problèmes qui sont : la reconnaissance des modèles, la régression et l'apprentissage par tri.

Le code source de SVM multi classe est disponible sur le site officiel de joachims [82]. Il peut être compilé pour toutes les plateformes. La relation entre notre application et le package SVM multi classe peut être vu selon un modèle de trois couches:

- **Application:** qui représente notre application;
- **Interface:** qui permet la communication entre l'application et les modules de calcul;
- **Calcule:** qui permet de réaliser les calculs nécessaire.

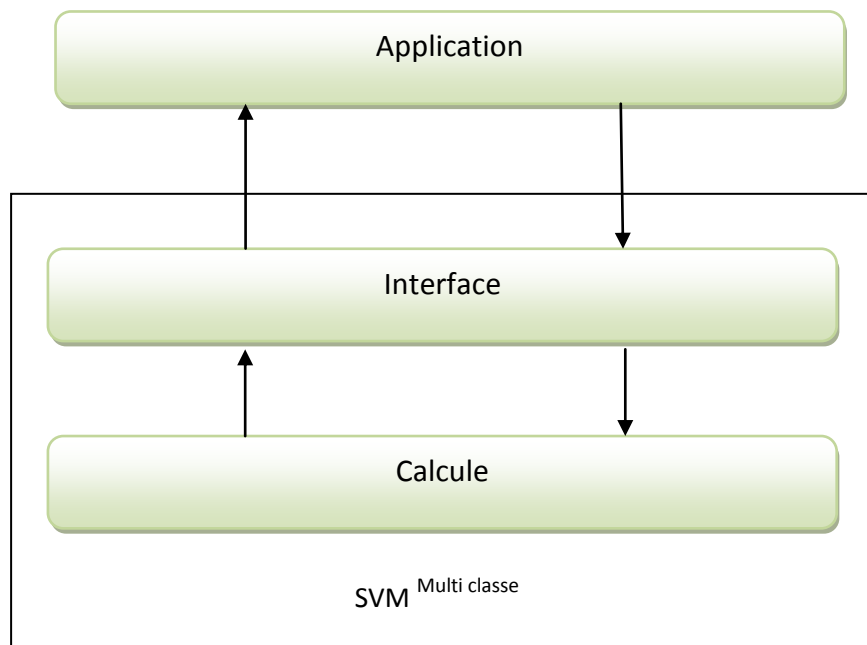


Figure 4.1 : Illustration de la relation entre notre application et SVM ^{multi classe}.

La couche interface est présentée par les deux modules:

- **train_sparse:** pour la phase d'apprentissage.
- **predict_sparse:** pour la phase de test (prédiction).



Figure 4.2 : Fenêtre principale de notre application LACDS.

4.3. La description de base des données utilisée

La base de données utilisée dans ce travail est constituée d'un ensemble de 3920 images de caractères arabes manuscrits (2940 pour l'ensemble d'apprentissage, 980 pour l'ensemble de test et validation), chaque lettres se trouvent dans les différentes positions (isolée, début, milieu et fin).

La base de test constitue 30% de la base totale 5% de cette base pour la validation. La (figure 4.3) montre la décomposition de la base construite.



Figure 4.3: Décomposition de la base des caractères.

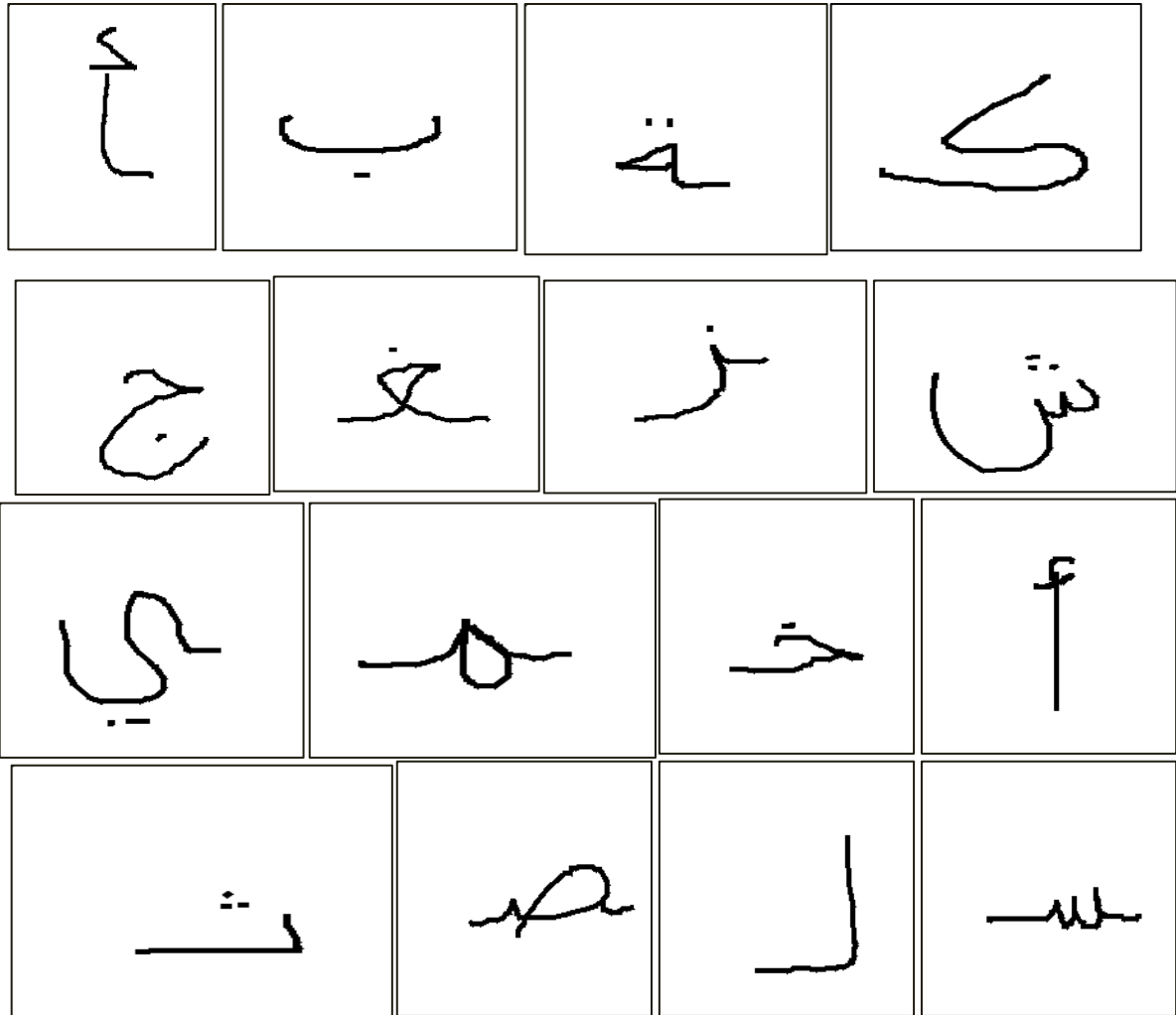


Figure 4.4: Des exemples de notre base de données.

Le tableau 4.1 indique quelques détails sur la base d'apprentissage. Le nombre des classes est 98, et le nombre d'échantillons par classe est 30 provenant de 3 scripteurs, chacun écrit le caractère 10 fois et chaque caractère est dans les quatre formes (début, milieu, fin et isolé).

Classe	Caractère	Nb scripteurs	Echant/script	Echant/classe
1	ا	3	10	30

2	ل	3	10	30
.
.
98	ﻉ	3	10	30
Totale	2940			

Nb : nombre de scripteurs, Echant/script : nombre d'échantillons écrits par chaque scripteur, Echant/classe : nombre d'échantillons par classe.

Tableau 4.1: Description de la base d'apprentissage.

Le tableau suivant décrit la base de test indépendante:

Classe	Caractère	Nb scripteurs	Echant/script	Echant/classe
1	ا	2	5	10
2	ل	2	5	10
.
.
98	ﻉ	2	5	10
Totale	980 caractères			

Nb : nombre de scripteurs, Echant/script : nombre d'échantillons écrits par chaque scripteur, Echant/classe : nombre d'échantillons par classe.

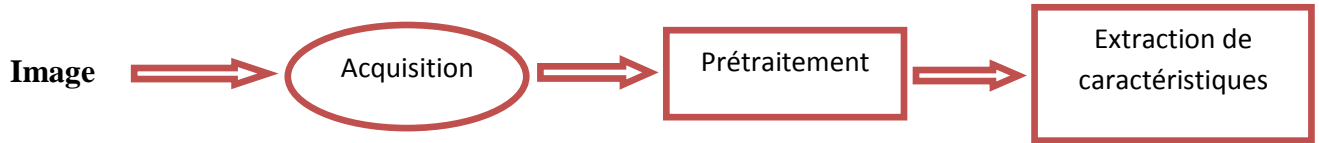
Tableau 4.2: Description de la base de test indépendante.

4.4. Description de notre système proposé LACDS

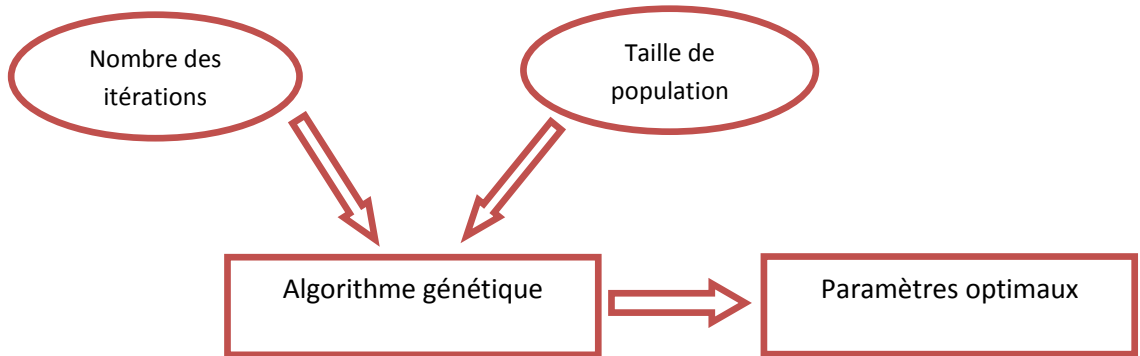
L'objectif que nous nous sommes assignés s'articule autour du développement d'un système pour la reconnaissance des caractères arabes manuscrits et la sélection des paramètres SVM.

Compose de trois phases supplémentaires:

Premier phase: "Extraction des caractéristiques des caractères"



Deuxième phase: "Sélection des paramètres optimaux"



Troisième phase: "Apprentissage et classification"

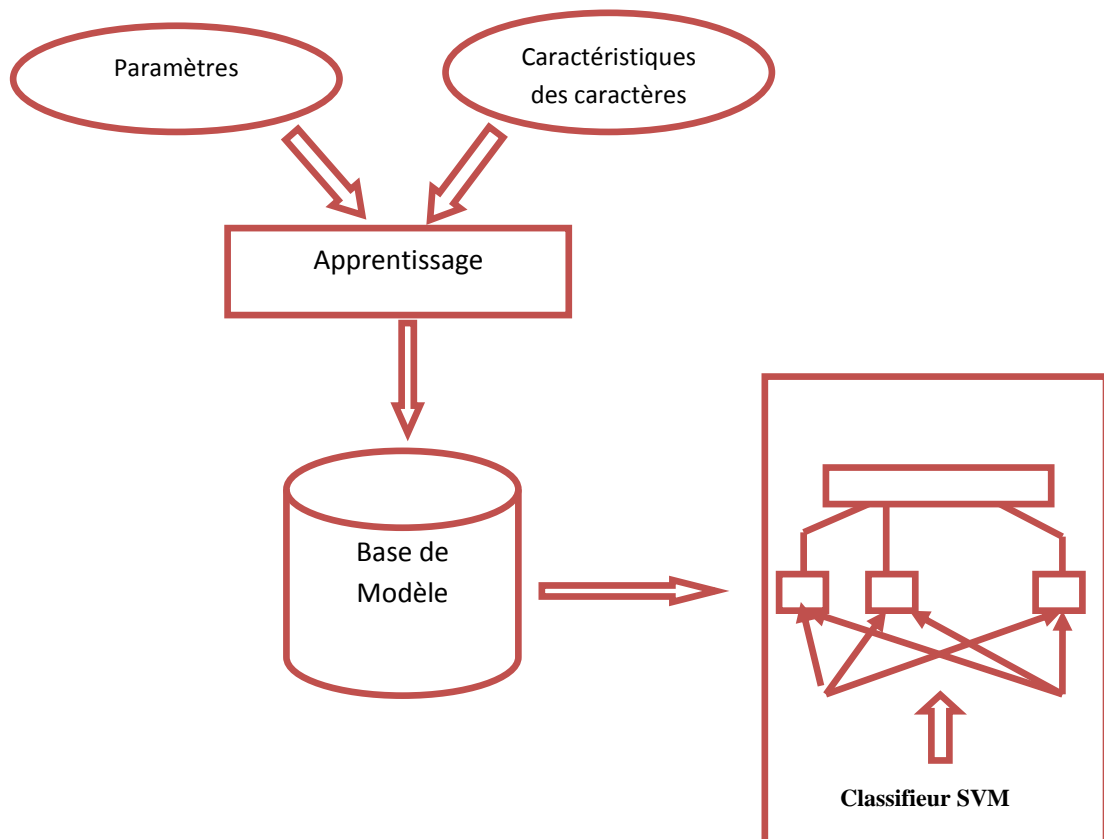


Figure 4.5: Architecture du système LACDS.

4.4.1. L'acquisition

L'acquisition permettant la conversion du dessin sous la forme d'une image numérique. La numérisation est opérée directement sur le dessin. Le résultat est range dans un fichier de points, appelés pixels. Dans notre système nous utilisant une seule matérielle de saisir (la souris).

4.4.2. Prétraitement

Le prétraitement inclut toutes les fonctions effectuées avant de commencer le traitement pour produire une version << nettoyée >> de l'image origine afin qu'elle puisse être utilisée directement et efficacement. Ainsi le prétraitement comprend la binarisation, la normalisation et l'encadrement.

a) Binarisation

La binarisation est la première étape de prétraitement elle consiste à convertir l'image numérisée en une image binaire. Cependant, la binarisation est une opération qui produit deux classes de pixels, en général, ils sont représentés par des pixels noirs et des pixels blancs. Ainsi les pixels correspondant a des points élevés doivent être binarisés en noire (valeur=1) et ceux dans les creux doivent être binarisés en blanc (valeur=0). La figure 4.6 montre une image avant est après binarisation.



Image avant binarisation

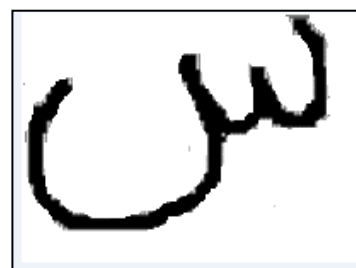


Image après binarisation

Figure 4.6 : Exemple de binarisation d'une image.

b) Encadrement

L'encadrement c'est le processus de localisation de la lettre, c'est de définir les coordonnées de la lettre dans l'image. Pour cela nous avons crée une fonction qui permet de

donner les propriétés suivantes : haut, bas, gauche, droite, afin de passer à l'encadrement de la lettre.

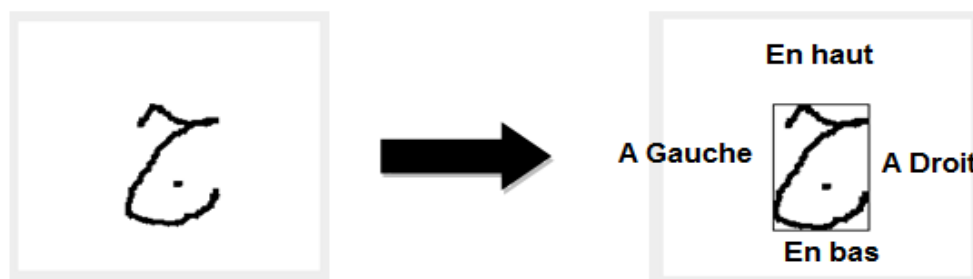


Figure 4.7 : Exemple d'encadrement d'une lettre dans une image.

c) Normalisation (Adaptation)

La normalisation consiste à transformer la taille de l'image et l'adapter à une dimension fixée a priori par l'utilisateur (voir figure 4.8), pour cela nous avons proposés une procédure qui permet de normaliser l'image encadrée dans une dimension de 70*70 pixels.

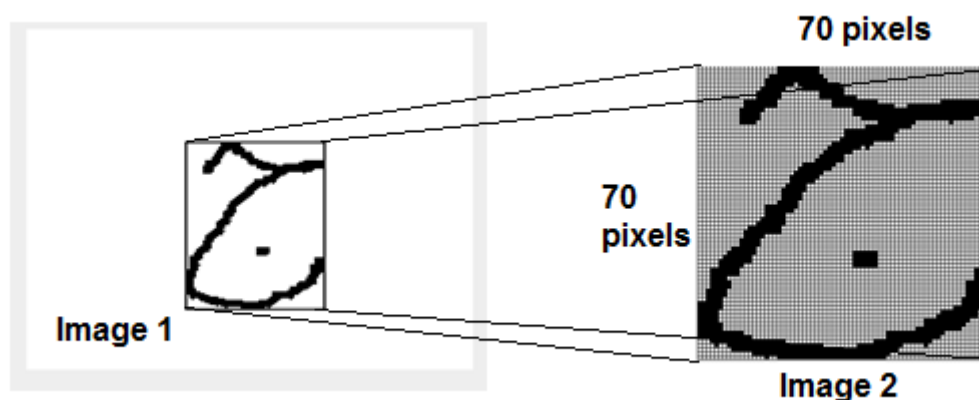


Figure 4.8: Exemple de normalisation d'une lettre.

4.4.3. Extraction de caractéristiques

Dans cette étape, on a utilisé le principe de matrices de distribution pour extraire les caractéristiques (primitives) des images des lettres puis on les stocker dans une base de données (corpus).

a) Construction de la matrice de distribution

La construction de la matrice de distribution est l'une des phases importantes dans notre système. On suppose qu'on a une image comme celle de la figure 4.9 contenant la lettre 'jim' encadrée de N x N pixels, on applique maintenant une matrice de 5x5 sur cette lettre. Pour une matrice de distribution de taille N x N, le principe consiste à superposer une grille (N×N) sur l'image du caractère et calculer pour chacune des cellules [i,j] résultantes le nombre de pixels noir, puis attribuer ce nombre à la case[i,j] de la matrice de distribution.

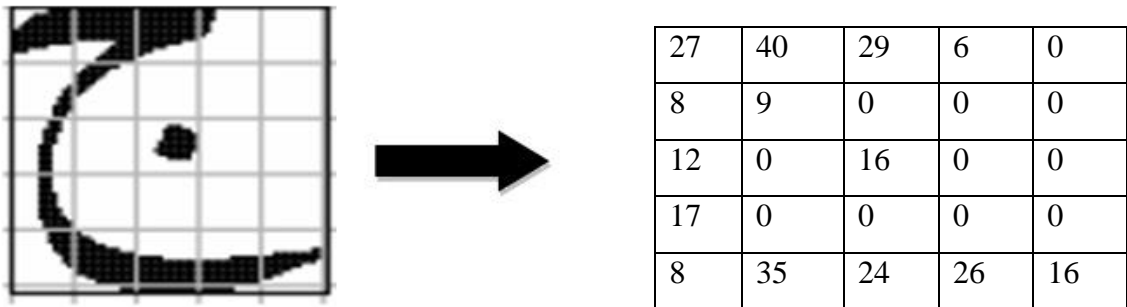


Figure 4.9 : Exemple de matrice de distribution (5*5) de la lettre "Jim".

Pour notre système proposé LACDS nous avons utilisé une matrice de distribution de taille 7 x 7 et l'image de caractère est de taille 70 x 70 après la normalisation. Nous présentons ci-après notre algorithme qui permet de construire la matrice de distribution :

```

M : Image de lettre 70 x 70;
nb : compteur ;
MD : Matrice de distribution 7 x 7 ;
I, J, L, H: entier ;
L=0 ; H=0 ;
  Tant que (L < 70) faire
    H=0;
    Tant que (H < 70) faire
      nb=0 ;
      Pour ( I=L jusqu'à L+10)
        Pour ( J=H jusqu'à H+10)
          Si pixel M[I,J] est noire alors nb=nb+1 ;
        Fin pour
      H=H+10;
    L=L+10;
  Fin tant que
    
```

```

MD[L div 10, H div 10]=nb ;
H=H+10 ;
  Fin Tant que
L=L+10 ;
Fin Tant que
    
```

b) Vectorisation

Le vecteur de caractéristique est de taille 49, représentent les valeurs des sous matrices de matrice de distribution :

0	1	2	3	4		11	12		24
27	40	29	6	0	12	0	16

Figure 4.10: Exemple de vecteur des caractéristiques de matrice 5*5.

c) Base de modèle (corpus d'apprentissage)

Dans cette partie nous allons extraire les modèles des lettres pour chaque scripteur dans ses différentes formes. Nous allons analyser chaque modèle et d’extraire leurs matrices de distributions afin de les enregistrer dans la base de données d’apprentissage (figure 4.10). Le schéma général d’apprentissage est illustre par la figure 4.11

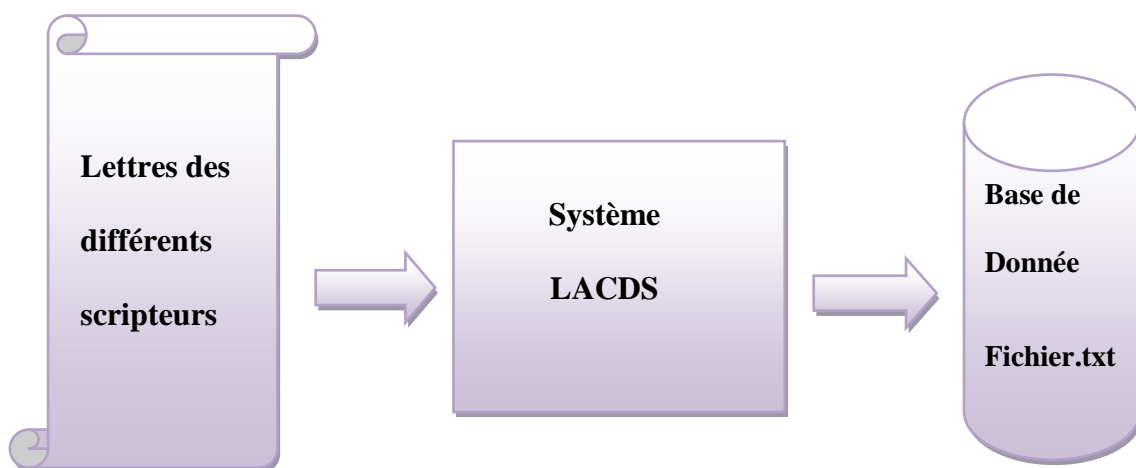


Figure 4.11 : Modèle d’apprentissage.

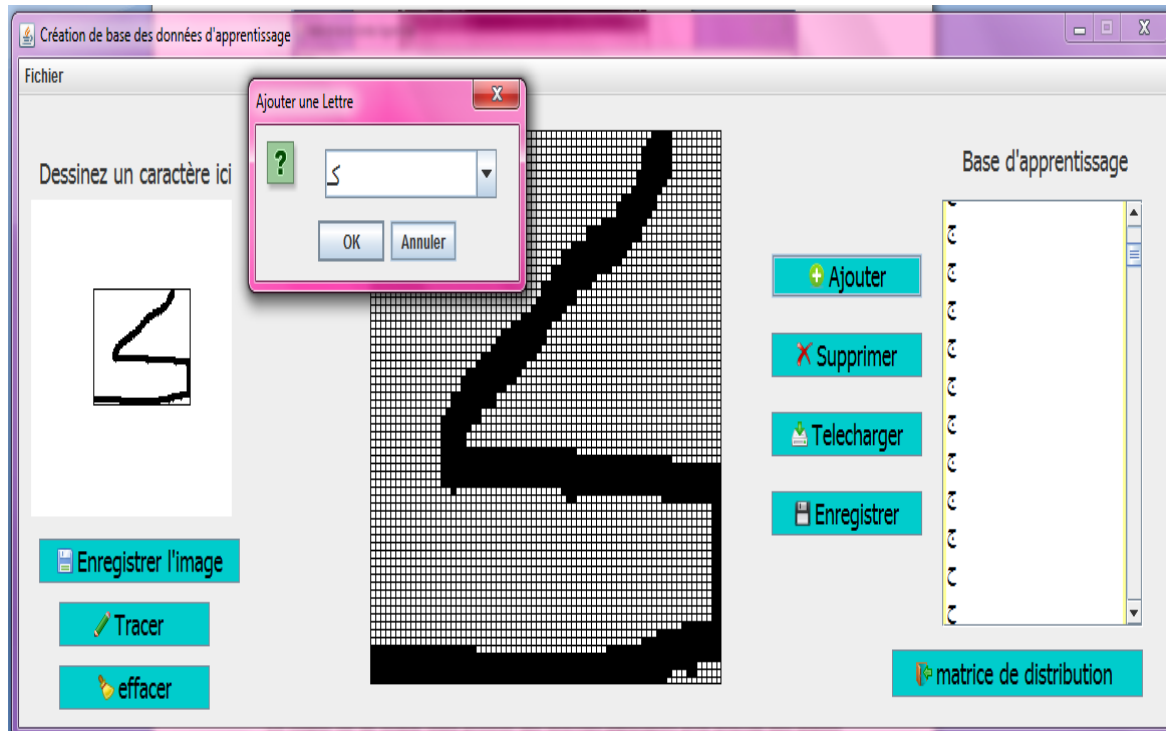


Figure 4.12: la création de corpus par le système LACDS.

Le corpus est un fichier texte possède une structure particulière pour pouvoir être utilisée directement par l'application SVM multi classe:

<label> <index 1> : <valeur 1> <index 2> : <valeur 2> <index N> : <valeur N>

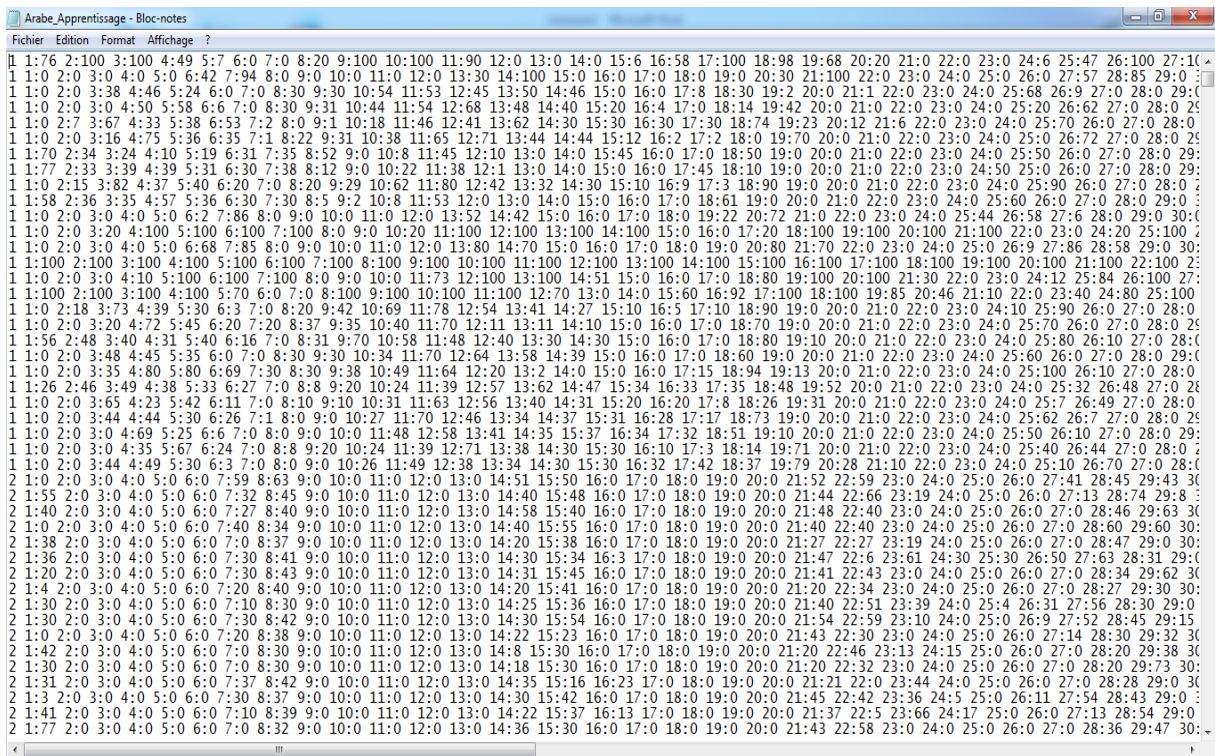


Figure 4.13 : Forme des données.

4.4.4. Apprentissage

Après l'extraction de caractéristiques, et la création de corpus des données d'apprentissage (data file), ce fichier est l'un des paramètres passés au programme d'apprentissage des SVM multi classe (train_sparse) pour la création du fichier modèles.

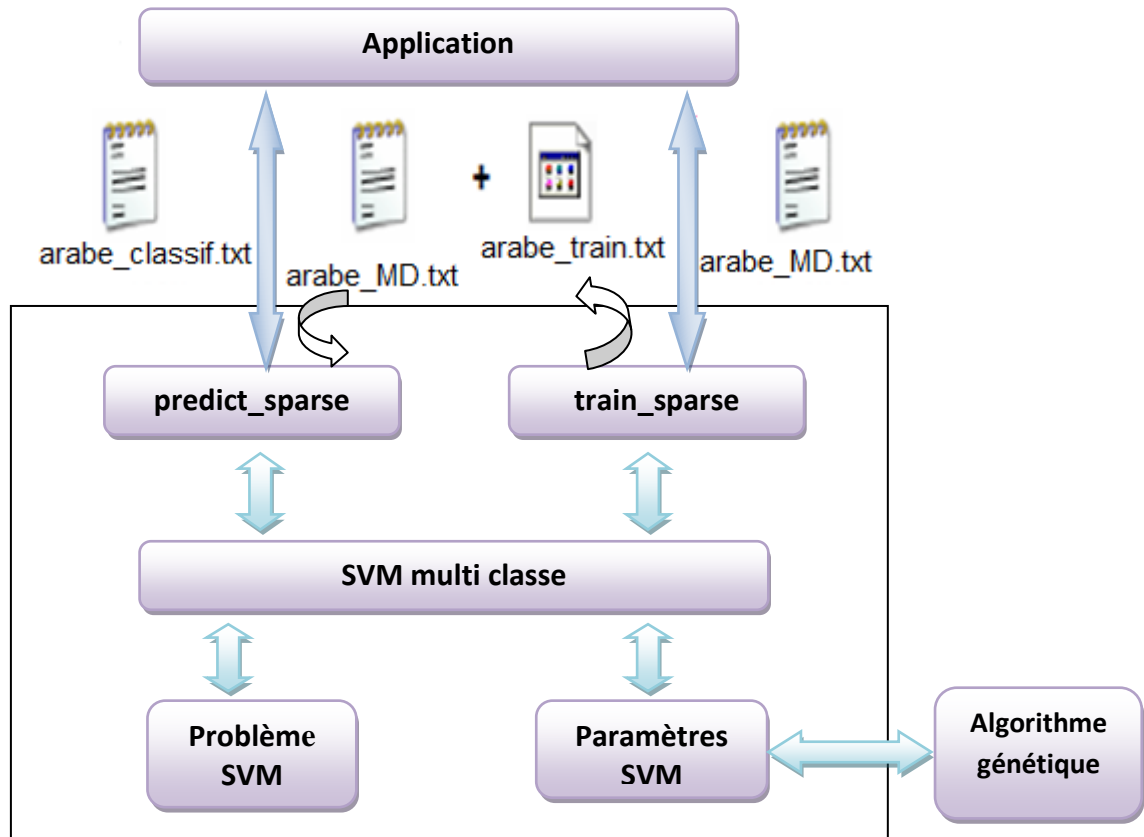


Figure 4.14: relation entre SVM^{multi classe} et notre application.

4.4.5. Description d'algorithme génétique

Les paramètres de la population doivent être fixés avant l'exécution de l'algorithme et ils ont une importance marquée dans la résolution du problème.

- **Nombre d'individus de la population:** ce paramètre vise à fixer le nombre d'individus dans la population pour toute la durée de l'exécution de l'algorithme. La taille de la population ne doit pas être trop grande car après une certaine limite la performance de l'algorithme diminue. En effet un nombre d'individus trop élevé affecte la rapidité de la résolution du problème.
- **Paramètre des conditions d'arrêt:** l'arrêt de l'algorithme dans la recherche de la meilleure solution intervient dès qu'un des paramètres atteint la valeur fixée

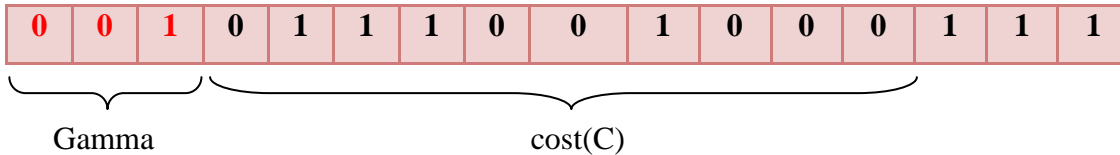
4.4.5.1. Codage

Nous utilisons une structure de donnée basée sur le code binaire pour implémenter chaque individu.

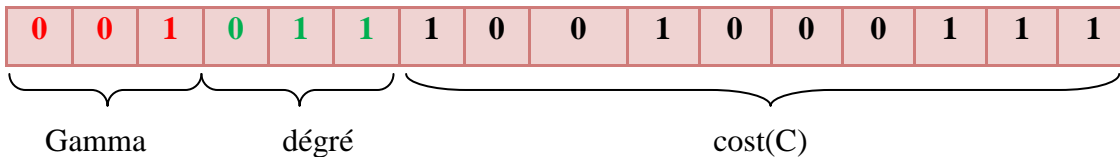
-La taille totale d'individu (16 bits) :



-La taille d'individu pour le noyau RBF :



-La taille d'individu pour le noyau Polynomial :



-La taille d'individu pour le noyau Sigmoide :

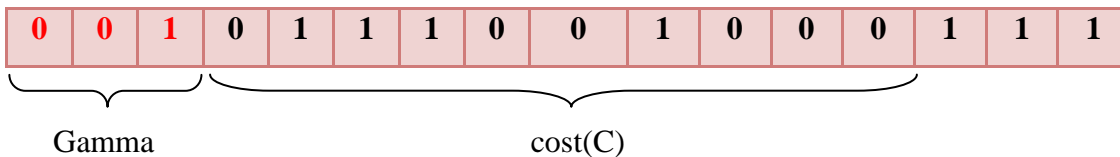


Figure 4.15: structure d'individu.

4.4.5.2. Configuration d'AG:

Paramètre	Valeur
Nombre d'individus	Convertible
Nombre des générations	Convertible
Probabilité de mutation	0.015
La taille de tournoi	5
Taille d'individu	16 bit

Tableau 4.3: Description des paramètres utilisés pour l'AG.

4.4.5.3. Algorithme génétique utilisé dans le processus de sélection

Les étapes de l'AG de notre système définie comme suite :

- 1) Initialisation de population, se fait aléatoirement et la taille des individus reste fixe (16 bit) avec un code binaire;
- 2) En suit, on va évaluer tout les individus un par un (calculer valeur de fitness). la valeur de fitness représenté par le taux de reconnaissance (accuracy);
- 3) Nous avons utilisé la méthode de sélection par tournois pour la sélection des individus la taille de tournois égale à 5;
- 4) Croisement : pour générer les individus enfants on utilisant la technique de "*One point Crossover*" nous fixons ce point au milieu (l'enfant hérite 50% de chaque parent);
- 5) Après en passe à la mutation par l'opérateur d'échange (swamp) consiste à changer de 0 à 1 et l'inverse;
- 6) On passe à l'évaluation et la sélection pour le remplacement avec la stratégie de "*remplacement élitisme*" c'est-à-dire en garder toujours la meilleure solution (Fittest) et les autre individus sera remplacer par le croisement que nous avons indiqué a l'étape quatre;
- 7) Si condition d'arrêt n'est pas vérifiée (nombre d'itérations maximale ou la valeur maximale de fitness 0.989), répéter les étapes précédents;
- 8) En fin retourner meilleure solution trouvée.

4.5. Tests, résultats et discussions

Nous avons ajouté une extension au système LACDS qui se base sur la reconnaissance des caractères arabes manuscrits avec un processus de post-traitement supervisé (SVM).

Durant les expérimentations effectuées, nous avons utilisé trois types des noyaux, polynomial, RBF et sigmoïde avec les deux stratégies *une contre une* et *une contre reste*.

4.5.1. Résultat obtenue

- Noyau RBF $K(x_i, x_j) = \exp \frac{-\|x_i - x_j\|^2}{2\delta^2}$

Nous avons précisé deux paramètres pour la sélection, paramètre *Gamma* δ et le paramètre C avec le choix de quelle stratégie, que ce soit *une contre une* ou bien *une contre reste*.

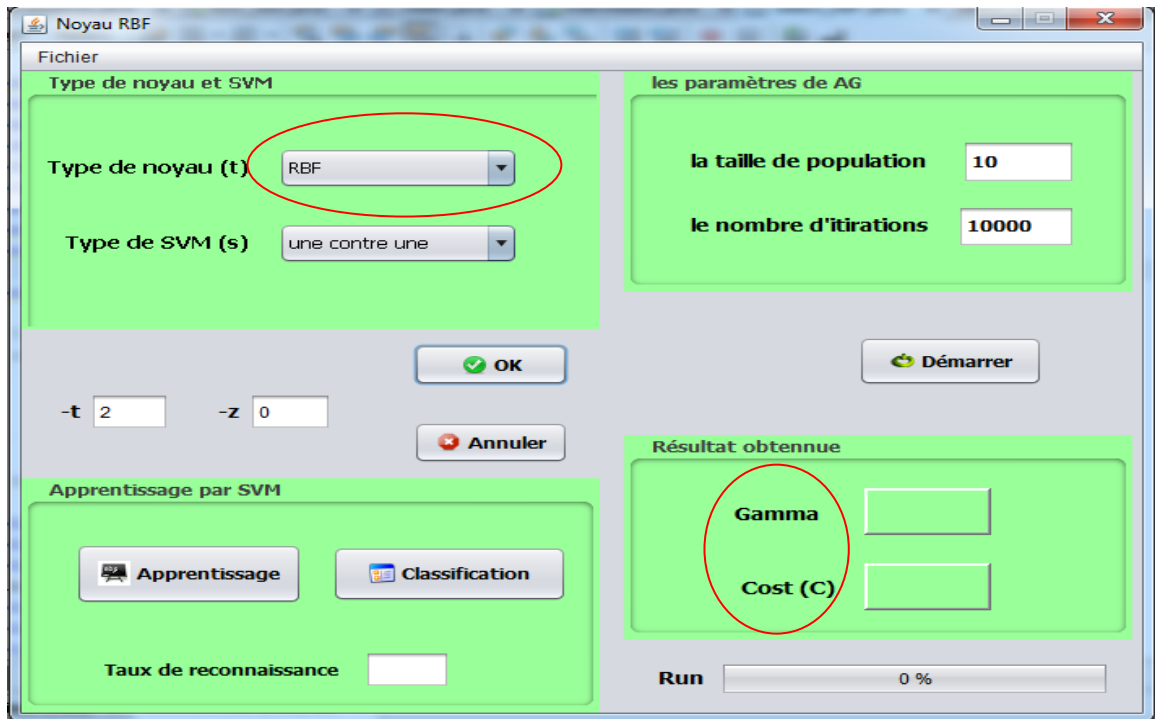


Figure 4.16: permet les interfaces de sélection des paramètres par AG (noyau RBF).

Exemples des résultats obtenus:

Une contre une				
Paramètre C	Paramètre δ	Temps d'apprentissage	Temps de classification	Taux de Reconnaissance
554	1	8.105s	2.158s	79.18%
799	0.25	8.764s	2.49s	98.30%
882	3	8.060s	1.85s	98.10%
1000	0.5	8.990s	2.125s	99.89%

Une contre reste				
Paramètre C	Paramètre δ	Temps d'apprentissage	Temps de classification	Taux de Reconnaissance
265	1.25	8.98s	1.63s	89.11%
552	2	236.90s	1.59s	63.98%
620	0.25	268.47s	1.61s	90.00%
1000	3	8.06s	1.85s	98.91%

- **Noyau Polynomial** $K(x_i, x_j) = (coef_0 + \delta x_i \cdot x_j)^d$

La même chose pour le noyau polynomial mais cette fois avec trois paramètres Gamma δ , le paramètre C et le paramètre de degré d. le paramètre **coef₀** = 0 (valeur par défaut).

Et comme exemples des résultats obtenus :

Une contre une					
Paramètre C	Paramètre δ	Paramètre d	Temps d'apprentissage	Temps de classification	Taux d'apprentissage
1440	0.5	8	6.68s	1.40s	89.63%
1420	0.25	6	6.64s	1.24s	99.18%
1000	0.75	1	5.83s	1.28s	99.80%
1200	1	4	5.06s	1.23s	90.75%

Une contre reste					
Paramètre C	Paramètre δ	Paramètre d	Temps d'apprentissage	Temps de classification	Taux d'apprentissage
818	1.5	3	7.28s	1.26s	91.11%
1400	0.25	4	8.20s	1.22s	99.19%
384	1.25	2	6.06s	1.17s	99.00%
1011	0.75	3	7.05s	1.30s	98.26%

- **Noyau Sigmoid** $K(x_i, x_j) = \tanh(\delta x_i \cdot x_j + coef_0)$

Aussi pour ce noyau avec les deux paramètres Gamma δ et C et le choix entre les deux stratégies.

Une contre une				
Paramètre C	Paramètre δ	Temps d'apprentissage	Temps de classification	Taux de Reconnaissance
420	1	6.33s	1.03s	01.90%
663	0.5	6.39s	1.29s	29.00%
729	2.5	6.36s	1.93s	19.20%
1000	0.75	6.62s	1.29s	09.81%

Une contre reste				
Paramètre C	Paramètre δ	Temps d'apprentissage	Temps de classification	Taux de Reconnaissance
819	0.3	2.25s	1.03s	0.08%
1022	0.75	3.36s	1.29s	0.08%
746	2	2.51s	1.39s	0.08%
1000	1	2.44s	1.29s	0.08%

4.5.2. Évaluation de système LACDS

Le but des expériences adressées dans cette partie est de présenter les performances des SVM avec différents noyaux. D'après les résultats, on a remarqué le taux d'erreur est très élevé en utilisant un noyau polynomial ou sigmoïde, par contre, le taux de reconnaissance est significativement augmenté avec un noyau RBF.

Le noyau RBF a donné des résultats meilleurs que le noyau polynomial et sigmoïde, particulièrement avec $\delta = 0.5$, $C=1000$ et la stratégie *une contre une* cette dernier est plus efficace et plus rapide que l'*une contre reste*, nous avons acquis à 98.89 %. Le noyau RBF présente un temps d'apprentissage beaucoup plus long que les deux noyaux.

Le taux de reconnaissance ne dépasse même pas 1% avec le noyau sigmoïde, donc ce noyau est très faible.

Dans notre application nous avons ajouté la possibilité d'effectuer des teste rapide, on introduisant la lettre et le système affiche le résultat immédiatement.



Figure 4.17: interface de test séparable de notre système LACDS.

La figure 4.23 représente un processus évolutif typique de l'AG avec le nombre des générations concernant le noyau RBF. Cette procédure est caractérisée par cinq phases.

À la première phase, les valeurs de taux de reconnaissance (fitness) augmente progressivement à partir de la valeur d'initialisation (de 8%) à 79.10% jusqu'à la génération 100. Deuxième phase, les valeurs montent plus lentement (de 79.10%) à 98.90% à la génération 1000, puis la troisième phase les valeurs remisent en forme reste le même: 98.90% jusqu'à la génération 2500. La quatrième phase sont augmente à 99.89% à la génération 3000, et la dernière phase est caractérisée par la stabilité, lorsque le numéro de génération atteint 3000, la valeur maximale est obtenue est reste le même (99.89%).

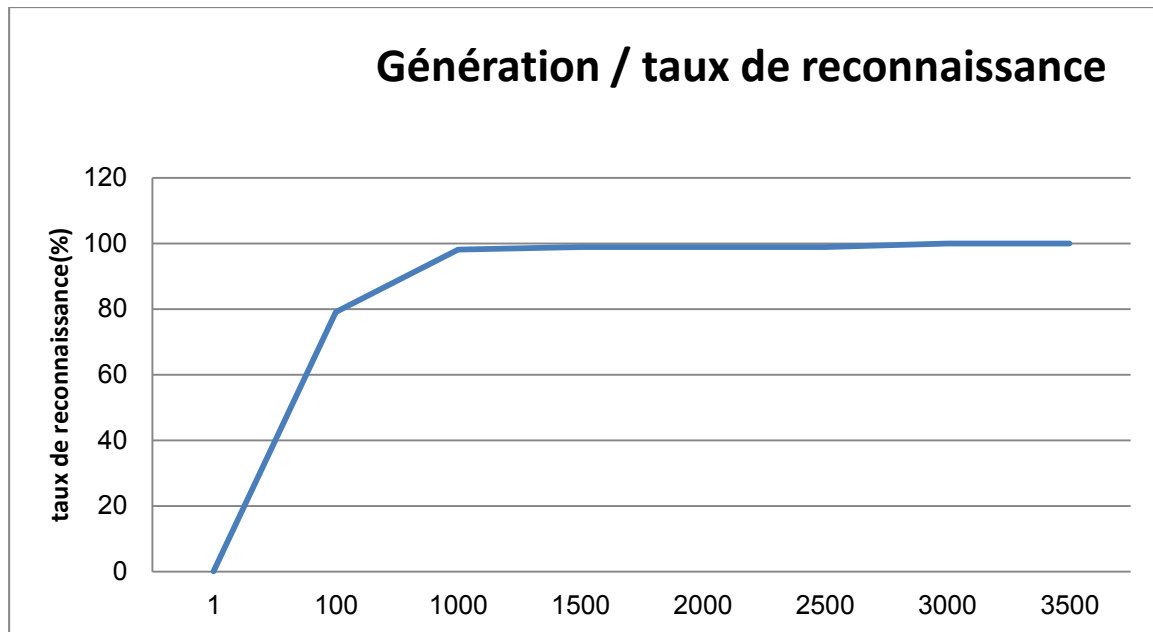


Figure 4.18: processus d'itération de l'AG pour l'optimisation simultanée (Génération / taux de reconnaissance).

Nous avons remarqué aussi que la taille de population n'a aucun impact ou bien changement remarquable sur le taux de reconnaissance (fitness).

Les résultats obtenus sont très encourageants et prometteurs et montrent que l'utilisation des méta-heuristiques peut être bénéfique à une meilleure reconnaissance des caractères arabes manuscrits par les SVM

4.6. Conclusion

Dans ce chapitre, nous avons présenté un système à base des SVM pour la reconnaissance des caractères arabes manuscrits, on se basant sur plusieurs traitements: binarisation, normalisation, extraction de caractéristiques et finalement la décision.

La classification des caractères arabes est un problème multi classes, bien que le SVM soit un classifieur binaire. Avec une méthode méta-heuristique pour la sélection des paramètres parfaits. De ce fait, nous avons combiné plusieurs SVM pour décider la classe d'appartenance de chaque caractère. Le système a été testé sur une base contenant 3920 images.

Les taux obtenus de nos jours en reconnaissance de l'écriture arabe manuscrite ne sont pas encore parfaitement suffisants, et la programmation qu'ils requièrent est nettement plus exigeante que celle que nous pouvions développer dans le cadre d'un projet scientifique, puisque c'est un domaine interdisciplinaire (la nécessité aux matériels, traitement d'image, la théorie d'information, méthode fiable d'apprentissage automatique,...).

Conclusion général

Le travail s'inscrit dans le cadre générale de la reconnaissance automatique d'écritures arabes manuscrit plus précisément, il exploite une approche basée sur les machines à vecteurs de support appliquée à la reconnaissance des caractères arabes manuscrits (isolée, au début, au milieu et à la fin). Le SVM représente un modèle dont la complexité est en fonction du nombre de vecteurs de support et des valeurs des hyper-paramètres, mais elle ne dépend pas du nombre de caractéristiques.

Dans notre système, nous avons opté deux stratégies *Un Contre Une* et *Une Contre Reste* et les fonctions noyaux utilisées et leurs paramètres sont déterminés de façon expérimentale en utilisant les algorithmes génériques, avec deux critères pour définir une relation d'ordre relative à la notion de taux de reconnaissance. Il est montré que ces deux critères sont plus performants si le critère d'arrêt de notre méthode générique est modifié pour forcer la sélection des paramètres, et l'autre pour augmenter le taux de reconnaissance (fitness). Nous montrons expérimentalement qu'il est possible de réduire la complexité des fonctions de décision produites tout en augmentant les capacités de généralisation.

L'absence d'une base de lettres arabes manuscrites qui est standard sur laquelle nous pouvons tester le système proposé, pour cela nous avons travaillé sur notre base locale composée de 3920 caractères.

Les résultats préliminaires obtenus pour le premier jeu de caractéristiques employé sont encourageants, car la qualité de la base de données est très réaliste avec des erreurs et des différences d'écriture d'un scripteur à un autre.

Conclusion général

Perspectives de recherche

La reconnaissance de l'écriture arabe manuscrite (hors ligne ou en ligne) est un axe de recherche récent et qui est prometteur pour des travaux futurs en vue d'augmenter le taux de reconnaissance, la fiabilité, et d'optimiser les performances de nos machines et l'intégrité de système de reconnaissance. Pour les améliorer, il faut chercher des nouvelles méthodes plus efficaces que les anciennes avec moins d'erreurs. La méthodes que nous avons proposées améliorent les performances de la méthode SVM, ce système est en premier stade d'expérimentation et l'expérience s'est avérée intéressante mais des extensions restent envisageables :

- Élargir la base de données en introduisant un plus grand nombre des scripteurs, pour inclure plus de variations dans les styles d'écritures, ainsi la généralisation pourra être plus performante.
- Étendre le système pour l'appliquer à la reconnaissance des mots arabes moyennant une approche analytique ou globale. Dans le cas de l'approche analytique, il faut procéder par la segmentation des mots en caractères et effectuer la reconnaissance par le système proposé.
- Utilisée d'autre méthode d'optimisation (colonie de fourmie, recuit simulé, ...)

Bibliographie

- [1] MEHENNAOUI Zahra," reconnaissance de l'écriture arabe manuscrite a base des machines a vecteurs de supports", diplôme de Magister, universite badji –mokhtar – Annaba, 2006.
- [2] ZAIZ Faouzi, Djefal Abdelhamid, Babahenini Mohamed Chaouki, "An Approach Based on Structural Segmentation for the Recognition of Arabic Handwriting", doi:10.4156/aiss.vol2. issue4.2, 2012.
- [3] DJEDDI Chawki, SOUCI-MESLATI Labiba," Identification de scripteurs pour l'écriture arabe par une approche locale", Laboratoire LRI, Université Badji Mokhtar, BP 12, 23000, Annaba, Algérie.
- [4] M.Abelhamid DJEFFAL, "Utilisation des méthodes Support Vector Machine (SVM) dans l'analyse des bases de données", Thèse de doctorat, Université Mohamed Khider – Biskra, 2012.
- [5] Hanene BOUKERMA," Combinaison de classifieurs flous pour la reconnaissance de l'écriture arabe manuscrite", diplôme de MAGISTER, UNIVERSITE 20 AOUT 1955 – SKIKDA, 09 décembre 2010.
- [6] N.D. Ayat, "Sélection Automatique de Modèle Dans Les Machines à Vecteurs de Support: Application à La Reconnaissance d'Image De Chiffres Manuscrits", Thèse de Doctorat, Université de Québec, 2004.
- [7] A. Belaïd : " Reconnaissance automatique de l'écriture et du document ". LORIA-CNRS, Campus scientifique B.P. 239, 54506 Vandoeuvre-Lès-nancy, France
- [8] M. ZAIZ Faouzi, "Les Supports Vecteurs Machines (SVM) pour la reconnaissance des caractères manuscrits arabes", diplôme de magister , Université Mohamed Khider – Biskra, Juillet, 2010.

Bibliographie

- [9] Guermeur, Y. and Paugam-Moisy, H. (1999). "Apprentissage Automatique, chapitre Théorie de l'apprentissage de Vapnik et SVM, Support Vector Machines, pages 109–138.
- [10] MUSTAPHA AMROUCH, " Reconnaissance de caractères imprimés et manuscrits, textes et documents basée sur les modèles de Markov cachés", thèse de doctorat es-sciences, UFR : Mathématiques et Informatique Appliquées, Université Ibn Zohr, 2012.
- [11] Lorigo L.M., Govindaraju V., "Offline Arabic handwriting recognition: a survey", IEEE Transactions on PAMI, Vol. 28, N°.5, pp. 712-724, May, 2006
- [12] N.ESSOUKRI Ben Amara, Utilisation des Modèles de Markov Cachés Planaires en Reconnaissance de l'écriture Arabe Imprimé, Thèse de Doctorat, Ecole National d'Ingénieur de Tunisie, Février 1999.
- [13] V.Govindaraju, R.K.Srihari, S.N. Srihari, "Handwritten Text Recognition", In Internal Association for Pattern Recognition Workshop on Document Analysis Systems (DAS'94), pp 157-171, Kaiserslautern, Germany, September 1994.
- [14] A.Bennasri, A.Zahour, B. Taconet : " Extraction des lignes d'un texte manuscrit arabe". Vision Interface '99, Trois-Rivières, Canada, 19-21 Mai 1999.
- [15] J. Callut : " Implementation efficace des Support Vector Machines pour la classification". Mémoire présentée en vue de l'obtention du grade de Maître en Informatique, Université Libre de Bruxelles, 2002-2003.
- [16] R. Rakotomalala. Arbres de décision. Revue Modulad, 33 :163–187, 2005.
- [17] R. Collobert and S. Bengio. Svmtorch : Support vector machines for large-scale regression problems. The Journal of Machine Learning Research, 1 :143–160, 2001.

Bibliographie

- [18] P. Mahé : " Noyaux pour graphes et Support Vector Machines pour le criblage virtuel de molecules ". Rapport de stage, DEA MVA 2002/2003,Septembre 2003
- [19] C. Cortes, V.Vapnik, "Support-vector networks", *Machine Learning*, 20(3): pp. 273-297, 1995.
- [20] Chris J.C. Burges, B. Schölkopf. "Improving the Accuracy and Speed of Support Vector Machines". In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, vol 9, pp 375,The MIT Press, 1997.
- [21] V. Blanz, B. Scholkopf, H.H. Bulthoff, C. Burges, V. Vapnik,T.Vetter. "Comparison of View-Based Object Recognition Algorithms Using Realistic 3d Models". In ICANN, pp.251–256, 1996.
- [22] E. Osuna, R. Freund, F. Girosi. "Training Support Vector Machines: an Application to Face Detection", 1997.
- [23] A. Smola, *Learning with Kernels*. PhD thesis, GMD First, Berlin, Germany, 1998.
- [24] G. Wahba. *Support vector machines, reproducing kernel hilbert spaces*, gacv, 1998.
- [25] MOHAMADALLY Hasan, FOMANI Boris, " SVM : Machines a Vecteurs de Support ou Separateurs a Vastes Marges",BD Web, ISTY3 Versailles St Quentin, France, 16 janvier 2006
- [26] N. Cristianini, J. Shawe-Taylor : "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods". Cambridge University Press, 2000.

Bibliographie

- [27] A. Bordes. Nouveaux Algorithmes pour l'Apprentissage de Machines à Vecteurs Supports sur de Grandes Masses de Données. PhD thesis, Université Paris VI, Pierre et Marie Curie, 2010.
- [28] L. Hamel. Knowledge discovery with support vector machines. Wiley Edition, 2009.
- [29] N.D. Ayat, M.Cheriet, C.Y.Suen, "Un Système neuro-flou pour la Reconnaissance de Montants Numériques de Chèques Arabes, In Colloque International Francophone sur l'écrit et le documents, pp 171-180, France, 2000.
- [30] A.Belaid, " Reconnaissance Automatique de l'Ecriture et du Document". Pour la science, 2001.
- [31] C.Bahlman, B.Haasdonk, H.Bukhardt, "On-line Handwriting Recognition with Support Vector Machines – A Kernel Approach", Proc. Of the 8th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR), pp. 49–54, 2002.
- [32] Hasan.Mohamadally et Boris.Fomani , « SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges » Versailles St Quentin, France 16 janvier 2006.
- [33] Gilles Lebrun, « Sélection de modèles pour la classification supervisée avec des SVM(Séparateurs à VasteMarge). Application en traitement et analyse d'images ». doctorat de l'universite de Caen/basse-Normandie, soutenue le 24 novembre 2006.
- [34] M. Pom H. Wu : " Handwritten Character Recognition ".Queensland University, October 2003.
- [35] Mahmoud ZENNAKI, Mamoun MAMOUNI, Kaddour SADOUNI, a comparative study of svm models for learning handwritten arabic characters, PO 1505 EL M'naoer, Bir el Djir, Oran , Algeria

Bibliographie

- [36] Muhammad Naeem ,Ayyaz, Imran Javed and Waqar Mahmood , Handwritten Character Recognition Using Multiclass SVM Classification with Hybrid Feature Extraction, Pak. J. Engg. & Appl. Sci. Vol. 10, Jan., 2012 (p. 57-67)
- [37] Sami Gazzah1 , Najoua Essoukri et Ben Amara, Une approche a priori pour l'identification du scripteur en reconnaissance optique de l'écriture arabe, Ecole Nationale d'Ingénieurs de Sousse, 2006
- [38] Sabri A. Mahmoud and Sameh M. Awaida, recognition of off-line handwritten arabic (indian) numerals using multi-scale features and support vector machines vs. hidden markov models, Information and Computer Science, King Fahd University of Petroleum and Minerals Dhahran, Saudi Arabia, Paper Received December 2, 2008; Paper Revised April 15, 2009; Paper Accepted May 27, 2009
- [39] Eric Anquetil, Reconnaissance d'écriture manuscrite et Interaction homme-document, Équipe IMADOC IRISA/INSA, 27 octobre 2008.
- [40] Dushyant Khurana, "Online Handwritten Gurmukhi Character Recognition using Support Vector Machine",school of mathematics and computer applications thapar university patiala – 147004, july 2013
- [41] Danelle Shah , Joe Schneider, "Learning Written Gesture Sequences for Robot Control",Final Project Report.
- [42] M. MAHOUSONZOU Adankon, "optimisation de ressources pour la sélection de modèle des svm", l'obtention de la maîtrise en génie de la production automatisée, école de technologie supérieure université du QUÉBEC, Montréal, le 16 septembre 2005.
- [43] X. Z. Li, and J. M. Kong, Application of GA–SVM method with parameter optimization for landslide development prediction, Nat. Hazards Earth Syst. Sci., 14, 525–533, 2014.
- [44] Li Zhuo , Jing Zheng Fang Wang, Xia Li, Bin Ai, Junping Qian, "a genetic algorithm based wrapper feature selection method for classification of hyperspectral images using

Bibliographie

- support vector machine", Commission VII, WG VII/3, School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275
- [45] Mohamed H. Ghaleb, Loay E. George, and Faisal G. Mohammed, "Printed and Handwritten Hindi/Arabic Numeral Recognition Using Centralized Moments", International Journal of Scientific & Engineering Research, Volume 5, Issue 3, March-2014 140 ISSN 2229-5518.
- [46] C. Chatelain, S. Adam, Y. Lecourtier, L. Heutte, Y. Oufella, T. Paquet, "Optimisation multi-objectif pour la sélection de modèles SVM", Laboratoire LITIS, Université de Rouen, Avenue de l'université, 76800 Saint Etienne du Rouvray, France, 2 juillet 2007.
- [47] Xin-She Yang, Suash Deb and Simon Fong, "Accelerated Particle Swarm Optimization and Support Vector Machine for Business Optimization and Applications", Communications in Computer and Information Science, Vol. 136, Springer, pp. 53-66 (2011).
- [48] M'hamed Billel Abidine Belkacem Fergani, "Evaluation de Méthodes à Noyaux pour la Sélection de Modèle SVM : Application au Traitement de Documents Audio", USTHB, B.P. 32, El Alia, Bab Ezzouar, Alger, ALGERIE.
- [49] G. Wahba, Y. Lin, and H. Zhang, "Generalized Approximate Cross Validation for Support Vector Machines, or, Another Way to Look at Margin-Like Quantities," Departement of Statistics, University of Wisconsin, Madison, Rapport Technique n° 1006, February 25 1999.
- [50] V. N. Vapnik, Statistical learning theory. New York: John Wiley and Sons, 1998.
- [51] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," in Advances in Large Margin Classifiers, A. J. Smala, P. Bartlett, B. Schoelkopf, and D. Schuurmans, Eds., 2000, pp. 61-74.

Bibliographie

- [52] N. A. Syed, H. Liu, and K. K. Sung, "Incremental Learning with Support Vector Machines," presented at International Joint Conference on Artificial Intelligence, 1999.
- [53] P. Mitra, C.A. Murthy, and S. K. Pal, "Data Condensation in Large Databases by Incremental Learning with Support Vector Machines," International Conference on Pattern Recognition, pp. 712-715, 2000.
- [54] F. GLOVER. " Tabu search: part I". Dans *on Computing*, 1(3), pages 190–206, 1989.
- [55] F. GLOVER. "Tabu search: part II" . Dans *on Computing*, 2(1), pages 4–32, 1989.
- [56] F. GLOVER & M. LAGUNA," Tabu search ", Kluwer Academic Publishers, Boston MA, 1997.
- [57] J.-K. HAO, P. GALINIER & M. HABIB,"Métaheuristiques pour l'optimisation combinatoire et l'affectation sous contraintes", Dans *Revue d'Intelligence Artificielle*, 13(2), pages 283–324, 1999.
- [58] J.k.Hao, P.Galinier, M.Habib," Méta-heuristiques pour l'optimisation combinatoire et l'affectation sous contraintes ", *Revue d'Intelligence Artificielle*, 13(2) : 283-324, 1999.
- [59] J.Haton," L'intelligence artificielle ", 263eme conférence de l'Université de tous les savoirs donnée, 2000.
- [60] Gilles Lebrun, "Sélection de modèles pour la classification supervisée avec des SVM(Séparateurs à VasteMarge). Application en traitement et analyse d'images", thèse Doctorat, université de caen/basse-normandie u.f.r.: sciences, école doctorale simem, le 24 novembre 2006.

Bibliographie

- [61] M.Kuri. "An Alternative Model of Genetic Algorithms as Learning Machines",Expert Systems with Applications, Volume 15, Issues 3-4, pp 351-356. 1998.
- [62] Hicham EL MOUBTAHIJ, Akram HALLI2 and Khalid SATORI, "review of feature extraction techniques for offline handwriting arabic text recognition", International Journal of Advances in Engineering & Technology, ISSN: 22311963, ,Mar. 2014.
- [63] Ginny Mak, "the implementation of support vector machines using the sequential minimal optimization algorithm", Master's Project, School of Computer Science McGill University, Montreal, Canada, 2000.
- [64] M. Asma BENDAHMENE, "une approche basée algorithme génétique pour l'apprentissage chez un agent",diplôme de Magister, Université Mohamed Khider Biskra, 2011.
- [65] Marino WIDMER, "les metaheuristiques : des outils performants pour les problemes industriels",3e Conférence Francophone de MOdélisation et SIMulation "Conception, Analyse et Gestion des Systèmes Industriels" , MOSIM'01 – du 25 au 27 avril 2001 - Troyes (France)
- [66] Sabri A. Mahmoud and Sameh M. Awaida, "recognition of off-line handwritten Arabic (indian) numerals using multi-scale features and support vector machines vs. hidden markov models",King Fahd University of Petroleum and Minerals Dhahran, Saudi Arabia, Paper Received December 2, 2008; Paper Revised April 15, 2009; Paper Accepted May 27, 2009
- [67] Jaakkola and D. Haussier, "Probabilistic kernel regression models," In D. Heckerman and J. Whittaker, editors, Workshop on Artificial Intelligence and Statistics 7. Morgan Kaufmann, 1999.

Bibliographie

- [68] Y. Yang, R. S. Chen, and Z. B. Ye, "Combination of particle-swarm optimization with least-squares support vector machine for FDTD time series forecasting," *Microw. Opt. Technol. Lett.*, vol. 48, pp. 141–144, 2006.
- [69] V. Cherkarsky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Netw.*, vol. 17, pp. 113–126, 2004.
- [70] BENYAHIA Souad, "Application de la classe des méthodes d'apprentissage statistique SVM(support vector machine) pour la reconnaissance des formes dans les images", diplôme de Magister, Université Abou Bakr Belkaid– Tlemcen, Juin 2011.
- [71] Claus Bahlmann, Bernard Haasdonk and Hans Burkhardt, "On-line Handwriting Recognition with Support Vector Machines- A Kernel Approach", University Freiburg, Germany, Workshop on Frontiers in Handwriting Recognition (IWFHR), pp. 49–54, 2002.
- [72] Rahul Agrawal, "Recognition of Online Handwritten Gurmukhi Strokes using Support Vector Machine", A Thesis Submitted in partial fulfillment of the requirements for the award of the degree of Master of Technology, University Patiala, Juin 2012.
- [73] Jin Chen, "Huaigu Cao and Rohit Prasad, Gabor Features for Offline Arabic Handwriting Recognition", DAS '10, June 9-11, 2010, Boston, MA, USA
- [74] Applied Media Analysis, Arabic-Handwritten-1.0, <http://appliedmediaanalysis.com/Datasets.htm>, 2013.
- [75] C. Chang and C. Lin. LIBSVM: a library for support vector machines. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, last access on January . 20th, 2015.
- [76] S. Kumar and C. Singh, "A Study of Zernike Moments and its use in Devnagari Handwritten Character Recognition", In Proc. International Conference on Cognition and Recognition, 2005, pp. 514-520.

Bibliographie

- [77] C. Burges, "A Tutorial on support Vector machines for pattern recognition", Data Mining and Knowledge Discovery, Vol. 2, pp.1-43, 1998.
- [78] EUN SEOG YOUNA," feature selection in support vector machines", a thesis presented to the graduate school, university of Florida, 2002.
- [79] [79] Yann Guermeur, "SVM Multiclasses, Théorie et Applications", mémoire pour l'obtention de l'habilitation à diriger des recherches, l'Université Nancy I, 28 novembre 2007.
- [80] J.-F. Scheid, "Introduction aux méthodes heuristiques", Graphes et RO TELECOM Nancy 2A, page 1-32, 2007
- [81] P.Pellerin," Méta-apprentissage des algorithmes génétiques ", thèse de doctorat, Université du Québec, Canada, 2005.
- [82] <http://svmlight.joachims.org/>
- [83] C.W. Hsu, C.J. Lin," A simple decomposition method for support vector machine", Mach. Learn. 46 (1-3) (2002) 219–314.
- [84] Youngsu Park, YoungKow Lee, Jong-Wook Kim and Sang Woo Kim, " Parameter optimization for SVM using dynamic encoding algorithm", June 2-5, KINTEX, Gyeonggi-Do, Korea.
- [85] J-C. Turlot, "Évaluation de la qualité d'une règle de décision et sélection de variables", in G. Celeux, J-P. Nakache, Eds, « Analyse discriminante sur variables qualitatives », Polytechnica, Chapitre 6, pages 147-180, 1994.
- [86] AMIRA BOUBAKEUR, HARROUZ AHMED AMINE , "Utilisation des métaheuristiques pour la résolution du problème de sélection d'attributs : Application à

Bibliographie

- la détection d'intrusions" ,diplôme de Magistère, École nationale supérieur d'informatique,2013
- [87] T. Back. Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, 1996.
- [88] Jason Weston, "Support Vector Machine (and Statistical Learning Theory) Tutorial", NEC Labs America, 4 Independence Way, Princeton, USA, jasonw@nec-labs.com.
- [89] YOUSSEF ES-SAADY, "Contribution au développement d'approches de reconnaissance automatique de caractères imprimés et manuscrits, de textes et de documents Amazighes", thèse de doctorat en sciences, Université Ibn Zohr , Faculté des Sciences d'Agadir, UFR : Mathématiques et Informatique Appliquées, Spécialité : Informatique, 28 janvier 2012.
- [90] Nello Cristianini," Support Vector and Kernel Machines", <http://www.support-vector.net/tutorial.html> (support de cours), 2001.
- [91] Shivali, "Recognition of online handwritten Punjabi numerals", A Thesis Submitted in partial fulfillment of the requirements for the award of the degree of Master of Science in Mathematics and Computing, University Patiala-147004(PUNJAB) INDIA, July 2011.
- [92] Thanh-Nghi Do, Jean-Daniel Fekete, François Poulet, " Algorithmes rapides de boosting de SVM", http://www.irisa.fr/texmex/people/poulet/index_fr.php .
- [93] Yichang Shih, Donglai Wei," Machine Learning Final Project: Handwritten Sanskrit Recognition using a Multi-class SVM with K -NN Guidance", Final Project.
- [94] BOUCHEKOUF Asma, " Perception du comportement de l'apprenant dans un environnement d'apprentissage", diplôme de MAGISTER en Informatique, université badji –mokhtar – Annaba, Faculté des Sciences de L'Ingéniorat, 2013.

Bibliographie

- [95] Abdallah BENOUARETH, Abdellatif ENNAJI, Mokhtar SELLAMI, "Reconnaissance des Mots Manuscrits Arabes par Combinaison d'une Approche Globale et une Approche Analytique", Laboratoire de Recherche en Informatique Institut d'Informatique- Université Badji Mokhtar - Annaba- BP, 12- 23000 Sidi Amar – Algérie,
- [96] Ludovic Mercier, "Les machines à vecteurs support pour la classification en imagerie hyperspectrale : implémentation et mise en œuvre ", CENTRE REGIONAL RHÔNE-ALPES, CENTRE D'ENSEIGNEMENT DE GRENOBLE, Travail d'Etude et de Synthèse Technique en INFORMATIQUE, 11 février 2010.
- [97] Jon Taylor, Siddarth Kumar, Irina Khaimovich, "Your Handwriting in Worse Than the Doctors!: Cursive Handwriting Segmentation and Character Recognition", 2007.
- [98] Ankita Kumar, Cristian Sminchisescu, "Support Kernel Machines for Object Recognition", University of Pennsylvania, TTI-Chicago.
- [99] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, " A Practical Guide to Support Vector Classification" , Department of Computer Science National Taiwan University, Taipei 106, Taiwan, <http://www.csie.ntu.edu.tw/~cjlin>, Initial version: 2003 Last updated: April 15, 2010.
- [100] "Chapitre14, Méthodes à noyaux", page de 435 à 487.
- [101] Mouhamadou Lamine SAMB, Fodé CAMARA, Samba NDIAYE, Yahya SLIMANI, Mohamed Amir ESSEGHIR, "Approche de sélection d'attributs pour la classification basée sur l'algorithme RFE-SVM, ARIMA Journal, vol. 17 (2014), pp. 197-219.
- [102] BENDJEDDOU Toufik, "Le Choix de paramètres pour la reconnaissance des chiffres manuscrits", Mémoire Présenté en Vue de l'Obtention Du Diplôme de Magistère, Option : Systèmes Intelligents, Université Badji Mokhtar Annaba, Faculté Des Sciences De L'ingénieur, Département D'électronique, 2007/2008.

Bibliographie

- [103] Sergio Di Martino, Filomena Ferrucci, Carmine Gravino, and Federica Sarro", A Genetic Algorithm to Configure Support Vector Machines for Predicting Fault-Prone Components ", University of Napoli "Federico II" Via Cinthia, 80126 Napoli, Italy, sergio.dimartino@unina.it, 2011.
- [104] Peter Bartlett, "Benchmarking Least Squares Support Vector Machine Classifiers ", 2004 Kluwer Academic Publishers, Manufactured in The Netherlands, 2004.
- [105] Marref Nadia, " Apprentissage Incrémental & Machines à Vecteurs Supports ", Présenté en vue de l'obtention du diplôme de Magister en Informatique, Option : Informatique Industrielle, Université HADJ LAKHDAR – BATNA, Faculté des sciences, Département d'Informatique, 18 /12 /2013.
- [106] Yuan Ren, Guangchen Bai, " Determination of Optimal SVM Parameters by Using GA/PSO ", JOURNAL OF COMPUTERS, VOL. 5, NO. 8, AUGUST 2010.
- [107] Stefan Lessmann, Robert Stahlbock, Sven F. Crone, "Genetic Algorithms for Support Vector Machine Model Selection ", 2006 International Joint Conference on Neural Networks, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, July 16-21, 2006.
- [108] O. Kadri, L. H. Mouss, M. D. Mouss, A. Abdelhadi, " Reconnaissance des Formes par SVM pour le Diagnostic du Système de Pasteurisation d'une Usine de Lait ", Revue des Sciences et de la Technologie –RST-, Volume 4 N°1 /janvier 2013.
- [109] Stefan Lessmann, Robert Stahlbock, Sven F. Crone, "Optimizing Hyperparameters of Support Vector Machines by Genetic Algorithms ", Inst. of Business Information Systems, University of Hamburg, D-20146 Hamburg, Germany,
- [110] Omari Mohammed, "Algorithmes Génétiques", Université d'Adrar, Support de cours, 2^{ème} Année Master en Informatique, Option : Réseaux et Systèmes Intelligents, omarinmt@gmail.com.

Bibliographie

- [111] Paulo Gaspar, Jaime Carbonell, and José Luis Oliveira, "On the parameter optimization of Support Vector Machines for binary classification", Journal of Integrative Bioinformatics, 9(3):201, 2012 <http://journal.imbio.de>.
- [112] NIDAL F. SHILBAYEH, MUSBAH M. AQEL, AND REMAH ALKHATEEB, "Recognition Offline Handwritten Hindi Digits Using Multilayer Perceptron Neural Networks", Department of Computer Science, AlZarqa University, Zarqa, Jordan, Aqelm06@yahoo.com.
- [113] Mohamed Ait Lhoussain, "Matrice de Gram", 5 décembre 2011.
- [114] Hassen Hamdi, Maher Khemakhem, " Distributing Arabic Handwriting Recognition System Based on the Combination of Grid MetaScheduling and P2P Technologies (Omnivore)", Universal Journal of Computer Science and Engineering Technology 1 (1), 31-35, Oct. 2010.
- [115] Sabri akram, Benali Moustafa, "Développement d'une application de traitement d'images", Mémoire de fin d'études Pour l'obtention du diplôme de Licence en Informatique, Université Abou Bakr Belkaid–Tlemcen, Faculté des Sciences, Département d'Informatique, 27 Juin 2013.
- [116] J EREMIE MARY, " Méthodes d'Apprentissage Avancées, SVM", équipe TAO LRI, 30 janvier 2006.
- [117] Arnaud Revel, "Support Vector Machines (Séparateurs à vaste marge)"(Support de cours), revel.arnaud@gmail.com.
- [118] Stéphane Canu, "Machine à noyaux pour l'apprentissage", Séminaire INRA, Laboratoire d'informatique, de Traitement de l'Information et des Systèmes, premon.nom@litislab.eu, 7 juillet, 2008.

Bibliographie

- [119] Kartar Singh Siddharth, Renu Dhir, Rajneesh Rani, "Handwritten Gurmukhi Numeral Recognition using Different Feature Sets", International Journal of Computer Applications (0975 – 8887), Volume 28– No.2, August 2011.
- [120] Jian-xiong Dong, Adam Krzyzak, Ching Y. Suen, " An improved handwritten Chinese character recognition system using support vector machine", Centre for Pattern Recognition and Machine Intelligence, 1455 de Maisonneuve Blvd. West, Suite GM-606, Montréal, Québec, Canada H3G 1M8.
- [121] Stéphane Canu, Xavier Mary et Alain Rakotomamonjy, " Construction de noyaux pour les séparateurs à vaste marge(SVM) Application aux noyaux d'ondelettes", INSA Rouen-Département ASI Laboratoire PSI, stephane.canu@insa_rouen.fr, asi.insa-rouen.fr/~scanu, p.1/27.
- [122] MAHDJANE Marima, "Détection d'anomalies sur des données biologiques par SVM", En vue de l'obtention du diplôme de Magister en électronique , Option télédétection, Université de Mouloud Mammeri de Tizi Ouzou, 14/10/2012.
- [123] PHAM Quang Dung, " RÉSEAUX DE NEURONES POUR LA RECONNAISSANCE DES FORMES", Travail d'Intérêt Personnel Encadré, Superviseur: Prof. HO Tuong Vinh, Hanoi, 15 juillet 2005.
- [124] Souquet Amédée Radet François-Gérard, " ALGORITHMES GENETIQUES", TE de fin d'année Tutorat de Mr Philippe Audebaud, 21/06/2004.
- [125] Gurpreet Singh, Chandan Jyoti Kumar, Rajneesh Rani, Dr. Renu Dhir, "Feature Extraction of Gurmukhi Script and Numerals: A Review of Offline Techniques", International Journal of Advanced Research in Computer Science and Software Engineering 3(1), Volume 3, Issue 1, January - 2013, pp. 257-263.