Université de Montréal

Apprentissage quantique

par
Sébastien Gambs

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures et postdoctorales
Philosophiæ Doctor (Ph.D.) en informatique

Juillet, 2008

Université de Montréal
Faculté des études supérieures et postdoctorales

Cette thèse intitulée :

Apprentissage quantique

présentée par :

Sébastien Gambs

a été évaluée par un jury composé des personnes suivantes :

Alain Tapp
président-rapporteur

Gilles Brassard
directeur de recherche

Esma Aïmeur
codirectrice

Pascal Vincent
membre du jury

Peter Høyer
examinateur externe

Khalid Benabdallah
représentant du doyen de la FES

RÉSUMÉ

L'informatique quantique s'intéresse aux implications de la mécanique quantique pour fins de traitement de l'information alors que l'apprentissage machine étudie les techniques permettant de donner à la machine la capacité d'apprendre à partir d'expériences passées. Utilisée en collaboration avec sa contrepartie classique, l'information quantique peut réaliser des prouesses hors de portée de l'information classique seule, comme factoriser efficacement des grands entiers, chercher dans un espace de recherche non-structuré avec un gain quadratique par rapport au meilleur algorithme classique possible ou encore permettre à deux personnes de communiquer de manière totalement confidentielle même sous les yeux d'un espion disposant d'une puissance de calcul illimitée.

Le but principal de cette thèse est de montrer que bien que l'informatique quantique et l'apprentissage machine peuvent sembler *a priori* très différents, ils ont déjà pu se rencontrer et interagir de multiples façons par le passé. L'apprentissage quantique est le domaine qui est né de l'intersection et des rencontres entre l'informatique quantique et l'apprentissage machine. La première contribution de cette thèse est d'introduire le domaine et de présenter un tour d'horizon des travaux antérieurs. Ensuite, deux nouvelles avancées de l'apprentissage quantique qui ont été réalisées dans le cadre de cette thèse seront détaillées.

La première avancée concerne la quantisation d'algorithmes d'apprentissage non-supervisé, où on remplace certaines parties d'algorithmes classiques par des sous-routines quantiques, afin d'obtenir une accélération du temps de calcul ou encore de sauver par rapport au coût de communication dans le cas d'une situation d'apprentissage distribué.

Enfin, la deuxième avancée consiste à définir l'analogue de l'apprentissage machine dans un monde où l'information et l'ensemble de données sont quantiques, ce qui influence directement le processus d'apprentissage et ses limites. Ainsi, en reformulant certains problèmes de la théorie de la détection quantique comme étant des tâches d'apprentissage, on peut amener des notions de l'apprentissage machine classique, telles que les réductions d'apprentissage, afin de résoudre ces problèmes.

Mots clés : apprentissage quantique, informatique quantique, apprentissage machine.

ABSTRACT

Quantum information processing is interested in the implications of quantum mechanics for information processing purposes whereas machine learning studies techniques to give to machines the ability to learn from past experiences. Classical and quantum information can be teamed together to realize wonders that are out of reach for classical information processing alone, such as being able to efficiently factorize large integers, search in an unstructured space with a quadratic speedup compared to the best classical algorithms and allow two people to communicate in perfect secrecy even under the nose of an eavesdropper having at his disposal unlimited computing power and technology.

The main purpose of this thesis is to show that although quantum information processing and machine learning may seem unrelated *a priori*, they have already met and interacted several times in the past. Quantum learning is the field that stems from the intersection and the encounters between quantum information processing and machine learning. The first contribution of this thesis is to introduce and give an overview of the field. Afterwards, two new advances of quantum learning that were realized during this PhD are detailed.

The first advance deals with the quantization of unsupervised learning algorithms, where some parts of classical learning algorithms are replaced by quantum subroutines, in order to provide a speed-up or save in the communication cost in the distributed setting.

Finally, the second advance concerns the definition of the analogue of machine learning in a world where the information and the dataset are quantum, which has a direct impact on the learning process and its limits. By rephrasing some problems of quantum detection and estimation theory as machine learning tasks, it is possible to bring in some notions of classical machine learning, such as learning reductions, to help solve these problems.

Keywords: Quantum Learning, Quantum Information Processing, Machine Learning.

TABLE DES MATIÈRES

RÉSUMÉ iii

ABSTRACT iv

TABLE DES MATIÈRES v

LISTE DES TABLEAUX ix

LISTE DES FIGURES x

LISTE DES SIGLES xii

NOTATION xiii

DÉDICACE xiv

REMERCIEMENTS xv

AVANT-PROPOS xvi

CHAPITRE 1 : INTRODUCTION 1

CHAPITRE 2 : SURVOL DE L'INFORMATIQUE QUANTIQUE 4

2.1 Présentation de l'informatique quantique 4

2.2 Notions de base d'informatique quantique 5

2.2.1 Qubit, transformation unitaire et mesure 5

2.2.2 Registre, circuit quantique et POVM 7

2.3 Propriétés de l'information quantique 10

2.3.1 Parallélisme quantique, intrication et interférence 10

2.3.2 Théorème de non-clonage, borne de Holevo et impossibilité d'apprendre de l'information sans perturbation 13

2.3.3 Codage superdense et téléportation quantique 15

2.3.4 Matrices densité 16

2.4 Quelques résultats quantiques 17

2.4.1 Algorithme de Grover 17
2.4.2 Algorithme de Shor 18
2.4.3 Cryptographie quantique 19

CHAPITRE 3 : SURVOL DE L'APPRENTISSAGE MACHINE 20

3.1 Présentation de l'apprentissage machine 20
3.2 Apprentissage supervisé 20
3.2.1 Classification, régression et ordonnancement 21
3.2.2 Généralisation et validation 23
3.2.3 Algorithmes d'apprentissage 24
3.3 Apprentissage non-supervisé 30
3.3.1 Catégorisation 31
3.3.2 Réduction de dimensionnalité 35
3.3.3 Estimation de densité 38
3.4 Conclusion 40

CHAPITRE 4 : TOUR D'HORIZON DE L'APPRENTISSAGE QUANTIQUE 42

4.1 Présentation de l'apprentissage quantique 42
4.2 Résultats fondamentaux en théorie de l'apprentissage 45
4.2.1 Apprentissage PAC 46
4.2.2 Apprentissage exact 48
4.3 Variantes quantiques d'algorithmes d'apprentissage 50
4.3.1 Réseaux de neurones quantiques 51
4.4 Algorithmes de catégorisation s'inspirant de la mécanique quantique . . . 53
4.5 Bornes en complexité de la communication quantique s'appuyant sur des notions d'apprentissage machine 54
4.6 Estimation de systèmes et de processus quantiques 55
4.7 Calcul bayésien pour les matrices densité 58
4.8 Cryptosystèmes basés sur des problèmes d'apprentissage difficiles 58
4.9 Apprendre à généraliser sur des mesures 59

CHAPITRE 5 : ALGORITHMES QUANTIQUES D'APPRENTISSAGE NON-SUPERVISÉ 61

5.1 Introduction 61
5.2 L'algorithme de Grover et ses variantes 63
5.3 Quantisation d'algorithmes de catégorisation 64
5.3.1 Modèle de boîte noire 65
5.3.2 Sous-routines quantiques 67
5.3.3 Construire explicitement l'oracle 71
5.4 Catégorisation par arbre couvrant minimal 79
5.5 Catégorisation divisive 82
5.6 k-médianes 84
5.6.1 Version standard 84
5.6.2 Version distribuée 87
5.7 Outils quantiques pour algorithmes d'apprentissage non-supervisé 93
5.7.1 Construction d'un graphe de voisinage 93
5.7.2 Détection d'anomalies 95
5.7.3 Initialisation des centres des catégories 97
5.8 Discussion et perspectives futures 99
5.8.1 Comparaison équitable entre algorithmes d'apprentissage quantique et classique et bornes inférieures 99
5.8.2 Quantisation d'Isomap 101
5.8.3 Algorithmes d'apprentissage basés sur le comptage 103
5.8.4 Autres directions de recherche 105

CHAPITRE 6 : APPRENTISSAGE MACHINE DANS UN MONDE QUANTIQUE 106

6.1 Apprendre dans un monde quantique 106
6.1.1 Apprendre avec un ensemble d'entraînement quantique 107
6.1.2 Classes d'apprentissage 108
6.1.3 Réductions entre tâches d'apprentissage 112
6.1.4 Fidélité, Control-Swap test et matrice de similarité 115
6.2 Classification quantique 118

6.2.1 Classification binaire 119
6.2.2 Classification binaire pondérée 127
6.2.3 Classification multiclasse 131
6.3 Apprentissage non-supervisé quantique 143
6.3.1 Catégorisation 143
6.3.2 Réduction de dimensionnalité quantique 146
6.3.3 Estimation de densité 148
6.4 Discussion et perspectives futures 151
6.4.1 Algorithmes quantiques de classification 152
6.4.2 Généralisation 152
6.4.3 Mise au point d'ensembles de données quantiques 153

CHAPITRE 7 : CONCLUSION 155
7.1 Première contribution : tour d'horizon de l'apprentissage quantique 155
7.2 Deuxième contribution : quantisation d'algorithmes d'apprentissage non-supervisé 156
7.3 Troisième contribution : apprentissage machine dans un monde quantique 157

BIBLIOGRAPHIE 159

LISTE DES TABLEAUX

5.1 Tableau résumant la taille et profondeur des différentes parties formant le circuit E qui fait l'encodage de l'ensemble de données. 77

5.2 Tableau résumant les bornes inférieures et supérieures connues pour les algorithmes d'apprentissage non-supervisé présentés dans le chapitre 5. . . 100

6.1 Tableau résumant les coûts d'entraînement et de classification des différentes tâches quantiques d'apprentissage présentées dans le chapitre 6. 151

LISTE DES FIGURES

1 Qui est l'apprenant ? xix

2.1 Exemple d'un simple circuit quantique. 6
2.2 Porte Swap. 8
2.3 Porte Control-U. 8
2.4 Une architecture générique de POVM. 9
2.5 Calcul d'une fonction par oracle quantique. Le symbole $\oplus$ représente l'opération qui fait le OU exclusif entre deux bits (ou deux chaînes de bits). 11
2.6 Calcul d'une fonction par retour de phase. 11
2.7 Circuit calculant f sur une superposition de toutes les entrées. 11
2.8 Circuit générateur d'états de Bell. 12
2.9 Représentation d'un circuit hypothétique permettant de réaliser un clonage parfait. 14

3.1 Illustration d'une tâche de classification binaire. 22
3.2 Illustration d'une tâche de catégorisation. 32
3.3 Exemple du rouleau suisse. 37
3.4 Illustration d'une tâche d'estimation de densité. 39

5.1 Calculer une fonction par retour de phase. 63
5.2 Illustration de l'oracle de distance. 66
5.3 Circuit réalisant le changement de phase dans la sous-routine quant_trouver_max. 68
5.4 Sous-circuit P réalisant le changement de phase. 69
5.5 Oracle calculant la somme des distances entre un point et tous les autres points 70
5.6 Forêt d'arbres binaires de FANOUT réalisant la copie des bits d'entrées. . 73
5.7 Circuit réalisant la fonction indicatrice $I\{i = j\}$. 74
5.8 Forêt d'arbres binaires de FANOUT réalisant la copie des fonctions indicatrices. 75

5.9 Circuit encodant la valeur d'un bit d'un attribut. 76
5.10 Circuit implémentant l'oracle de distance O. 78

6.1 Illustration du principe de réduction d'apprentissage. 113
6.2 Circuit réalisant le Control-SWAP test. 116
6.3 Illustration de l'oracle d'Helstrom. 124
6.4 Illustration d'un arbre de classification. 138

LISTE DES SIGLES

AdaBoost	Adaptive Boosting
ACP	Analyse en Composantes Principales
Bagging	Bootstrap Aggregating
EM	Expectation-Maximization
GHZ	Greenberger-Horne-Zeilinger
ID3	Iterative Dichotomiser 3
LLE	Local Linear Embedding
POVM	Positive-Operator Valued Measurement
PGM	Pretty-Good Measurement
PAC	Probably Approximately Correct
Qubit	Bit quantique

NOTATION

À l'exception de ceux qui incluent une référence dans leurs titres, tous les théorèmes, lemmes, corollaires, questions ouvertes et réductions présentés dans cette thèse sont issus de la recherche effectuée dans le cadre de mon doctorat. Par contre concernant les définitions, il m'arrive parfois d'omettre la référence spécifique lorsqu'il s'agit d'un concept standard ou que la définition a été adaptée au language et à la notation utilisée dans cette thèse. Je prie d'avance le lecteur de m'excuser et d'être indulgent à ce propos.

Les principales notations utilisés en informatique quantique et l'apprentissage quantique sont définies respectivement dans les chapitres 2 et 3. Il peut arriver cependant qu'une notation spécifique soit définie juste au moment où le besoin s'en fait sentir. Enfin, dernière remarque concernant la notation utilisée, le logarithme, dénoté par le terme log dans les formules mathématiques, est toujours en base 2 à moins que cela ne soit précisé autrement dans le texte.

Dédicace à un Ewok et mes filleuls, Maxence et Ness.

REMERCIEMENTS

Avant tout, je voudrais remercier mes deux directeurs de recherche, Gilles et Esma, pour m'avoir suivi et soutenu tout au long de ma thèse et pour toutes les discussions que nous avons eu ainsi que les idées échangées. De par leurs connaissances, et même leurs caractères, je pense qu'ils sont très complémentaires et que leur association forme une synergie exceptionnelle. J'ai beaucoup apprécié ma collaboration avec eux durant les années de mon doctorat et de ma maîtrise et j'ai énormément appris à leur contact. Je les remercie aussi de m'avoir laisser choisir librement mon sujet de thèse tout en me permettant de satisfaire ma curiosité intellectuelle en travaillant sur d'autres sujets.

J'aimerais aussi remercier tous mes amis et collègues étudiants du laboratoire Héron et du laboratoire d'informatique théorique et quantique (LITQ), dont particulièrement Anne, David, Éric, Frédéric, Guido, Hugue, Laurent, Michaël, Oliver, Paul et Som. Je tiens à remercier tout spécialement Anne, Frédéric et Michaël ainsi que Mehdi, Cédric et Florent pour avoir accepter de lire et commenter certaines sections de ma thèse. Merci aussi à Claude pour la découverte de la machine à expresso qui a été très appréciée durant la dernière ligne droite de la rédaction de cette thèse. Merci enfin encore à Michaël pour avoir tenu le rôle de fournisseur officiel de vitamines.

J'en profite pour remercier aussi ma famille, dont particulièrement mon père, pour m'avoir soutenu et encouragé tout au long de mes études. Je remercie aussi mes amis d'enfance de Strasbourg qui jouent à mes yeux le rôle de ma seconde famille depuis la mort de ma mère quand j'étais petit, soit Phil, Jeff, Fremlin, Hub, Lionel, Bouki, Aurélien, Louis, Yog, Giov, Mehdi, Olivier, Forti, Sylvain, Anne, Bleu, Antonin et Élodie. Merci aussi à mes amis de Montréal dont Florent, Cédric, Jasmine, Emmanuel, Clément, Lise, Violaine, Maryam, Meriem, Serge, Manu, ... Enfin merci aussi aux amis que j'ai connu à Montréal mais qui sont repartis depuis longtemps voguer vers d'autres horizons, soit Pierre, Corinne, Vincent, Benoît, Mathieu, Jordan et François.

Je m'excuse aussi d'avance pour toute personne que j'aurais oublié de mentionner explicitement dans ces remerciements.

AVANT-PROPOS

La plupart des travaux sur l'apprentissage quantique présentés dans cette thèse (dont principalement les chapitres 4, 5 et 6) se retrouvent dans des articles qui sont publiés ou actuellement en préparation [6–9, 92, 93]. Contrairement à certains domaines de recherche (comme l'apprentissage machine) où l'ordre des auteurs reflète généralement l'importance de leur contribution à un article, la convention en informatique quantique (ainsi que dans d'autres domaines tels que l'informatique théorique) est de faire figurer le nom des auteurs par ordre alphabétique. Je précise que j'ai contribué activement à tous les articles dont je suis co-auteur. En détails chapitre par chapitre, ces articles sont :

- Chapitre 4 : [93] Gambs, S. : *Quantum learning : a survey.* En préparation.
- Chapitre 5 : [9] Aïmeur, E., G. Brassard et S. Gambs : *Quantum clustering algorithms.* Dans Proceedings of the 24th Annual International Conference of Machine Learning (ICML'07), pages 1–8, 2007.

  [6] Aïmeur, E., G. Brassard et S. Gambs : *Quantum algorithms for unsupervised learning.* En préparation.
- Chapitre 6 : [8] Aïmeur, E., G. Brassard et S. Gambs : *Machine learning in a quantum world.* Dans Proceedings of the 19th Canadian Conference on Artificial Intelligence (Canadian AI'06), pages 433–444, 2006.

  [92] Gambs, S. : *Quantum classification.* En préparation.

  [7] Aïmeur, E., G. Brassard et S. Gambs : *Quantum learning tasks.* En préparation.

D'autres travaux que j'ai réalisé durant mon doctorat on fait l'objet de publications [10, 11, 44, 94] mais ne se retrouvent pas dans cette thèse car leurs thématiques ne sont pas directement reliés à l'apprentissage quantique. Les références de ces articles sont :

- [10] Aïmeur, E., G. Brassard, S. Gambs et B. Kégl : *Privacy-preserving boosting.* Dans Proceedings of Workshop on Privacy and Security Issues in Data Mining, in conjunction with the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'04), pages 51–69, 2004.
- [94] Gambs, S., B. Kégl et E. Aïmeur : *Privacy-preserving boosting.* Data Mining and Knowledge Discovery, 14(1) :131–170, 2007.
- [44] Brassard, G., A. Broadbent, J. Fitzsimons, S. Gambs et A. Tapp : *Ano-*

nymous quantum communication. Dans Proceedings of 13th Annual International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'07), pages 460–473, 2007. Accepté pour présentation at the 11th Workshop on Quantum Information Processing (QIP), New Delhi, décembre 2007.

– [11] Aïmeur, E. et S. Gambs : *Privacy-preserving data mining*. Encyclopedia of Data Warehousing and Mining (2nd edition), 2008. À paraître.

> "[...] making a shift from the observation that *Information is physical*, that has much influenced the development of Quantum Information Processing so far, to new one, namely that *Physics is informational*."

Josef Gruska, extrait du résumé d'un article intitulé "Quantumization of Informatics" qui a été publié dans les actes de l'atelier "Quantum Computing and Learning 99" source :`http://www.mdh.se/ima/personal/rbr01/courses/riga99proc/gruska.pdf`

> "Why unify information theory and machine learning ? Because they are two sides of the same coin. In the 1960s, a single field, cybernetics, was populated by information theorists, computer scientists, and neuroscientists, all studying common problems. Information theory and machine learning still belong together."

David J.C. MacKay, extrait de la préface de "Information Theory, Inference, and Learning Algorithms" (Cambridge University Press, 2003)

> "The core of the argument is that in modelling the universe through Machine Learning, we are obliged to make inferences based on finite and hence typically less-than-complete information. We can never know everything about a situation, and this gives us our link between quantum mechanics and statistical inference through machine learning."

David Lowe, extrait du résumé d'une présentation intitulée "Machine Learning, Uncertain Information, and the Inevitability of Negative Probabilities" (présentée à un atelier d'apprentissage machine, Sheffield 2004)

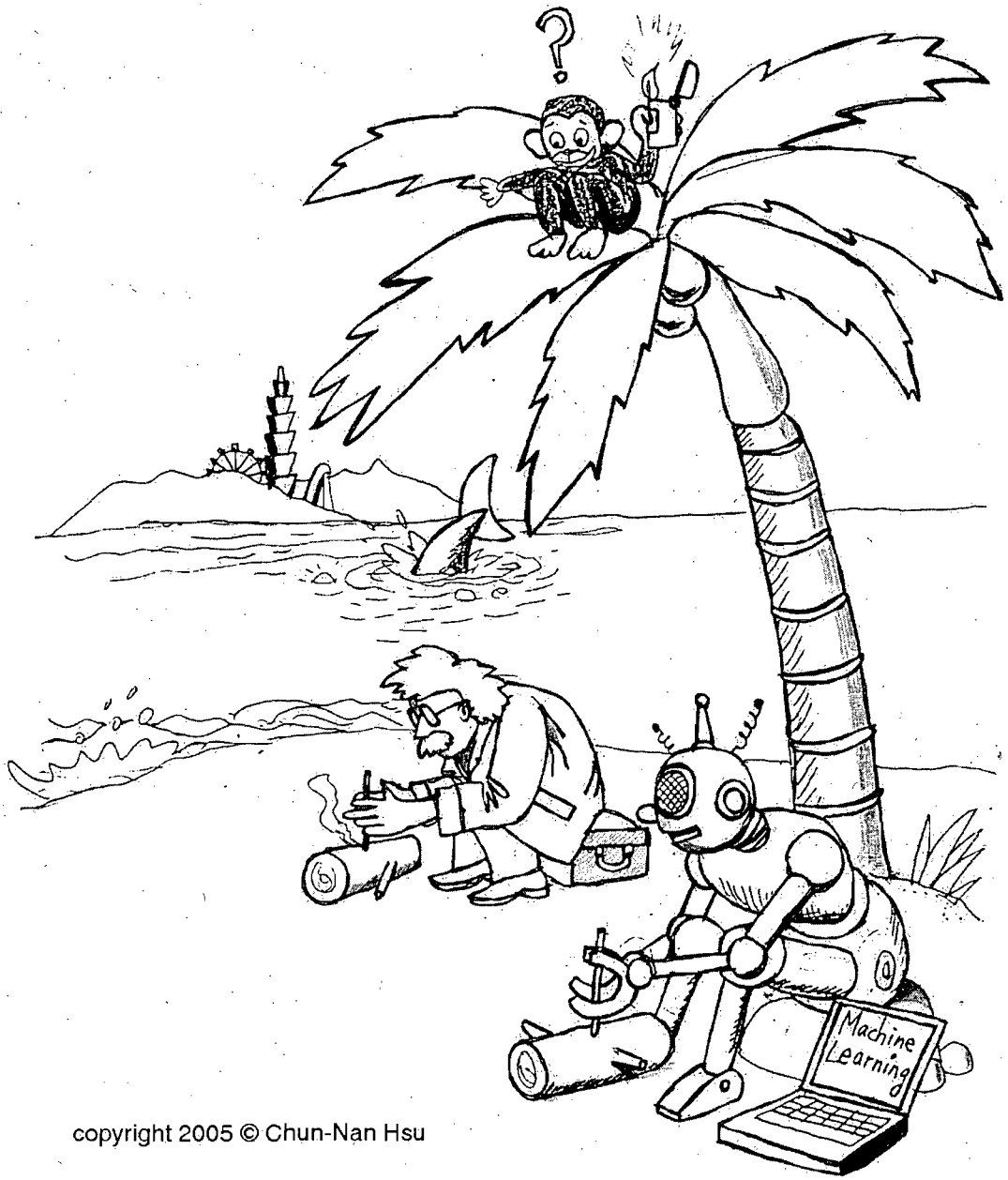

FIG. 1 – Qui est l'apprenant ?
(Image extraite de l'affiche d'une école d'été en apprentissage machine (Taipei, 2006) Source :`http://www.iis.sinica.edu.tw/MLSS2006/`).

CHAPITRE 1

INTRODUCTION

L'*informatique quantique* [126,152] s'intéresse aux *implications de la mécanique quantique pour fins de traitement de l'information* alors que l'*apprentissage machine* [79, 98, 114, 145, 182] étudie les *techniques permettant de donner à la machine la capacité d'apprendre à partir d'expériences passées.*

L'information quantique est intrinsèquement différente de sa contrepartie classique ; ainsi, elle ne peut pas être mesurée précisément et est perturbée par l'observation, mais peut en même temps exister en superposition d'états classiques. L'information classique et quantique peuvent être utilisées conjointement pour réaliser des prouesses hors de portée de l'informatique classique seule, comme par exemple factoriser efficacement des grands entiers [172], chercher dans un espace de recherche non-structuré avec un gain quadratique par rapport au meilleur algorithme classique possible [103] ou encore permettre à deux personnes de communiquer de manière totalement confidentielle même sous les yeux d'un espion disposant d'une puissance de calcul illimitée [23].

En apprentissage machine, les tâches typiques incluent la prédiction de la classe (*classification*) ou d'une caractéristique non observée (*régression*) d'un objet en se basant sur des observations sur celui-ci en *apprentissage supervisé*, ou encore la recherche de structures cachées à l'intérieur des données comme découvrir des catégories naturelles (*catégorisation*), trouver une représentation compacte qui préserve au mieux possible l'information (*réduction de dimensionnalité*) ou encore apprendre directement la fonction de probabilité décrivant la distribution de données (*estimation de densité*) en a*pprentissage non-supervisé.*

Pour n'importe quel domaine de l'informatique, il est naturel de se demander s'il est possible ou non, en utilisant le paradigme de l'information quantique d'obtenir des algorithmes plus efficaces, où la notion d'efficacité peut avoir des sens différents suivant le domaine considéré. Par exemple, cela pourrait signifier un algorithme plus rapide, ou encore économiser de la communication dans la réalisation de tâches distribuées, ou augmenter la sécurité dans un contexte cryptographique, etc. Quand on regarde spécifiquement l'apprentissage machine et l'informatique quantique, on peut imaginer plusieurs manières de

les faire interagir. Mon but principal dans cette thèse est de montrer que bien que l'informatique quantique et l'apprentissage machine peuvent sembler *a priori* très différents, ils peuvent (et ont déjà pu) se rencontrer de multiples façons dont ils ont mutuellement bénéficié. L'*apprentissage quantique* est le domaine qui est né de ces différentes rencontres entre l'informatique quantique et l'apprentissage machine.

Le plan de cette thèse est le suivant. Dans un premier temps, le chapitre 2 présente un survol de l'informatique quantique et introduit les notions de base du domaine, puis le chapitre 3 offre une présentation de l'apprentissage machine et de ses deux formes principales : l'*apprentissage supervisé* et l'*apprentissage non-supervisé*.

Ensuite, l'apprentissage quantique est défini dans le chapitre 4 et un tour d'horizon des rencontres antérieures entre l'informatique quantique et l'apprentissage machine est donné. Ces rencontres incluent :

- la comparaison de l'apprentissage quantique et classique en théorie calculatoire de l'apprentissage (appelée *computational learning theory* en anglais),
- les variantes quantiques d'algorithmes d'apprentissage dont les réseaux de neurones quantiques,
- les algorithmes classiques d'apprentissage s'inspirant de la mécanique quantique,
- les bornes en complexité de la communication quantique s'appuyant sur des notions d'apprentissage machine,
- l'estimation de systèmes et de processus quantiques,
- le calcul bayésien pour les matrices densité,
- les cryptosystèmes basés sur des problèmes d'apprentissage difficiles,
- apprendre à généraliser sur des mesures,
- ...

Le chapitre 5 se focalise sur un nouvel axe de recherche, la quantisation d'algorithmes d'apprentissage non-supervisé (dont principalement des algorithmes de catégorisation). Les principales contributions de ce chapitre sont :

- la description de sous-routines quantiques basées sur l'algorithme de Grover pouvant être utilisées pour quantiser des algorithmes d'apprentissage.
- une recette explicite montrant comment, à partir de la description classique d'un ensemble de données, construire un circuit quantique pouvant être utilisé ensuite comme une boîte noire à l'intérieur d'un algorithme d'apprentissage.

- des versions quantisées des algorithmes de catégorisation divisive, k-médianes (version standard et distribuée), construction d'un graphe de voisinage, détection d'anomalies et initialisation "intelligente" des centres des catégories. Toutes ces versions quantisées sont plus efficaces en terme de temps de calcul ou de complexité que leurs contreparties classiques.
- l'ouverture de perspectives futures comme une ébauche d'une version quantique d'un algorithme de réduction de dimensionnalité (Isomap) ou encore d'algorithmes quantiques d'estimation de densité basé sur des extensions de Grover permettant de compter le nombre de solutions.

Le chapitre 6 définit l'analogue de l'apprentissage machine quand l'ensemble de données est composé d'états quantiques, et non plus d'observations classiques sur des objets classiques. Ce changement de théorie sous-jacente à l'apprentissage a une influence directe sur le processus d'apprentissage et ses limitations. Les principales contribution de ce chapitre sont :

- le développement d'un cadre permettant de reformuler certains problèmes de la théorie quantique de la détection et de l'estimation en tâches d'apprentissage.
- la formalisation des notions de classes et de réductions d'apprentissage quantiques.
- la définition des variantes quantiques de la classification binaire, la classification binaire pondérée et la classification multiclasse.
- la description des réductions d'apprentissage permettant de réduire les versions pondérée et multiclasse de la classification à sa version standard.
- la définition des tâches d'apprentissage non-supervisé quantiques que sont la catégorisation, la réduction de dimensionnalité ou l'estimation de densité.

Finalement, la conclusion récapitule les principaux travaux d'apprentissage quantique décrits dans cette thèse et les mets en perspectives les uns par rapports aux autres.

CHAPITRE 2

SURVOL DE L'INFORMATIQUE QUANTIQUE

2.1 Présentation de l'informatique quantique

Définition 2.1 (Informatique quantique). *L'informatique quantique [126, 152] étudie les implications de la mécanique quantique pour fins de traitement de l'information.*

Lorsqu'on définit l'informatique comme « la science du traitement de l'information », on sous-entend implicitement que ce traitement de l'information s'effectue en suivant les lois de la physique classique qui décrivent les phénomènes à l'*échelle macroscopique*. La mécanique quantique, quant à elle, explique plutôt les phénomènes apparaissant à l'*échelle de l'atome* et le comportement des particules élémentaires. Les principes de la mécanique quantique sont très différents de ceux de la physique classique et peuvent parfois sembler contre-intuitifs au premier abord comme nous allons l'entrapercevoir dans les prochaines sections. En associant le traitement de l'information basé sur ces lois physiques avec l'informatique classique, il est possible de réaliser des prouesses hors de portée de l'informatique classique seule (cf. section 2.4).

L'informatique quantique est un domaine encore jeune qui est hautement multidisciplinaire car se trouvant au confluent de l'informatique, de la physique, des mathématiques et de la chimie :

- l'*informatique* pour la partie traitement de l'information,
- les *mathématiques* pour le formalisme servant à décrire le domaine,
- la *physique* pour les principes gouvernant le modèle de calcul et la réalisation physique de l'ordinateur,
- et la *chimie* pour certaines implémentations.

Historiquement, Feynman a été l'un des premiers à suggérer en 1981 qu'un ordinateur basé sur les principes de la mécanique quantique pourrait permettre de *simuler efficacement d'autres systèmes physiques quantiques* [85]. En 1985, Deutsch a ensuite développé l'idée qu'un tel ordinateur pourrait *offrir un avantage calculatoire comparé à un ordinateur classique* et a défini le modèle de la *machine de Turing quantique* [68], avant de définir en 1989 celui des *circuits quantiques* [69]. La preuve d'équivalence entre les deux

modèles et le raffinement du modèle des circuits quantiques a été apporté en 1993 par Yao [194]. Depuis, c'est principalement dans le langage des circuits que sont décrits la plupart des algorithmes quantiques.

2.2 Notions de base d'informatique quantique

La section suivante présente les principes et notions de base des circuits quantiques, qui est le modèle de calcul généralement utilisé pour décrire le fonctionnement des algorithmes quantiques.

2.2.1 Qubit, transformation unitaire et mesure

Un *qubit* (ou *bit quantique*) est l'analogue quantique d'un bit classique. Par contraste avec sa contrepartie classique qui peut seulement être dans un état à la fois (soit zéro, soit un), le qubit peut exister en une *superposition* d'états. Ainsi tout système physique qui peut exister en superposition cohérente d'états est un bit quantique potentiel.

Exemple 2.1 (Système quantique). *Un atome pouvant se retrouver dans l'état fondamental ou excité, la polarisation d'un photon (horizontale, verticale ou circulaire) ou encore le spin d'un électron qui peut pointer vers le haut ou le bas sont tous des implémentations physiques potentielles de* systèmes quantiques.

Un des prérequis pour les utiliser pleinement comme système quantique, est de pouvoir les mettre en superposition de plusieurs états, et non un seul état comme le serait un bit classique.

Définition 2.2 (Qubit). *Formellement, en utilisant la notation de Dirac, un* qubit *peut être décrit comme* $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, *où* α *et* β *sont des nombres complexes appelés les* amplitudes *des états* $|0\rangle$ *et* $|1\rangle$, *respectivement. Les valeurs de* α *and* β *sont telles que* $|\alpha|^2 + |\beta|^2 = 1$. *De manière équivalente,* $|\psi\rangle$ *peut être décrit comme un* vecteur colonne *représentant un état quantique vivant dans un espace de Hilbert.*

Lorsqu'un qubit est mesuré, on va observer un des deux états classiques avec un probabilité qui dépend de son amplitude mise au carré.

Définition 2.3 (Effet de la mesure). *Lorsque l'état* $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ *est* mesuré, *soit* $|0\rangle$ *va être observé avec probabilité* $|\alpha|^2$, *soit* $|1\rangle$ *sera observé avec probabilité* $|\beta|^2$.

En fait, la mesure est le seul acte *irréversible* car une fois celle-ci effectuée le qubit *sera devenu l'état mesuré*. Le reste du temps, les calculs effectués sont réversibles et sont réalisés à l'aide de transformations *unitaires*, qui correspondent aux opérations autorisées par les lois de la mécanique quantique.

Définition 2.4 (Transformation unitaire). *Une* transformation U *sera dite* unitaire *si et seulement si* $\mathsf{U}\mathsf{U}^\dagger = \mathsf{U}^\dagger\mathsf{U} = \mathsf{I}$, *où* $\mathsf{U}^\dagger$ *est la transposée conjuguée de* U *et* I *la matrice identité. Une transformation unitaire agissant sur d qubits est généralement représentée sous la forme d'une* matrice carrée *de taille* 2^d *par* 2^d, *où les entrées de la matrice correspondent à des nombres complexes.*

Définition 2.5 (Porte unaire : Walsh–Hadamard). *Un exemple d'une telle opération qui ne possède pas de contrepartie classique est la porte de* Walsh–Hadamard H *qui transforme l'état* $|0\rangle$ *en* $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ *et l'état* $|1\rangle$ *en* $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$.

De façon équivalente, on peut voir l'action de H comme $|a\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle + (-1)^a\frac{1}{\sqrt{2}}|1\rangle$, où a est un bit classique. Si on commence avec un qubit arbitraire $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ et qu'on lui applique H, on obtient par *linéarité* $\alpha(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) + \beta(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle) = \frac{\alpha+\beta}{\sqrt{2}}|0\rangle + \frac{\alpha-\beta}{\sqrt{2}}|1\rangle$.

Considérons l'exemple suivant qui illustre les notions vues pour l'instant[1] :

FIG. 2.1 – Exemple d'un simple circuit quantique.

Si on initialise $|\psi\rangle = |0\rangle$, cela signifie que l'on va commencer avec $|0\rangle$, appliquer la porte de Walsh–Hadamard et mesurer l'état. Après la porte de Walsh–Hadamard, l'état $|\psi\rangle$ va avoir évolué vers $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ et après la mesure, on va observer soit le bit classique 0 avec probabilité $\frac{1}{2}$, soit 1 avec la même probabilité. On peut remarquer d'ailleurs qu'un tel circuit correspond à un *générateur parfait de bit aléatoire*. Il est particulièrement

[1] Dans la notation des circuits utilisée dans cette thèse, un trait simple représente un fil transportant un qubit alors qu'un trait double est utilisé pour un bit classique. De même, un bloc rectangulaire représente une transformation unitaire agissant sur un (ou des) qubit(s) alors qu'un bloc qui est arrondi à droite symbolise une mesure qui prend en entrée un (ou des) qubit(s) et produit en sortie de l'information classique.

important de comprendre qu'après la mesure, $|\psi\rangle$ ne sera plus en superposition mais qu'il *sera devenu* la valeur observée (soit $|0\rangle$ ou $|1\rangle$), d'où l'aspect irréversible de la mesure.

Les autres transformations unitaires opérant sur un qubit rencontrées fréquemment sont le Not, le changement de phase P et R_θ, la rotation par un angle θ.

Définition 2.6 (Portes unaires : Not, P et R_θ). *L'effet de la porte quantique* Not *est exactement le même que celui de la version classique, c'est à dire de transformer* $|0\rangle \mapsto |1\rangle$ *et* $|1\rangle \mapsto |0\rangle$. *Le changement de phase* P, *par contre, ne connaît pas d'équivalent classique, il amène* $|0\rangle \mapsto |0\rangle$ *et* $|1\rangle \mapsto -|1\rangle$. *La rotation par un angle arbitraire* R_θ *transporte* $|0\rangle \mapsto \cos\theta\,|0\rangle - \sin\theta\,|1\rangle$ *et* $|1\rangle \mapsto \sin\theta\,|0\rangle + \cos\theta\,|1\rangle$.

On peut aussi remarquer que les portes H, Not et P sont leurs propres inverses, c'est à dire que $\mathsf{U} = \mathsf{U}^\dagger$, où U est la transformation unitaire considérée. Ainsi dans l'exemple de la figure 2.1, si on avait appliqué une seconde fois la porte H plutôt que de mesurer directement, on serait de nouveau revenu sur $|\psi\rangle = \mathsf{HH}\,|0\rangle = |0\rangle$.

2.2.2 Registre, circuit quantique et POVM

La notion de qubit possède une extension naturelle qui est celle du *registre quantique.*

Définition 2.7 (Registre quantique). *Un* registre quantique $|\psi\rangle$ *composé de n qubits vit dans un espace de Hilbert* $\mathcal{H}$ *de dimension* 2^n. *Le registre quantique* $|\psi\rangle = \sum_{i=0}^{2^n-1} a_i\,|i\rangle$ *est décrit par* $a_0, \ldots, a_{2^n-1}$ *qui sont des nombres complexes tels que* $\sum_{i=0}^{2^n-1} |a_i|^2 = 1$. *L'état* $|i\rangle$ *représente l'encodage binaire de l'entier* i.

Le *produit tensoriel* $\otimes$ est généralement utilisé pour représenter la *composition* de deux systèmes quantiques (comme par exemple de deux qubits).

Exemple 2.2 (Composition). *Si on dispose de deux systèmes* $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ *et* $|\phi\rangle = \gamma\,|0\rangle + \delta\,|1\rangle$ *et qu'on les place l'un à côté de l'autre, on peut décrire le* système composite *comme* $|\Gamma\rangle = |\psi\rangle \otimes |\phi\rangle = \alpha\gamma\,|00\rangle + \alpha\delta\,|01\rangle + \beta\gamma\,|10\rangle + \beta\delta\,|11\rangle$.

La notation $|\psi\rangle^{\otimes k}$ est utilisée comme raccourci pour représenter un registre quantique composé de k copies du même état $|\psi\rangle$.

Les opérations unitaires peuvent aussi être généralisées à deux ou plusieurs qubits.

Définition 2.8 (Porte binaire : Swap). *L'action de la* porte binaire Swap *sur deux qubits* $|\psi\rangle$ *et* $|\phi\rangle$ *passés en entrée (figure 2.2) est d'échanger la place des qubits sur les fils du circuit.*

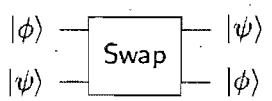

FIG. 2.2 – Porte Swap.

Définition 2.9 (Porte binaire : Control-U). *La porte Control-*U *est une* porte binaire, *où* U *peut être n'importe quelle opération unitaire sur un qubit. Quand cette porte est appliquée, le qubit situé sur le fil supérieur est responsable de décider si oui ou non la porte* U *sera appliquée sur le qubit inférieur. On appelle le qubit supérieur la* source *et le qubit inférieur la* cible.

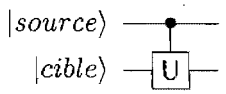

FIG. 2.3 – Porte Control-U.

Exemple 2.3 (Porte binaire : Control-NOT). *La porte Control-*NOT *est une porte binaire, où on applique la porte* NOT *sur la cible uniquement si la valeur de la source est* $|1\rangle$.

Si jamais la source et la cible sont en superpositions d'états, il peut arriver que non seulement la cible change de valeur, mais aussi que la source soit affectée. Ainsi dans le cas de la porte Control-NOT, si $|source\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ et $|cible\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$, nous avons comme entrée au circuit

$$|source\rangle \otimes |cible\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \tag{2.1}$$

ce qui équivaut à

$$\frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle . \tag{2.2}$$

Or, après l'application de la porte Control-NOT, nous avons comme sortie

$$\frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|11\rangle - \frac{1}{2}|10\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right). \quad (2.3)$$

Ainsi, la cible est inchangée mais l'état de la source est maintenant de $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ au lieu de $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ initialement.

En informatique quantique, tout ce qui correspond à une opération physique valide effectuée sur un système quantique peut être décrit par une mesure généralisée appelée POVM [156] (du terme anglais *Positive-Operator Valued Measurement*).

Théorème 2.1 (Implémentation d'un POVM, théorème de Neumark (voir [156] par exemple)). *Il est toujours possible de représenter l'action d'un POVM sur un état quantique $|\psi\rangle$ dans le language des circuits par l'ajout d'un système ancillaire $|0\rangle^{\otimes k}$ (dont la dimension 2^k dépend du POVM spécifique qu'on souhaite réaliser), l'application d'une opération unitaire* U*, opérant sur les n qubits composant $|\psi\rangle$ plus les k qubits du système ancillaire, suivi d'une mesure à la fin.*

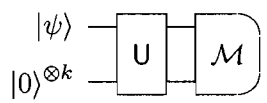

FIG. 2.4 – Une architecture générique de POVM.

En pratique, cependant, décomposer l'opération unitaire U, qui peut-être décrite par une matrice carrée de taille 2^{n+k} par 2^{n+k} où les entrées sont des nombres complexes, sous forme de portes unaires et binaires est souvent une tâche non-triviale. De plus, afin d'être considéré efficace, la taille d'un circuit quantique doit être polynomiale par rapport au nombre de qubits composant l'entrée $|\psi\rangle$. Tout l'art de développer un algorithme quantique pour une tâche précise est de trouver un circuit qui est plus efficace que ce qui est classiquement possible/connu.

Lemme 2.1 (Principe du délai de la mesure (voir page 186 dans [152])). *Le* principe du délai de la mesure *énonce que même si une séquence d'opérations sur un registre quantique inclut des mesures en cours de route et un comportement adaptatif en fonction des résultats des mesures, il est toujours possible de reformuler cette séquence d'opérations*

en un POVM qui ne comporte qu'une seule mesure à la fin et dont tout le reste de l'évolution est unitaire (et donc représentable sous forme d'un circuit quantique).

La porte Control-Not (ou n'importe quelle porte binaire autre que Swap et l'identité I), ainsi que pratiquement n'importe quelle porte unaire qui ne préserve pas la base de calcul, sont suffisants pour obtenir l'*universalité* du modèle [171], c'est-à-dire pouvoir calculer n'importe quelle fonction. Par opposition, en informatique classique, si on requiert que le calcul soit universel *et* réversible [20], alors il faut utiliser des portes faisant interagir *au moins 3 bits en même temps* tel que la porte de Toffoli [180] ou la porte de Fredkin [88].

2.3 Propriétés de l'information quantique

Comme nous allons le voir dans cette section, l'information quantique est de par sa nature très différente de l'information classique et elle possède des propriétés très particulières.

2.3.1 Parallélisme quantique, intrication et interférence

L'informatique quantique puise sa force dans trois ressources n'ayant pas de contre-partie classique : le *parallélisme quantique*, l'*intrication* et l'*interférence*.

Définition 2.10 (Parallélisme quantique). *Le* parallélisme quantique *réfère au fait que de par le* principe de superposition *et la* linéarité *de la mécanique quantique, il est possible de calculer en parallèle les valeurs d'une fonction sur plusieurs entrées.*

Ainsi, pour une fonction booléenne qui prendrait en entrée une chaîne de n bits, il est possible de calculer en une seule fois, le résultat de la fonction sur une superposition des 2^n entrées possibles. Comparé au cas classique, où on ne peut calculer la fonction qu'avec une seule entrée à la fois, on a donc un gain potentiel qui est exponentiel. Cependant, si on cherche naïvement à observer directement le résultat de la fonction, on va seulement observer une seule sortie de la fonction parmi le nombre exponentiel de sorties calculées. Il est malgré tout possible de tirer avantage des valeurs calculées sans chercher à les mesurer directement comme le font les algorithmes quantiques (voir section 2.4).

Quantiquement, un circuit quantique (figure 2.5) calculant une fonction f peut être représenté par une transformation unitaire U qui prend en entrée $|x\rangle$ et un qubit ancillaire

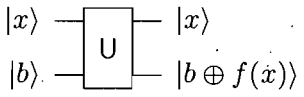

FIG. 2.5 – Calcul d'une fonction par oracle quantique. Le symbole $\oplus$ représente l'opération qui fait le OU exclusif entre deux bits (ou deux chaînes de bits).

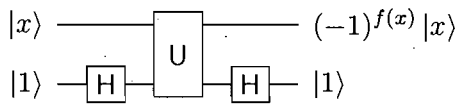

FIG. 2.6 – Calcul d'une fonction par retour de phase.

$|b\rangle$, et qui retourne en sortie $|x, b \oplus f(x)\rangle$ (ce qui implique que le circuit est réversible). Ce même circuit peut être utilisé (figure 2.6) pour stocker la valeur de $f(x)$ directement dans la phase en utilisant la technique du *retour de phase* [60]. Soit le circuit représenté dans la figure 2.7 qui illustre comment on peut interroger une transformation unitaire U réalisant une fonction booléenne $f : \{0,1\}^n \rightarrow \{0,1\}$ en superposition de toutes les entrées possibles. Le circuit prend en entrée $n+1$ qubits, les n premiers qubits servant à représenter les chaînes de bits $x \in \{0,1\}^n$ passées en entrée à la fonction f et le qubit ancillaire servant à recueillir le résultat de l'application de cette fonction. La tour de Walsh–Hadamard $\mathsf{H}^{\otimes n}$ va appliquer H sur tous les qubits du registre quantique (sauf le qubit ancillaire). Elle génèrera une superposition uniforme de toutes les entrées possibles $\frac{1}{\sqrt{n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |0\rangle$. Après l'application de U, le résultat de la fonction sera stocké dans le qubit ancillaire et le registre quantique sera dans une superposition uniforme des entrées et des sorties de la fonction f, soit $\frac{1}{\sqrt{n}} \sum_{x=0}^{2^n-1} |x\rangle \otimes |f(x)\rangle$. En un seul appel de U, on a donc évalué la fonction f sur 2^n entrées en parallèle. Cependant, si on observe directement le registre sans faire aucun post-traitement, on verra simplement une paire $(x, f(x))$ choisie aléatoirement parmi toutes les paires d'entrée/sortie possibles.

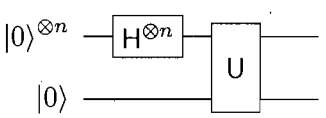

FIG. 2.7 – Circuit calculant f sur une superposition de toutes les entrées.

Il existe des états quantiques qu'on dit *intriqués* car ils ne peuvent pas être décrits de façon indépendante. L'état $|\Psi^-\rangle = \frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle$, introduit pour la première fois en 1935 par Einstein, Podolsky et Rosen dans un papier essayant de prouver en vain que la mécanique quantique est incomplète [81], est un de ces états.

Définition 2.11 (États séparables et intriqués). *Les états quantiques qui peuvent être exprimés comme le produit tensoriel d'au moins deux systèmes sont dits* séparables, *alors que tous les autres sont considérés comme étant* intriqués.

Bell a prouvé en 1964 que les corrélations qu'offrent certains états intriqués sont impossibles à reproduire classiquement, même en utilisant une théorie à variables locales cachées [19].

Définition 2.12 (Systèmes bipartites intriqués : états de Bell). *Les états* $|\Psi^-\rangle$, $|\Psi^+\rangle = \frac{1}{\sqrt{2}}|01\rangle + \frac{1}{\sqrt{2}}|10\rangle$, $|\Phi^-\rangle = \frac{1}{\sqrt{2}}|00\rangle - \frac{1}{\sqrt{2}}|11\rangle$ *et* $|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ *sont appelés* états de Bell.

Si on mesure la moitié d'un des états de Bell, le résultat de la mesure sur cette moitié va parfaitement déterminer le résultat de la mesure sur l'autre moitié. Cependant, pour une personne qui n'aurait aucune information sur le résultat de cette mesure, le comportement qu'elle pourra observer de la part de l'autre moitié de la paire sera exactement identique avant ou après la mesure. Cela a pour conséquence que l'intrication ne peut pas être utilisée comme moyen de communication instantanée. Par contre, de par les corrélations qu'elle offre qui ne peuvent pas être reproduites même avec des variables aléatoires partagées, l'intrication peut être utilisée comme une ressource, par exemple pour diminuer le coût de communication de certaines tâches distribuées [43, 189].

Soit le circuit suivant :

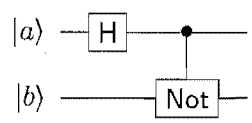

FIG. 2.8 – Circuit générateur d'états de Bell.

L'entrée du circuit (figure 2.8) est composée de l'état $|a\rangle \otimes |b\rangle$ où a et b sont deux bits classiques. Après l'application de la porte de Walsh–Hadamard H, le système se trouvera

dans l'état $(\frac{1}{\sqrt{2}}|0\rangle + (-1)^a \frac{1}{\sqrt{2}}|1\rangle) \otimes |b\rangle$ ce qui est équivalent à $\frac{1}{\sqrt{2}}|0b\rangle + (-1)^a \frac{1}{\sqrt{2}}|1b\rangle$. À la sortie du circuit, on obtiendra après l'action du Control-Not, l'état $\frac{1}{\sqrt{2}}|0b\rangle + (-1)^a \frac{1}{\sqrt{2}}|1\bar{b}\rangle$. En faisant varier les deux bits classiques a et b mis à l'entrée du circuit, on peut générer les quatre états de Bell. Plus précisément, si $a = 0$ alors on obtient « $^+$ » sinon « $^-$ », et de même si $b = 0$ alors $|\Phi\rangle$ sinon $|\Psi\rangle$. Par exemple si $ab = 11$, on aura généré $|\Psi^-\rangle$. L'intrication peut se généraliser à des états sur n qubits, comme par exemple l'état GHZ généralisé à n participants (aussi appelé *état chat*) qui a la forme $\frac{1}{\sqrt{2}}|0^n\rangle + \frac{1}{\sqrt{2}}|1^n\rangle$.

Définition 2.13 (Interférence). *On nomme* interférence, *le phénomène quantique qui fait que les chemins logiques de calcul puissent interférer entre eux de manière* constructive *ou* destructive.

L'amplitude d'un chemin particulier est calculée en additionnant les amplitudes des branches qui le composent. Comme les amplitudes sont des nombres complexes, l'amplitude d'un chemin de calcul sera elle aussi un nombre complexe. Lorsqu'on fait la somme des amplitudes des chemins menant à un état particulier, ces valeurs peuvent se consolider entre elles, si par exemple toutes les amplitudes ont des valeurs réelles positives (ou encore si elles ont toutes des valeurs réelles négatives). Ces valeurs peuvent aussi s'annuler entre elles si par exemple toutes les amplitudes des chemins ont la même norme mais qu'exactement la moitié sont des nombres positifs et que l'autre moitié soient des nombres négatifs. L'interférence est dite *constructive* si les chemins se consolident entre eux et *destructive* lorsqu'ils s'annulent entre eux. Lorsque de l'interférence constructive a lieu lors d'un calcul, elle s'accompagne aussi forcément de sa contrepartie destructive. La plupart des algorithmes quantiques utilisent le phénomène d'interférence de manière "intelligente" afin de faire ressortir le résultat recherché lors de la mesure finale.

2.3.2 Théorème de non-clonage, borne de Holevo et impossibilité d'apprendre de l'information sans perturbation

Comme nous avons pu l'apercevoir dans la section 2.2.1, un qubit est perturbé par l'acte de la mesure. Trois théorèmes fondamentaux en théorie de l'information quantique posent directement des limites sur ce qui peut être fait/ce qu'on peut apprendre d'un état inconnu $|\psi\rangle$. Le premier théorème, appelé *théorème de non-clonage* [192], est dû à Wootters et Żurek, et indépendamment Dieks [72], et peut être énoncé de la façon

suivante.

Théorème 2.2 (Théorème de non-clonage [192]). *Il est impossible à partir d'un état inconnu $|\psi\rangle$ d'en produire une copie additionnelle parfaite à moins que l'état $|\psi\rangle$ n'appartienne à un ensemble connu d'états orthogonaux*[2].

Démonstration. Le preuve de ce théorème est relativement simple. Si un appareil parfait de clonage existait alors son action pourrait être représentée par le circuit suivant :

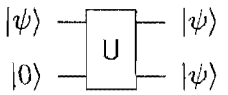

FIG. 2.9 – Représentation d'un circuit hypothétique permettant de réaliser un clonage parfait.

Pour que le circuit U réalise effectivement le clonage, il faut en particulier qu'il envoie $|0\rangle|0\rangle \mapsto |0\rangle|0\rangle$ et $|1\rangle|0\rangle \mapsto |1\rangle|1\rangle$. Or par linéarité, cela veut dire qu'il va transformer $(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|0\rangle$ vers $\frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|1\rangle$ ce qui est différent de $(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle) \otimes (\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle)$. Un tel circuit de clonage universel ne peut donc pas exister. Cette preuve n'est cependant pas complète car elle ne tient pas compte de la possibilité de faire une mesure en cours de route ce qui rendrait le circuit hautement non-unitaire. Voir [118] pour une preuve plus complète se basant sur une approche différente, celle du *principe de conservation de l'information.* □

Le clonage parfait reste possible si $|\psi\rangle$ appartient à un ensemble connu d'états *orthogonaux* (ce qui arrive en particulier si les qubits sont dans un état classique). Dans ce cas-ci, cloner pourrait être simplement réalisé en mesurant dans la base orthonormale formée par ces états, ce qui révèlerait exactement quel $|\psi\rangle$ était encodé dans cette base. Avec cette information, rien n'empêche ensuite en principe de réaliser autant de copies de $|\psi\rangle$ que désiré.

Le deuxième théorème fondamental [115], qui est dû à Holevo, borne la quantité d'information qui peut être extraite d'un système quantique.

[2] *Le terme "états orthogonaux" réfère à des états qui sont parfaitement distingables entre eux, tel que l'état $|0\rangle$ et l'état $|1\rangle$, ou encore l'état $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ et l'état $\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$.*

Théorème 2.3 (Borne de Holevo [59, 115] (version simplifiée)). *Si Alice veut envoyer n bits d'information à Bob via un canal quantique, elle devra pour cela lui envoyer au moins n qubits si Alice et Bob ne sont pas intriqués et au moins $\frac{n}{2}$ s'ils le sont.*

Bien que ce théorème semble destiné en premier lieu au domaine de la communication, il a aussi pour corollaire immédiat de limiter le nombre de bits d'information qui peuvent être extraits en mesurant un système quantique.

Corollaire 2.1 (Extraction d'information classique d'un système quantique). *Il est impossible d'extraire plus de n bits classiques d'un système quantique composé de n qubits.*

Démonstration. Supposons pour faire une preuve par contradiction, qu'il soit possible d'extraire plus de n bits classiques d'un système quantique composé de n qubits. Cela voudrait dire qu'Alice pourrait préparer et envoyer ce système pour transmettre plus de n bits classiques d'information à Bob ce qui contredirait la borne de Holevo. □

Ainsi, bien qu'il puisse être nécessaire de représenter un système de n qubits avec un nombre exponentiel d'amplitudes par rapport à n, seul une quantité linéaire d'information peut en être extraite.

Le troisième théorème fondamental [25], qui est dû à Bennett, Brassard et Mermin, est une extension du théorème de non-clonage (théorème 2.2).

Théorème 2.4 (Impossibilité d'apprendre de l'information sur un système sans le perturber [25]). *Aucun appareil de mesure ne peut avoir une probabilité non-zéro de révéler de l'information classique sur un état quantique $|\psi\rangle$ sans avoir en retour une probabilité non-zéro de perturber l'état de façon irréversible (sauf s'il est connu d'avance que $|\psi\rangle$ appartient à un ensemble d'états orthogonaux).*

2.3.3 Codage superdense et téléportation quantique

Soit un état de Bell vu dans la section précédente, dont on confie la moitié de la paire à Alice et l'autre moitié à Bob. Si Alice applique l'opération Not *localement* sur sa moitié de la paire, il est possible pour elle de transformer $|\Psi\rangle$ en $|\Phi\rangle$, et vice versa. De même, si elle applique localement le changement de phase P sur sa particule, elle peut faire passer « $^+$ » vers « $^-$ », et inversement. Nous avons maintenant tous les outils sous la main pour redériver le *codage superdense* [26], qui fut inventé par Bennett et Wiesner,

et qui permet à Alice de transmettre deux bits classiques à Bob par l'envoi d'un seul qubit s'ils partagent préalablement un état de Bell. Le protocole est le suivant. Alice et Bob partagent un état $|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ et Alice souhaite transmettre deux bits d'informations a et b à Bob. Si $a = 1$ alors Alice applique P sur sa particule, et si $b = 1$ elle va appliquer Not. Ensuite, elle envoie par un canal quantique sa moitié à Bob ; puis Bob applique l'inverse du circuit décrit dans la figure 2.8 et retrouve en sortie a et b qu'Alice voulait lui transmettre.

Définition 2.14 (Codage superdense). *Le codage superdense permet de* transmettre deux bits classiques au prix de l'utilisation d'un état de Bell et de la transmission d'un seul bit quantique.

De fait, pour transmettre $2n$ bits classiques de Alice vers Bob, il est possible soit de transmettre $2n$ qubits, soit de partager n états de Bell entre Alice et Bob et de faire transiter ensuite n qubits en utilisant le codage superdense.

Définition 2.15 (Téléportation quantique). *La* téléportation quantique *[24] permet de réaliser une tâche complémentaire du codage superdense, celle de* transporter un bit quantique inconnu $|\psi_?\rangle = \alpha_? |0\rangle + \beta_? |1\rangle$ de Alice vers Bob au prix de la transmission de 2 bits classiques et du sacrifice d'un état de Bell.

Le terme téléportation réfère au fait qu'à la fin de la téléportation, Alice n'a plus entre ses mains l'état $|\psi_?\rangle$ mais que c'est Bob qui en est maintenant le détenteur. La téléportation va donc permettre de transporter un qubit, mais pas de le cloner. Il faut aussi noter que la vitesse de cette téléportation est bornée par celle de la lumière puisqu'il faut le temps aux deux bits classiques de se rendre de Alice vers Bob pour que la téléportation soit complétée. Voir l'article original des six pères de la téléportation [24] pour plus de détails sur son fonctionnement.

2.3.4 Matrices densité

Définition 2.16 (Matrice densité). *La* matrice densité *est un formalisme qui permet de* représenter la connaissance que quelqu'un a sur un système quantique. *En particulier, elle constitue une* description complète *de ce qui peut être observé sur l'état du système [156].*

Si on sait à l'avance que l'état quantique $|\psi\rangle$ est dans une superposition spécifique d'états, alors la matrice densité ρ de cet état est égale à $|\psi\rangle\langle\psi|$.[3] Dans ce cas, on dit que ρ est un état *pur* et on a la simple relation $\rho^2 = \rho$. Un *mélange statistique* est une distribution de probabilités sur un ensemble d'état purs $\{(p_1, |\psi_1\rangle), \ldots, (p_k, |\psi_k\rangle)\}$, où $|\psi_i\rangle$ représente un état pur particulier et p_i est la probabilité associée à cet état, qui respecte la relation $\sum_{i=1}^{k} p_i = 1$. La matrice densité d'un mélange ρ est définie comme $\sum_{i=1}^{k} p_i |\psi_i\rangle\langle\psi_i|$. Si notre ignorance à propos d'un état est complète, alors on dit que cet état est *complètement mélangé*. Dans ce cas, $\rho = \frac{1}{d}\mathbf{I}$ où d est la dimension de l'espace de Hilbert dans lequel ρ vit, et $\mathbf{I}$ est la matrice identité.

2.4 Quelques résultats quantiques

Dans cette section, nous allons brièvement énoncer trois résultats parmi les plus importants du domaine. Dans l'ordre, nous allons passer à travers les deux algorithmes qui ont révolutionné le domaine et qui l'ont fait prendre son essor, celui de Grover et celui de Shor, puis passer en revue le premier résultat qui connaît une application commerciale, celui de la cryptographie quantique.

2.4.1 Algorithme de Grover

Un des deux algorithmes fondamentaux est l'algorithme de Grover [103]. Cet algorithme s'attaque au problème de la *recherche d'un élément dans un espace de recherche non structuré*. Formellement, on peut définir ce problème comme la recherche de l'élément unique x satisfaisant la relation $f(x) = 1$ alors que pour tous les autres éléments on aura $f(x) = 0$. Classiquement, l'algorithme le plus naïf qui consiste à interroger la base de données (possiblement de manière aléatoire) jusqu'à ce qu'on trouve l'élément est aussi le plus efficace. En moyenne, il faudra interroger la base de données sur la moitié de ses entrées avant de trouver l'élément souhaité, ce qui donne donc une complexité de $\Theta(n)$. Quantiquement, l'algorithme de Grover permet de résoudre le même problème en temps $\Theta(\sqrt{n})$.

[3] La notation *bra* $\langle\psi|$ est la notation duale de la notation *ket* $|\psi\rangle$ que nous avons utilisée précédemment sans la nommer. Formellement, $|\psi\rangle$ est un *vecteur colonne* représentant un état quantique vivant dans un espace de Hilbert et $\langle\psi|$ est la *transposée conjuguée* de ce vecteur. Par conséquent, $|\psi\rangle\langle\psi|$ est une matrice carrée.

La différence fondamentale entre classique et quantique, c'est que dans un monde quantique il est possible d'interroger la base de données en lui donnant comme entrée une superposition d'états alors que classiquement on ne peut l'interroger qu'élément par élément. L'algorithme de Grover est un algorithme itératif où chaque étape, appelée *itération de Grover*, consiste en une série de transformations unitaires simples. Voir la section 5.2 pour plus de détails sur les variantes de Grover.

2.4.2 Algorithme de Shor

En cryptographie, il arrive souvent que la sécurité de cryptosystèmes soit fondée sur des hypothèses calculatoires reliées à la difficulté de résoudre un certain problème. En particulier, on suppose qu'il existe des fonctions, appelées *fonctions à sens unique*, pour lesquelles il est facile de calculer $f(x)$ sachant x mais qui sont difficiles à inverser, c'est à dire de retrouver x sachant $f(x)$. La *factorisation* est considérée comme une de ces fonctions. Ainsi, il est facile de multiplier deux grands nombres premiers mais aucun algorithme classique efficace pour retrouver ces deux nombres à partir de seulement leur produit n'a été découvert pour l'instant[4]. Le système de cryptographie à clé publique RSA [164] (pour Rivest, Shamir et Adleman), qui est utilisé couramment de nos jours pour sécuriser les paiements électroniques, base sa sécurité exclusivement sur la difficulté de la factorisation.

Classiquement, le meilleur algorithme classique de factorisation connu actuellement (le *number field sieve* [136]) requiert un temps *quasi-exponentiel* par rapport à la taille du nombre à factoriser. Quantiquement, l'algorithme découvert par Shor en 1994 permet de *factoriser le produit de deux grands nombres premiers en temps polynomial* [172]. Cet algorithme utilise comme outil la *transformée de Fourier quantique*, qui peut être vue comme l'adaptation quantique de la transformée de Fourier rapide standard. La transformée de Fourier peut être implémentée de façon très efficace (en un nombre polynomial de portes) sur un ordinateur quantique. L'algorithme de Shor est un algorithme probabiliste de type Las Vegas qui utilise la transformée de Fourier quantique comme sous-routine. Pour être précis, l'algorithme ne s'attaque pas directement à la recherche des facteurs mais le réduit au problème de trouver la période d'une fonction injective sur sa

[4]Ou alors s'il a été découvert, il n'a pas été divulgué pour le moment.

période. Shor résout ce problème et utilise ensuite la réduction existante pour factoriser. Dans son article, Shor détaille aussi un algorithme permettant de résoudre efficacement le *logarithme discret*, un autre problème pour lequel on ne connaît pas de solution classique efficace et sur lequel sont bâtis la sécurité présumée de certains cryptosystèmes.

2.4.3 Cryptographie quantique

Comme nous l'avons vu dans les sections précédentes, bien utilisée l'information quantique peut permettre de réaliser des tâches qui semblent impossibles ou difficiles classiquement tel que factoriser efficacement. En particulier, l'algorithme de Shor condamne à plus ou moins court-terme les cryptosystèmes à clé publique tel que RSA. Il existe cependant un remède fourni par l'information quantique elle-même, celui de la *cryptographie quantique*. Classiquement, il existe un protocole permettant à deux personnes de communiquer de façon inconditionnellement sécuritaire. Ce *protocole du masque jetable*, qui a été découvert par Vernam [183], perfectionné par Mauborgne et prouvé sécuritaire par Shannon [170], requiert que les deux participants partagent une clé secrète au moins aussi longue que le message qu'ils veulent échanger. Une fois qu'un message a été chiffrée avec cette clé secrète, elle ne pourra malheureusement pas être réutilisée sous peine de complètement ruiner la sécurité du protocole. Le problème est que dans un monde classique, deux participants ne peuvent pas générer de clé secrète commune à moins de se rencontrer physiquement ou de déjà disposer d'un canal sécuritaire.

Dans le monde quantique, la situation est différente. En utilisant le *protocole de distribution de clés BB84* [23], Alice et Bob peuvent, s'ils disposent d'un canal quantique les reliant, réussir à générer une clé secrète aléatoire commune, et ceci même sous les yeux d'un espion disposant d'une puissance de calcul illimitée. Sans entrer dans les détails, si l'espion essaye d'apprendre de l'information sur les qubits qu'il voit passer sur le canal, il va nécessairement créer une perturbation qui pourra ensuite être détectée par Alice et Bob. Ce protocole offre une sécurité inconditionnelle qui n'est pas basée sur des hypothèses calculatoires mais sur les lois physiques de la mécanique quantique. La cryptographie quantique est la première application quantique qui est suffisament mature pour être passée du laboratoire à l'étape commerciale[5].

[5]Elle est actuellement vendue en ligne par plusieurs compagnies dont idQuantique (`http://www.idquantique.com`) et MagiQ Technologies (`http://www.magiqtech.com`).

CHAPITRE 3

SURVOL DE L'APPRENTISSAGE MACHINE

3.1 Présentation de l'apprentissage machine

Définition 3.1 (Apprentissage machine). *L'apprentissage machine [79, 98, 114, 145, 182] étudie les techniques permettant de donner à la machine la capacité d'apprendre à partir d'expériences passées.*

Contrairement à l'informatique "traditionnelle" où l'on programme explicitement la machine sur comment elle doit réagir en fonction de chaque situation, l'apprentissage machine cherche à donner la capacité d'apprendre à partir d'exemples de situations passées afin de pouvoir généraliser dans le futur à des situations nouvelles. Le domaine peut lui-même être séparé en trois sous-domaines : l'*apprentissage supervisé*, l'*apprentissage non-supervisé* et l'*apprentissage par renforcement.*

L'apprentissage supervisé et non-supervisé seront détaillés dans les sections suivantes. L'apprentissage par renforcement [177] cherche à représenter l'apprentissage d'un agent motivé par un but, qui interagit avec un environnement incertain afin de maximiser une fonction d'utilité/récompense. L'agent peut influer sur l'environnement par ses actions et ainsi apprendre de façon dynamique par des stratégies d'essai/erreur. Nous n'aborderons pas plus en détails l'apprentissage par renforcement dans ce chapitre, car il est relativement hors de propos par rapport à cette thèse. Nous verrons cependant dans le chapitre 4 qu'il existe des travaux ayant *quantisés* certains algorithmes d'apprentissage par renforcement (cf. section 4.3).

3.2 Apprentissage supervisé

Un *algorithme d'apprentissage* apprend à partir d'un ensemble de données, appelé *ensemble d'entraînement*, passé en entrée à l'algorithme. Cet ensemble d'entraînement contient des observations sur des objets, collectées soit en observant le passé, soit directement acquises auprès d'experts. Dans le cas de l'apprentissage supervisé, un ensemble d'entraînement contenant n points de données est décrit comme

$D_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, où x_i est un *vecteur représentant des observations* sur les caractéristiques du $i^{\text{ème}}$ objet (ou point de données)[1] de l'ensemble d'entraînement et y_i la *cible* associée à cet objet. Le but de l'algorithme d'apprentissage est d'apprendre, à partir de D_n, une transformation $f : x \mapsto y$. Autrement dit, f est une fonction qui associe une cible y à un vecteur d'observations x. Cette fonction est la sortie produite par l'algorithme suite à la phase d'entraînement. Un cas typique serait celui où chaque objet peut être décrit en utilisant d attributs réels et où on s'intéresse à faire de la classification binaire (voir figure 3.1 pour un exemple). Dans ce cas-ci, $x_i \in \mathbb{R}^d$ et $y_i \in \{-1, +1\}$. Considérons l'exemple suivant, qui illustre la classification binaire de manière plus concrète.

Exemple 3.1 (Classification binaire). *Soit une banque, qui à partir des données de ses clients, voudrait apprendre un classifieur lui permettant d'évaluer si une personne pourra ou non réussir à rembourser un prêt (classe* positive *ou* négative*). Dans ce cas, d serait le nombre d'attributs servant à décrire chaque client, et y serait égale à "+1" pour les clients dans l'historique de la banque ayant remboursé leur prêt et "−1" pour ceux qui ont failli à leur remboursement. Une fois le classifieur f appris, on peut lui soumettre le vecteur d'attributs $x_?$ d'un nouveau client afin d'évaluer si le client est intéressant ou non pour la banque selon la valeur $y_?$ prédit.*

3.2.1 Classification, régression et ordonnancement

Les trois tâches principales de l'apprentissage supervisé sont : la *classification*, la *régression* et l'*ordonnancement*. En classification, la fonction f qu'on cherche à apprendre est appelé *classifieur*. Elle permet d'associer à chaque vecteur d'observations x, sa classe correspondante y. Si on considère la classification multiclasse avec k classes différentes, on a $y \in \{1, \ldots, k\}$.

Exemple 3.2 (Classification). *Reconnaître les empreintes digitales ou le visage d'une personne (dans ce cas-là chaque classe correspond à un individu), classifier automatiquement une nouvelle fraîchement publiée comme appartenant à la section "culture" ou "sports", détection de cas de fraudes, ...*

[1] Dans cette thèse, les termes "objet" et "point de données" possèdent la même signification et seront utilisés de manière interchangeable. De même, les termes "classe", "groupe" et "catégorie" seront eux aussi utilisés de façon interchangeable.

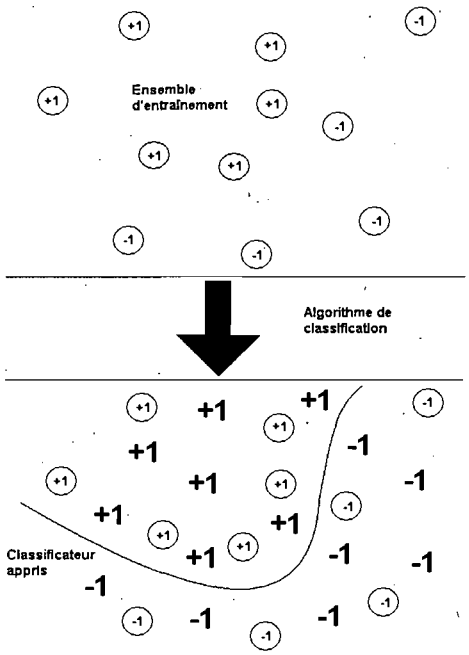

FIG. 3.1 – Illustration d'une tâche de classification binaire. À partir des points de données contenus dans l'ensemble d'entraînement, qui sont étiquetés d'après leur classe respective (ici « -1 » ou « +1 »), l'algorithme va apprendre une surface de décision qui fait la séparation des données et va jouer le rôle de classifieur.

En régression, on cherche à apprendre le même type de fonction que pour la classification. Cependant, au lieu que les y appartiennent à un ensemble fini, elles peuvent être choisies parmi un ensemble continu. Ainsi, la régression peut être vue comme la tâche de prédiction d'une certaine caractéristique de l'objet qui est choisie parmi un ensemble continu (et non discret comme dans le cas de la classification).

Exemple 3.3 (Régression). *Prédire l'âge d'un arbre à partir d'observations le concernant, estimer le degré de risque de non-remboursement par rapport à une personne (contre simplement risque/non-risque dans le cas de la classification) ou encore prévoir l'évolution future du cours d'une action en bourse.*

L'ordonnancement (ou *ranking* en anglais) cherche à ranger/ordonnancer différentes

instances de l'ensemble d'entraînement en fonction de leur pertinence par rapport à un point de données passé en entrée. L'ordonnancement correspond donc à prédire, pour chaque point d'entraînement, un score représentant sa pertinence par rapport à l'objet considéré actuellement.

Exemple 3.4 (Ordonnancement). *Supposons que chaque point de données de l'ensemble d'entraînement correspond à une page web et qu'on cherche à donner un score d'utilité pour chaque page par rapport à la requête actuelle. Dans le contexte de la recommandation de produits, on pourrait vouloir trier les articles d'une boutique en ligne en fonction de leur intérêt potentiel par rapport à un acheteur spécifique.*

3.2.2 Généralisation et validation

L'apprentissage machine est fondamentalement différent de l'*apprentissage par cœur*, où la machine chercherait simplement à mémoriser les exemples présentés sans rien apprendre d'eux. En effet, en apprentissage machine on espère que non seulement la machine va pouvoir reconnaître la plupart des exemples qu'on lui a présentés mais aussi et surtout qu'elle va pouvoir *généraliser* à des cas non observés auparavant. Pour tester l'efficacité d'un algorithme, on met généralement de côté un ensemble de données appelé *ensemble de test*, qui est indépendant de l'ensemble d'entraînement et sur lequel on testera la fonction f retournée par l'algorithme. L'erreur mesurée sur cet ensemble, appelée *erreur de test*, constitue un estimé de la véritable *erreur de généralisation*. Ce n'est pas le cas de l'erreur d'entraînement qui n'est généralement pas un bon indicateur de l'erreur de généralisation (car étant biaisée), bien que constituant elle-même une borne inférieure de l'erreur de test. Il existe souvent un compromis fondamental à faire entre le *surapprentissage*, où on aura choisi une représentation qui est très proche des données d'entraînement mais où la généralisation sera faible, et le *sous-apprentissage* où au contraire on aura considéré un ensemble de fonctions possibles d'où f est choisie qui n'est pas assez riche pour représenter la diversité présente dans les données.

Afin d'éviter ces deux situations qui conduisent à une faible généralisation, on cherche à contrôler la complexité du modèle et ensuite à évaluer la performance de l'algorithme sur différentes possibilités. Les méthodes d'évaluation les plus communément utilisées sont :

(1) La *validation simple*, où l'ensemble de données est séparé en un seul ensemble d'entraînement et de test (en utilisant par exemple 80 % des données pour l'entraînement et 20 % pour le test).

(2) La *validation croisée*, où on répète plusieurs fois l'étape d'entraînement/test mais en mélangeant aléatoirement les données à chaque fois. La mesure d'erreur considérée est la moyenne des erreurs obtenues lors des différentes tentatives.

(3) La *validation par blocs* [73] (appelée *k-fold cross-validation* en anglais), où on coupe l'ensemble de données en k blocs distincts. Chaque bloc sert une seule fois en temps qu'ensemble de test et est utilisé le reste du temps comme ensemble d'entraînement. Un avantage de la validation par blocs est que comme chaque point de données est utilisé une seule fois pour le test, l'estimé obtenu de l'erreur de généralisation est plus robuste et moins biaisé que dans le cas de la validation croisée [131].

(4) La *validation "jackknife"* (aussi appelée *leave-one-out*), où on entraîne sur tous les points sauf un sur lequel on mesure l'erreur de test. On répète ensuite cette procédure pour chacun des points de données (ce qui revient à faire une validation par blocs avec $k = n$ blocs, pour n le nombre de points de données total).

Les mêmes méthodes de validation peuvent être utilisées afin d'optimiser un *hyperparamètre* de l'algorithme, comme par exemple celui qui contrôle la complexité du modèle. Dans ce cas, on construit un troisième ensemble appelé *ensemble de validation*. Une fois que le (ou les) paramètres de l'algorithme ont été optimisés, les données de l'ensemble de validation sont ensuite intégrées à l'ensemble d'entraînement pour la vraie phase d'entraînement.

3.2.3 Algorithmes d'apprentissage

Le but de l'apprentissage machine n'est pas de trouver l'algorithme d'apprentissage "ultime" mais plutôt de développer un sac d'outils contenant une variété d'algorithmes afin de pouvoir s'adapter à différentes situations d'apprentissage. En effet, il n'existe aucun algorithme d'apprentissage qui puisse battre tous les autres pour toutes les distributions possibles de données[2]. Cette notion peut être formalisée sous la forme d'un

[2]Cet énoncé n'exclut pas qu'un algorithme d'apprentissage particulier puisse "battre" les autres algorithmes sur la plupart des distributions de données rencontrées en pratique, qui constituent un petit sous-ensemble de toutes les distributions théoriquement possibles.

théorème d'impossibilité pour l'apprentissage du type "no-free-lunch".

Théorème 3.1 (No-free lunch pour l'apprentissage supervisé (version simplifiée) [191]). *Au regard de la probabilité de bien généraliser sur un point de données non-observé auparavant, tous les algorithmes sont équivalents en moyenne. Autrement dit, pour n'importe quelle paire d'algorithmes d'apprentissage, il y a théoriquement autant de situations (de distributions des données possibles) où le premier algorithme battra le second que vice-versa.*

Dans le cas de l'apprentissage supervisé, il existe de nombreux algorithmes très différents tel que :

(1) Les *méthodes de voisinage* sont parmi les algorithmes d'apprentissage les plus simples qui existent. Lorsque le temps de classifier un point inconnu $x_?$ est arrivé, une méthode de voisinage cherche les voisins de $x_?$ dans l'ensemble d'entraînement D_n et base sa prédiction sur un vote de majorité par rapport aux classes de ses voisins. Parmi ces méthodes se trouvent les k-plus proches voisins [71], où on considère seulement les k voisins les plus proches de $x_?$, et les *fenêtres de Parzen* [155], où on va considérer tous les points en leur associant un poids qui dépend de leur distance avec $x_?$. Un cas particulier des fenêtres de Parzen considère seulement les points qui sont dans un voisinage fixe de distance ϵ.

Avantages :

- Très simple conceptuellement et facile à implémenter.
- Robuste par rapport au bruit.
- Ne requiert aucune phase d'entraînement (tout le travail s'effectue au moment de la classification).

Inconvénients :

- Le paramètre de l'algorithme, tel que k (le nombre de voisins) ou ϵ (la taille du voisinage), doit être choisi avec discernement pour obtenir une bonne généralisation (souvent par validation croisée).
- Importance de la pertinence de la distance utilisée. Certaines techniques apprennent la mesure de distance qui est la plus informative par rapport à une situation d'apprentissage spécifique (voir [99] pour un exemple adapté aux méthodes de voisinage).

- Calculer la distance entre $x_?$ et tous les points de l'ensemble d'entraînement peut être coûteux en temps de calcul (surtout si n est grand). Certaines structures de données permettent cependant de rendre la recherche des voisins plus efficace lorsqu'on est en faible dimension (voir [27] par exemple).

(2) Dans un *arbre de décision* [160, 161], chaque *nœud* représente un *test sur un ou plusieurs attributs* dont le résultat va déterminer quelle branche suivre pour descendre dans l'arbre. Chaque *feuille* est en général étiquetée par une classe. Pour classifier un nouveau point de données, on parcourt l'arbre en partant de la racine jusqu'à atteindre une feuille et on prédit la classe qui correspond à l'étiquette de cette feuille. Crée par Quinlan, ID3 [160] (pour *Iterative Dichotomiser 3*) est parmi les tous premiers algorithmes développés pour construire un arbre de décision. Il s'agit d'un algorithme récursif qui construit l'arbre en créant à chaque récursion un nœud dont le test porte sur l'attribut qui *maximise le gain d'information* (ou de façon équivalente *minimise l'entropie de Shannon*).

Avantage :

- Compréhensibilité du modèle. Les arbres de décision sont parmi les seules méthodes dont la description du modèle permet d'expliciter la raison des prédictions effectuées par celui-ci.

Inconvénients :

- Instabilité de l'algorithme. Une petite perturbation des données de l'ensemble d'entraînement peut conduire à la génération d'un arbre de décision complètement différent. Les méthodes d'ensemble comme les *forêts aléatoires* [49], qui basent leurs prédictions sur une forêt d'arbres plutôt que sur un arbre individuel, permettent de contourner ce problème tout en améliorant au passage la généralisation.
- Problème de surapprentissage si l'arbre construit est trop profond. Certains algorithmes (comme C4.5 [161]) utilisent des techniques d'élagage pour contrôler explicitement la profondeur et la taille de l'arbre.

(3) Les *classifieurs bayésiens* [135] estiment explicitiment pour un nouveau point de données *sa probabilité d'appartenance à chacune des classes* en se basant sur les données de l'ensemble d'entraînement. Un classifieur de Bayes naïf (appelé *naïve Bayes* en anglais) calcule ces probabilités d'appartenance en se basant sur une ap-

plication directe de la formule de Bayes et sur l'hypothèse que tous les attributs sont indépendants les uns des autres (d'où le terme "naïf"). Cette hypothèse simplifie considérablement la formule de Bayes et donc son évaluation explicite.

Avantages :

- Très simple conceptuellement et facile à implémenter.
- Minimise l'erreur de Bayes.

Inconvénients :

- L'hypothèse d'indépendance des attributs d'un classifieur de type Bayes naïf est souvent non fondée en pratique.
- Minimiser l'erreur de Bayes ne garantit pas de pouvoir généraliser correctement.

(4a) Les *réseaux de neurones artificiels* [36] s'inspirent du fonctionnement des réseaux de neurones dans le cerveau humain. Un réseau de neurones artificiels se compose de plusieurs couches de neurones, dont au moins une couche d'entrée et une couche de sortie. Les neurones sont connectés entre eux par des *synapses*, c'est à dire des liens comportants des poids. Chaque neurone est une petite unité de calcul qui prend en entrée les valeurs retournées par les neurones de la couche précédente multipliées par les poids des synapses, leur applique une fonction (comme par exemple la fonction *sigmoïde* ou *tanh*) puis produit en sortie la valeur résultante.

Avantages :

- En pratique, les réseaux de neurones offrent généralement une bonne performance s'ils sont correctement optimisés.
- Le *théorème de l'approximation universelle* [90] précise qu'il est possible d'approximer n'importe quelle fonction sur les nombres réels avec un réseau de neurones contenant une seule *couche cachée*[3] (c'est à dire une seule couche intermédiaire entre la couche d'entrée et de sortie).

Inconvénients :

- Incompréhensibilité du modèle. Le réseau de neurones fonctionne à peu près comme une boîte noire et il est difficile d'expliquer ce qui l'a conduit à faire une prédiction particulière.
- Beaucoup de paramètres à optimiser. Pour que le réseau de neurones artificiels

[3] Il est cependant possible que cette couche contienne un nombre exponentiel de neurones par rapport au nombre de neurones dans la couche d'entrée.

offre une bonne généralisation, il faut optimiser de nombreux paramètres dont le nombre de neurones dans la (ou les) couche(s) cachée(s) ou le choix des fonctions calculées par les neurones.

(4b) Les *réseaux profonds* [113] sont des variantes des réseaux de neurones basées sur une architecture profonde qui comporte de nombreuses couches intermédiaires. En partant de la couche d'entrée, chaque couche suivante peut être vue comme apprenant des caractéristiques de la distribution des données à un niveau de plus en plus abstrait. Ces architectures sont devenues populaires récemment suite à des travaux de l'équipe de Hinton (université de Toronto) et de Bengio (université de Montréal) prouvant qu'il est possible d'apprendre efficacement les poids initiaux des liens reliant les différentes couches en utilisant une méthode non-supervisée. En partant ensuite de l'architecture apprise, et en y ajoutant une méthode supervisée, on obtient un algorithme qui généralise parfois mieux sur certaines ensembles de données que ce qui était auparavant considéré comme l'état de l'art. Leurs avantages et inconvénients sont relativement similaires à ceux des réseaux de neurones standards, si ce n'est que les réseaux profonds offrent souvent une meilleure généralisation.

(5) Les *machines à vecteurs de support* [182] se focalisent sur les points de données proches de la surface de décision (appelées *vecteurs de support*), plutôt que sur l'ensemble des points d'entraînement. Ces méthodes cherchent explicitement à maximiser la *marge*, qui est fonction directe de la distance entre les vecteurs de support et la frontière de décision. Il a été démontré, aussi bien théoriquement qu'empiriquement, que la maximisation de la marge conduit souvent à une bonne généralisation. Supposons que l'effet du bruit peut être modélisé comme déplaçant un point de données légèrement dans l'espace dans lequel il vit. Intuitivement, nous pouvons nous rendre compte qu'une marge importante réduit la probabilité que ce point "traverse" la surface de décision et donc que le classifieur fasse une prédiction erronée quant à sa classe. Afin de pouvoir s'occuper des données non-linéairement séparables, les machines à vecteur de support utilisent le "*truc du noyau*" [3] (appelé *kernel trick* en anglais) pour transformer implicitement l'espace de représentation des données d'entrée en un espace de plus haute dimension. Cette transformation vers le nouvel espace permet ensuite de séparer les données en utilisant un classifieur linéaire.

Avantage :

- Très bonne généralisation. En pratique, les machines à vecteurs de support sont parmi les méthodes qui obtiennent les meilleurs résultats.

Inconvénient :

- Gourmand en terme de temps de calcul. Les machines à vecteurs de support utilisent des méthodes d'optimisation pour maximiser la marge dont le temps d'exécution est souvent quadratique en n, le nombre de points de l'ensemble d'entraînement.

(6) Les *méthodes d'ensemble* sont l'illustration du proverbe "l'union fait la force" et combine plusieurs classifieurs en un seul classifieur efficace, souvent par un mécanisme de vote. Parmi les méthodes d'ensemble, on compte les algorithmes de type *boosting* (comme AdaBoost [89]), l'*empilement* [190] (ou *stacking* en anglais), le *bagging* [48] (pour *Bootstrap Aggregating*) ou encore les *forêts aléatoires* [49]. Chacune de ces méthodes d'ensemble a ses propres avantages et inconvénients. Ainsi, si on prend l'exemple d'AdaBoost (pour *Adaptive Boosting*), qui combine plusieurs classifieurs dits *faibles*[4] en un classifieur efficace, on peut identifier les avantages et inconvénients suivants.

Avantages :

- Très bonne généralisation. Dans de nombreux domaines, la performance d'AdaBoost est comparable à celle des autres algorithmes de l'état de l'art (tel que les machines à vecteurs de support).
- Il s'agit d'un des rares algorithmes pour lequel surapprentissage (comme amener l'erreur d'entraînement vers zéro) ne rime pas forcément avec mauvaise généralisation. Ceci peut-être expliqué par le fait qu'AdaBoost essaye indirectement de maximiser la marge.

Inconvénients :

- Parfois particulièrement sensible à certain type de bruit présent dans les données.
- Importance du choix des classifieurs faibles. En particulier, il ne faut pas que chaque classifieur faible pris individuellement soit trop "bon". Faute de quoi, il y a des risques que le classifieur final combiné soit pire en terme de généralisation que les classifieurs faibles considérés individuellement.

[4] Un classifieur faible est un classifieur dont les prédictions peuvent être à peine meilleures qu'un choix aléatoire.

Pour résumer, chaque méthode d'apprentissage est appropriée à certaines situations et possède des avantages et des inconvénients qui lui sont propres. Parmi les avantages, on peut citer : une bonne performance en terme de généralisation pour les machines à vecteurs de support, les méthodes d'ensemble ou encore les architectures profondes, un pouvoir d'explication quant aux prédictions pour les arbres de décision, la minimisation de l'erreur de Bayes pour les classifieurs bayésiens ou encore la nécessité de faire des calculs seulement au moment de la classification pour les méthodes de voisinage. Côté inconvénients, on trouve : la sensibilité à des faibles variations dans les exemples ou dans l'ensemble d'entraînement pour les arbres de décision, le manque de clarté dans la raison des prédictions pour les réseaux de neurones ou encore le coût en temps de calcul élevé pour classifier un point de test pour les méthodes comme les k-plus proches voisins ou les fenêtres de Parzen.

La plupart des algorithmes d'apprentissage décrits plus haut ont été formulés initialement pour résoudre la tâche de classification mais peuvent être adaptés pour résoudre les tâches de régression ou d'ordonnancement.

3.3 Apprentissage non-supervisé

L'apprentissage non-supervisé se différencie du cas supervisé principalement par le fait que dans le premier cas, la valeur de la classe y_i associée à chaque objet x_i est inconnue[5]. Plus précisément, on pourrait connaître le spectre des différentes étiquettes possibles mais ne pas connaître les étiquettes *spécifiques* de chacun des points de données, soit même *complètement ignorer le nombre de classes et leurs étiquettes.* Le but de l'apprentissage non-supervisé est de *découvrir la structure cachée se trouvant à l'intérieur des données.* Contrairement à sa contrepartie supervisée, elle constitue donc une approche plus exploratoire de l'apprentissage.

Les trois principales formes d'apprentissage non-supervisé sont :

(1) La *catégorisation*[6] (ou *clustering* en anglais) qui cherche à révéler les catégories

[5]Il existe une situation d'apprentissage intermédiaire appelée *apprentissage semi-supervisé* où l'on dispose de beaucoup de données non-étiquetées et peu de données étiquetées. En pratique, cette situation est courante car typiquement il « coûte » cher d'obtenir des données étiquetées si c'est un humain qui doit faire le travail d'étiquetage alors que collecter des données non étiquetées est plus facile. Voir [198] pour un état de l'art des algorithmes dans ce domaine.

[6]Cette tâche est aussi parfois appelée *partitionnement* en français.

naturelles présentes à l'intérieur des données.

(2) La *réduction de dimensionnalité* qui se préoccupe de trouver une représentation en faible dimension des données de l'ensemble d'entraînement qui, eux, sont en haute dimension[7].

(3) L'*estimation de densité* qui veut apprendre explicitement une fonction de probabilité (aussi appelé *fonction de densité*) qui représente la vraie distribution des données.

Ces trois tâches sont loin d'être indépendantes, et même au contraire fortement interconnectées. Ainsi, la catégorisation, qui associe à chaque point de données une catégorie, est une forme de réduction de dimensionnalité où chaque point de données pourrait ensuite être représenté sous la forme d'un seul index indiquant sa catégorie. Aussi, l'estimation de densité permet en général de faire de la réduction de dimensionnalité puisque la fonction de densité apprise correspond souvent une représentation compacte des données.

3.3.1 Catégorisation

Le problème de la catégorisation peut être formulé de la façon suivante. Soit l'ensemble d'entraînement $D_n = \{x_1, \ldots, x_n\}$ tel qu'on cherche à associer à chaque point de données une étiquette $y_i \in \{1, \ldots, k\}$ de manière à ce que les objets similaires se retrouvent associés à la même catégorie et que les objets dissemblables se retrouvent dans des catégories différentes (voir figure 3.2 pour une illustration).

Exemple 3.5 (Catégorisation). *Révéler les groupes sociologiques se trouvant à l'intérieur d'une population à partir de données démographiques, trouver les catégories typiques existant parmi les clients d'un supermarché, regrouper automatiquement les chansons qui sont musicalement proches, réunir dans un même groupe des gènes ayant un comportement biologique similaire afin d'inférer leur fonction.*

Pour fonctionner, un algorithme de catégorisation doit avoir accès à un critère lui permettant de comparer deux objets de l'ensemble d'entraînement, tel que par exemple une notion de *distance* ou une *mesure de similarité*. L'algorithme va ensuite chercher à regrouper les objets en se basant sur ce critère. Une des notions de distance la plus

[7] Autrement dit, le but est de trouver une représentation de l'ensemble de données original plus compacte tout en préservant au maximum l'information présent dans celui-ci, comme par exemple la distance entre chaque paire de points.

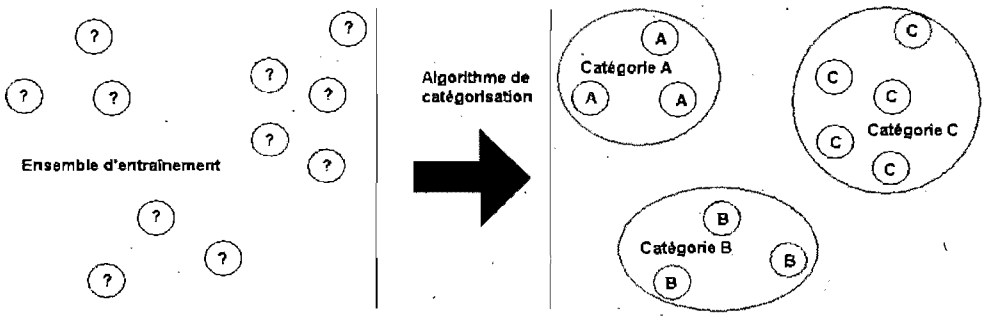

FIG. 3.2 – Illustration d'une tâche de catégorisation. Contrairement à l'apprentissage supervisé, les classes de chacun des points de données ne sont pas connues à l'avance et c'est le rôle de l'algorithme de catégorisation que de regrouper les objets similaires dans la même catégorie.

générale est la *distance de Minkowski* $d(x_a, x_b) = (\sum_{i=1}^{d} \parallel x_a^{(i)} - x_b^{(i)} \parallel^p)^{\frac{1}{p}}$. Elle équivaut à la *distance de Manhattan* (norme-L_1) si on choisit $p = 1$ ou la traditionnelle *distance euclidienne* (norme-L_2) si $p = 2$. Il est aussi possible de définir une distance dans un espace qui dépend directement de l'ensemble de données tel que la *distance de Mahalanobis* [140]. Le choix de la métrique est très important et a un impact substantiel sur le résultat de la catégorisation. En effet, c'est elle qui définit dans quelle mesure deux points de données sont proches ou éloignés. Chaque métrique réagit aussi différemment aux transformations qu'un point de données pourrait subir dans l'espace. Ainsi, à titre d'exemple, la distance euclidienne entre deux points est invariante sous translation, rotation et réflexion mais pas sous la plupart des autres transformations.

Pour représenter une catégorie, la méthode la plus simple consiste à utiliser un *centroïde*, c'est à dire à créer un point virtuel qui représente la moyenne des points de données formant la catégorie. Une autre approche, celle du *médoïde* représente la catégorie par un point de données existant qui est la médiane de la catégorie. Ces deux approches sont appropriées lorsque les points ont tendance à se regrouper en catégories ayant une forme sphérique sous la métrique considérée. Cependant, elles peuvent ne pas du tout capturer la distribution réelle des données si les catégories sont de forme plus complexes, ou même potentiellement arbitraires. Dans ces cas-là, une méthode cherchant à représenter une catégorie sous forme de quelques points significatifs (comme celle adoptée par l'al-

gorithme BIRCH [197]) donnera de biens meilleurs résultats.

Un principe général que cherche à suivre (implicitement ou explicitement) tout algorithme de catégorisation est de *maximiser l'intra-similarité* à l'intérieur d'une catégorie (c'est-à-dire d'essayer de faire en sorte que les points contenus dans une catégorie soient le plus similaires possible) et l'*extra-dissimilarité* entre les catégories (c'est-à-dire que les points de deux catégories différentes doivent être le plus dissemblables possible). Pour mesurer l'intra-similarité, on utilise généralement la distance moyenne entre les points de la catégorie comme métrique mais d'autres mesures sont possibles comme une métrique qui serait fonction de la densité. Pour mesurer la dissimilarité entre deux catégories, les mesures qui sont couramment utilisées sont celles de la *distance minimum*, de la *distance moyenne* ou de la *distance maximum* séparant les points des deux catégories. Un choix particulier de mesure va influencer la formation des catégories. Ainsi, si on choisit la distance maximum comme mesure de dissimilarité, il est possible que deux catégories qui sont très différentes, mais qui sont situées proches l'une de l'autre dans l'espace des données, se retrouvent à être fusionnées en une seule catégorie.

L'algorithme des *k-moyennes* [138] est l'archétype d'un algorithme de catégorisation. Il est d'ailleurs systématiquement utilisé pour fin de comparaison lors de la mise au point d'un nouvel algorithme de catégorisation. Voir l'algorithme 1 pour les détails sur son fonctionnement.

Algorithme 1 k-moyennes(D,k)

Choisir k points uniformément au hasard comme étant les centres initiaux des catégories
Répéter
  Pour chaque point de données dans D **faire**
    Attacher ce point à son centre le plus proche
  fin pour
  Pour chaque catégorie Q **faire**
    Recalculer le centre de la catégorie en faisant la moyenne des points à l'intérieur de cette catégorie
  fin pour
Jusqu'à stabilisation des catégories
Retourner les catégories trouvées et leurs centroïdes

L'algorithme des k-moyennes procède de manière itérative jusqu'à ce que les points de données soient assignés de manière fixe à un centroïde (c'est-à-dire qu'il y ait stabi-

lisation). Il s'agit d'un algorithme de type "*hill-climbing*" dont on peut démontrer qu'il cherche à minimiser une fonction d'erreur quadratique. Comme la plupart des algorithmes de hill-climbing, il y a toujours possibilité qu'il se retrouve coincé dans un minimum local. En pratique, il est relativement rapide et sa convergence se fait en général en un nombre d'itérations plus petit que n, la taille de l'échantillon. L'algorithme possède cependant deux restrictions importantes : celui d'assigner une catégorie unique à chaque point de données et celui de connaître à l'avance k, le nombre de catégories présentes dans les données. Il est néanmoins possible de remédier à ces restrictions ; respectivement en développant une *variante floue* de l'algorithme qui permet à un point de données d'appartenir à plusieurs catégories à différents degrés [35] et en utilisant des techniques permettant d'*estimer le nombre optimal k de catégories* [105] (en utilisant par exemple une méthode de validation croisée).

Tout comme les algorithmes d'apprentissage supervisé, il existe de très nombreux types d'algorithmes de catégorisation qui abordent chacun le problème de la catégorisation sous un angle différent (voir [29, 119] par exemple). Parmi les familles d'algorithmes de catégorisation les plus répandues, on trouve :

(1) Les algorithmes de type *hiérarchique* qui construisent un arbre (aussi appelé *dendrogramme*) qui représente une hiérarchie de catégories. La racine de l'arbre est formée par l'ensemble de la population des points de données alors que les feuilles de l'arbre sont constituées d'un seul point de données ou encore d'un groupe de points très similaires. Les algorithmes *agglomératifs* [66] construisent cet arbre par le bas en faisant fusionner itérativement les points de données avec les catégories les plus similaires. Les algorithmes utilisant l'approche *divisive* construisent plutôt l'arbre par le haut en divisant la population de manière récursive jusqu'à qu'une certaine condition d'arrêt soit remplie.

(2) Les algorithmes de type *partitionnement* (dont fait partie l'algorithme des k-moyennes) assignent les points à des catégories en utilisant une approche gloutonne. Ces algorithmes fonctionnent de manière itérative en essayant d'optimiser un critère global. Ils s'arrêtent lorsqu'un optimum (possiblement local) est atteint et que les points sont stabilisés dans leurs catégories respectives.

(3) Les algorithmes de catégorisation qui travaillent à partir d'un *graphe* représentant la

structure de l'ensemble de données [106]. Dans ce graphe, les nœuds sont constitués par les points de données et les arêtes ont un poids proportionnel à la distance entre les nœuds. Les algorithmes de cette famille manipulent la structure de graphe pour révéler les catégories.

(4) Les algorithmes basés sur la *densité* (comme DBSCAN [82] ou DENCLUE [112]) construisent des catégories qui sont denses et fortement connectées. Leur principal avantage est de pouvoir considérer des catégories de forme arbitraire mais aussi d'avoir la tendance à ne pas connecter deux catégories qui sont relativement proches mais séparées par un espace de faible densité.

(5) Les algorithmes de catégorisation *conceptuelle* (comme COBWEB [87] ou CLARISSE [5]) ne se contentent pas de regrouper des points ensemble mais expliquent aussi leurs choix. Ces méthodes ont l'avantage d'offrir une interprétation de leurs résultats contrairement à la plupart des autres méthodes qui produisent souvent des catégories sans les expliciter.

Une des façons standards de tester la performance et de comparer des algorithmes de catégorisation est de prendre un ensemble de données dont on connaît les étiquettes individuelles des points, mais dont on cache cette information à l'algorithme. Ensuite, pour mesurer la qualité des catégories retournées par l'algorithme, on regarde s'il a été capable de grouper ensemble les points ayant à l'origine la même étiquette ou si, au contraire, il les a placés dans des catégories différentes.

Une autre manière de tester un algorithme est d'avoir un critère global [143], idéalement indépendant de l'algorithme utilisé, qui permet de juger de la qualité de la catégorisation retournée. Ce critère est généralement fonction directe de l'intra-similarité et de l'extra-dissimilarité des catégories. Cependant, c'est souvent l'humain qui aura le dernier mot pour évaluer la pertinence d'une catégorisation particulière. En effet, il n'existe pas de critère universel permettant de le faire et il y aura toujours une part de subjectivité quant à l'interprétation des catégories obtenues.

3.3.2 Réduction de dimensionnalité

En pratique, il est courant que les points de l'ensemble de données soient décrits par un nombre important d'attributs (*espace à haute dimension*) alors qu'en réalité ils

ont été générés dans un sous-espace de dimension beaucoup plus faible. Le but de la réduction de dimensionnalité est de trouver une représentation en faible dimension. Soit l'ensemble données $D_n = \{x_1, \ldots, x_n\}$, tel que les points de données x_i sont définis dans un espace de dimension d. Un algorithme de réduction de dimensionnalité va apprendre une application $f : x_i \mapsto x'_i$ telle que les points obtenus x'_i soient décrits par k attributs, pour $k \ll d$. Pour que cette définition ait du sens, il faut aussi préciser que l'application apprise doit préserver au maximum l'information présente dans les données, dans un sens qui sera spécifique à la méthode de réduction de dimensionnalité utilisée.

Exemple 3.6 (Réduction de dimensionnalité). *Compression d'images ou de sons, visualisation en deux ou trois dimensions de données.*

L'*analyse en composantes principales* [120] (ACP, aussi parfois appelée transformation de Karhunen-Loève) découvre une transformation linéaire qui projette les points de données dans un sous-espace de faible dimension tout en préservant la variance présente à l'origine dans les données. L'algorithme part des coordonnées des points de données et apprend les *vecteurs et les valeurs propres* de la distribution à partir de la matrice de covariance. Seul les k valeurs et vecteurs propres les plus significatifs (appelés composantes principales) sont ensuite gardés et utilisés pour réaliser la projection des données dans l'espace de dimension k. Dans le cas de l'ACP, l'erreur de reconstruction engendrée par la réduction de dimensionnalité est directement proportionelle à la somme des valeurs propres non utilisées.

L'*échelonnement multidimensionnel* [62] cherche une représentation en faible dimension qui préserve le mieux possible la distance entre chaque paire de points de données. Il est possible de l'utiliser même si on dispose seulement des distances (ou d'une mesure de similarité) entre chaque paire d'exemples mais pas de leurs coordonnées exactes. Il a été démontré que cet algorithme produit exactement les mêmes résultats que l'ACP si on utilise la distance euclidienne comme métrique. Aussi bien l'ACP que l'échelonnement multidimensionnel sont couramment utilisés comme *étape de prétraitement* avant d'utiliser d'autres algorithmes d'apprentissage. Ils permettent en effet de réduire considérablement le nombre d'attributs nécessaires pour représenter un point de données tout en préservant l'information contenue dans les données.

Lorsque le sous-espace ayant réellement engendré les données n'est pas linéaire mais de

courbure beaucoup plus complexe, l'ACP et l'échelonnement multidimensionnel ne sont pas vraiment appropriés et il faut faire appel à des méthodes permettant d'identifier des variétés non-linéaires. Isomap [179] est un algorithme qui préserve la *distance géodésique* entre les points de données, c'est-à-dire la distance pour aller d'un point à un autre en suivant la courbure du sous-espace réel. Dans l'espace de haute dimension, deux points peuvent sembler proches alors qu'en vérité leur distance géodésique dans l'espace de faible dimension est grande (cf figure 3.3 pour un exemple).

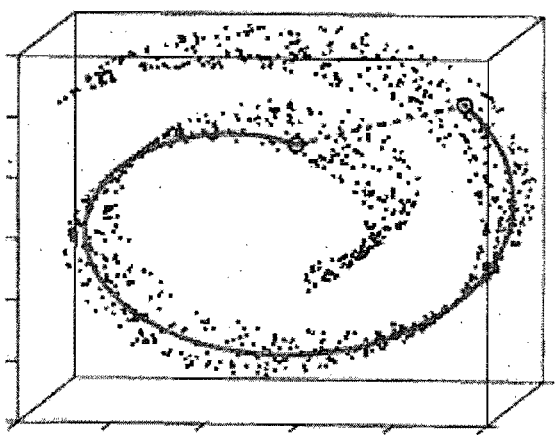

FIG. 3.3 – Exemple du *rouleau suisse* tiré de [179]. La distance géodésique entre les points de données dans le sous-espace réel peut être très différente de la distance telle que mesurée dans l'espace en haute dimension.

L'idée principale d'Isomap (voir algorithme 2) est d'approximer la distance géodésique localement par la distance entre deux points puis globalement par le plus court chemin à travers les voisinages.

LLE (pour *Local Linear Embedding* en anglais) [165] est une autre méthode de réduction de dimensionnalité capable d'identifier des courbures de l'espace non-linéaire. Tout comme Isomap, elle capture la courbure du sous-espace à partir d'informations locales puis en optimisant une composante globale. Tout d'abord, LLE identifie pour chaque point de données ses plus proches voisins. Ensuite, l'algorithme assigne un poids à chacun des voisins de manière qu'on puisse reconstruire le point actuel à partir d'une combinaison linéaire de ses voisins (avec la contrainte additionnelle que les poids locaux somment à 1). En partant des poids locaux, il est possible de calculer les k valeurs et

Algorithme 2 Isomap(D)

[Étape 1] Calculer la distance entre tous les points dans un voisinage fixe (tel qu'un ϵ-voisinage où tous les points situés à distance ϵ du point d'origine sont considérés comme des voisins).
Représenter ces distances sous forme d'un graphe G où il existe une arête entre deux points seulement s'ils sont dans le même voisinage et où le poids de cette arête est proportionnel à la distance entre ces deux points.
[Étape 2] Estimer la distance géodésique entre chaque paire de points en utilisant un algorithme pour trouver le plus court chemin sur le graphe G.
[Étape 3] Appliquer l'échelonnement multidimensionnel sur la matrice de distances générée à l'étape 2.

vecteurs propres qui globalement minimisent l'erreur de reconstruction.

La *catégorisation spectrale* [151] combine l'approche de la réduction de la dimensionnalité avec un algorithme de catégorisation. L'idée principale est de commencer d'appliquer une méthode de réduction de dimensionnalité (comme l'ACP) sur les données, avant d'utiliser un algorithme de catégorisation (tel que les k-moyennes) sur la nouvelle représentation des données obtenue. La phase de réduction de dimensionnalité accélère la partie catégorisation puisqu'elle réduit le nombre d'attributs à considérer. De plus, elle conduit souvent à la découverte de catégories plus stables, voulant dire que même sous des conditions initiales légèrement différentes les catégories retournées par l'algorithme seront sensiblement équivalentes.

3.3.3 Estimation de densité

L'estimation de densité est la forme la plus générale d'apprentissage non-supervisé car la plupart des autres tâches d'apprentissage peuvent se réduire à l'estimation de densité. Elle modélise la structure de la distribution sous-jacente aux données sous la forme d'une fonction de densité.

Exemple 3.7 (Estimation de densité). *Apprendre la structure musicale des morceaux d'un compositeur particulier, estimer la distribution de plusieurs espèces en fonction de différentes conditions environnementales, dériver un modèle permettant de prédire l'évolution du climat ou du cours de la bourse.*

Une manière générique de représenter une fonction de densité est sous la forme d'un *mélange de densités* $p(x|\theta) = \sum_{i=1}^{k} p(x|C_i, \theta_i)P(C_i)$ où k est le nombre de classes,

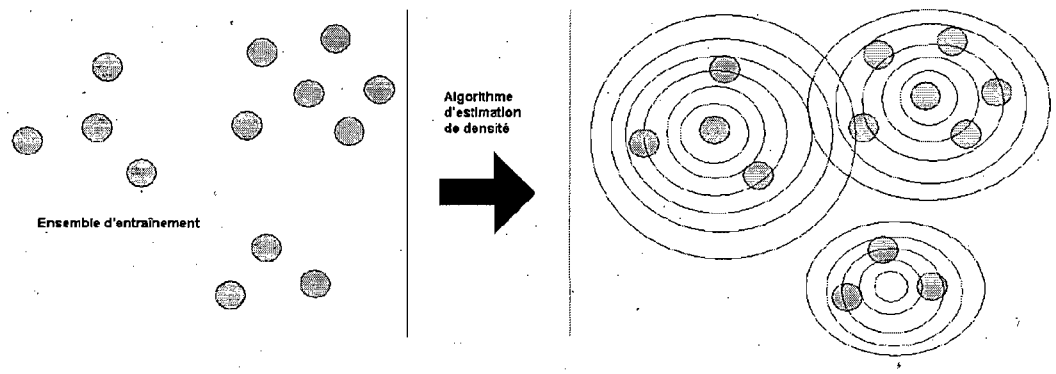

FIG. 3.4 – Illustration d'une tâche d'estimation de densité en utilisant un mélange de gaussiennes [64]. L'algorithme d'estimation de densité va apprendre un modèle représentant la distribution des données à partir de l'ensemble d'entraînement. Dans cet exemple particulier, l'ensemble de données sera résumé sous la forme d'un mélange composé de 3 gaussiennes différentes.

$\theta = (\theta_1, \ldots, \theta_k)$ un vecteur de paramètres, $p(x|C_i, \theta_i)$ la densité de la $i^{\text{ème}}$ composante du mélange et $P(C_i)$ la probabilité *a priori* de cette composante. Le but de l'estimation de densité peut se résumer à apprendre (estimer) le vecteur de paramètres θ et les probabilités *a priori* des composantes $\{P(C_1), \ldots, P(C_k)\}$ étant donné un ensemble de données $D_n = \{x_1, \ldots, x_n\}$. Un type de mélange souvent utilisé est celui des *mélanges de gaussiennes* [142] où $p(x \mid C_i, \theta_i)$ est alors une normale de moyenne μ_i et de matrice de covariance Σ_i. Le principal avantage des mélanges de gaussiennes est qu'ils constituent un approximateur universel des densités. Autrement dit, n'importe quelle distribution de probabilités peut être approximée par un mélange de gaussiennes.

Lorsqu'on utilise l'approche du *maximum de vraisemblance*, on modélise la vraisemblance d'un échantillon comme $p(D_n \mid \theta) = \prod_{j=1}^{n} p(x_j \mid \theta)$ et on cherche à maximiser une fonction de coût $l = \sum_{j=1}^{n} \log p(x_j \mid \theta)$, appelée *log vraisemblance*. Intuitivement, cette fonction de coût mesure à quel point un modèle spécifique semble vraisemblable par rapport aux données observées. Plus cette fonction de coût est élevée, plus le modèle appris est vraisemblable. D'autres fonctions de coût peuvent être utilisées comme le critère de *maximum de vraisemblance pénalisé* [80] ou encore le critère de la *taille minimale de la description* [102]. Le but reste toujours de trouver le modèle qui maximise (ou minimise

pour certains critères) cette fonction de coût. En pratique, il est souvent impossible de trouver la solution exacte car l'algorithme qui la calculerait pourrait demander un temps exponentiel par rapport à n, le nombre de points de données. L'*algorithme EM* [67] (pour *Expectation-Maximization* en anglais) est souvent utilisé pour contourner ce problème en calculant une approximation de la solution optimale. Il s'agit d'un algorithme d'optimisation itératif de type hill-climbing qui permet d'estimer un vecteur de paramètres θ. Il est possible que l'algorithme se retrouve coincé dans un maximum local. L'algorithme des k-moyennes présenté dans la section 3.3.1 peut être reformulé comme une variante de l'algorithme EM.

L'*approche bayésienne* est une autre approche pouvant être utilisée pour faire de l'estimation de densité. Contrairement à l'approche du maximum de vraisemblance, elle permet d'incorporer facilement les connaissances *a priori* dont on dispose sur les données, mais par contre elle est souvent plus gourmande en calculs. Pour les ensembles de données de grande taille, les deux approches convergent souvent vers le même résultat. Par contre, pour des jeux de données de taille moindre, les modèles retournés par les deux méthodes peuvent être sensiblement différents.

Dans le cas où les données qu'on modélise ont une structure très spécifique, il faut faire appel à des modèles adaptés à cette situation. Ainsi, dans le cas de données séquentielles évoluant au fil du temps, on peut par exemple les modéliser en utilisant des *chaînes de Markov cachées*. Dans ce cas-là aussi, les paramètres du modèle peuvent être appris en utilisant une variante de l'algorithme EM appelé l'*algorithme de Baum-Welch*. Enfin les *modèles graphiques* [121] tel que les *réseaux bayésiens* ou les *réseaux de croyance* servent eux à modéliser les dépendances conditionnelles entre plusieurs variables. L'apprentissage dans les modèles graphiques consiste aussi bien à apprendre la structure du graphe que les dépendances entre les différents paramètres. Pour plus de détails sur les différents modèles voir l'article de survol suivant [98].

3.4 Conclusion

Du point de vue théorique, l'étude de l'apprentissage machine permet de mieux cerner le fonctionnement, les mécanismes ansi que les limites de l'apprentissage chez la machine. Du côté pratique, l'apprentissage machine a eu énormément de succès et a trouvé des

applications dans des domaines aussi diversifiés que la vision, la musique, la prédiction financière, l'analyse du climat, l'astronomie ou encore la bio-informatique.

L'apprentissage supervisé convient parfaitement aux situations où on dispose d'exemples d'entrée/sortie souhaitées qui peuvent jouer le rôle d'ensemble d'entraînement alors que l'apprentissage non-supervisé est plus adapté à un mode exploratoire de l'apprentissage où on cherche à révéler la structure cachée à l'intérieur des données. Ces deux formes d'apprentissage sont cependant fortement interconnectées. Par exemple, il arrive souvent qu'un algorithme de réduction de dimensionnalité soit utilisé comme étape de pré-traitement avant de lancer un algorithme de classification afin d'accélérer celui-ci.

Il est important aussi de se rappeler que l'apprentissage machine n'a pas la prétention d'offrir un algorithme "ultime" mais plutôt un répertoire d'algorithmes ayant chacun des forces et des faiblesses différentes, parmi lesquels on peut piocher selon la situation d'apprentissage rencontrée.

CHAPITRE 4

TOUR D'HORIZON DE L'APPRENTISSAGE QUANTIQUE

4.1 Présentation de l'apprentissage quantique

L'*apprentissage quantique* (ou *quantum learning*[1] en anglais) est le domaine qui est né des différentes rencontres entre l'informatique quantique et l'apprentissage machine. Historiquement, le terme "quantum learning" est apparu pour la première fois dans le titre d'un article [57] publié en 1995 par Chrisley[2], un philosophe des sciences. La définition originale que Chrisley donne au terme est la suivante.

Définition 4.1 (Quantum learning, Chrisley 95). *Quantum computers that modify themselves in order to improve their performance in some way.*

Cette définition est relativement vague et laisse beaucoup de place à l'interprétation du lecteur quant au but ou à la méthode utilisée pour l'apprentissage. L'article original de Chrisley [57] (ainsi qu'un autre article qui suivit [58]) discutent de la possibilité d'avoir des ordinateurs quantiques qui modifient leurs paramètres et leurs comportements de manière adaptative suite à des interactions avec l'environnement, ce qui constitue une forme d'apprentissage. L'implémentation d'un tel ordinateur que Chrisley définit est celle d'une variante quantique des réseaux de neurones (cf. section 4.3.1). La façon dont les poids de ce réseau de neurones quantiques sont mis à jour est basée sur une analogie avec le fonctionnement de l'interférence dans les fentes d'Young. Le reste de la discussion de ses articles est principalement de teneur philosophique.

Bonner et Freivalds ont tenu à la fin des années 90 et au début des années 2000, une série de trois ateliers sur le thème du calcul et de l'apprentissage quantique (le nom exact en anglais était "*International Workshop on Quantum Computation and Learning*"). Dans les actes du dernier atelier (Riga 2002), ils ont écrit un survol[3] du domaine

[1] Attention car faire une requête sous le terme "quantum learning" sous Google peut donner l'impression qu'il s'agit d'une méthode pour améliorer ses compétences ou encore que l'expression se rapporte à un institut culinaire de Nouvelle-Zélande.

[2] Bien que publiées pour la première fois en 1995, les idées développées dans cet article ont apparemment été présentées lors d'un atelier intitulé "Brain, mind and Physics" en 1993 à Prague.

[3] Peut-être parce qu'il est écrit dans les actes de conférence d'un atelier, ce survol semble être passé

de l'apprentissage quantique [39]. De leur propre avis, le domaine ne semblait pas être encore mature à l'époque. Ainsi, quasiment tous les articles présentés au dernier atelier[4] portaient soit sur le thème du calcul quantique, soit sur celui de l'apprentissage, mais rarement sur les deux en même temps (exception faite de deux articles dus à Bonner et Freivalds [38, 39]) alors qu'un des buts avoués de l'atelier était justement de faire se rencontrer les deux communautés. Le nombre d'articles sur l'arXiv quant-ph[5] contenant le mot "learning" dans son titre était seulement de 5 à l'époque (soit en 2002) contre 14 aujourd'hui (1 juillet 2008). De même, la recherche sur le site CiteSeer^{x}[6] indique qu'il y avait 17 articles avec le terme "quantum learning" dans son résumé à la date de 2002 contre 102 maintenant (toutes ne sont pas pertinentes). Il semble donc que le domaine de l'apprentissage quantique grandisse lentement mais sûrement. La définition donnée par Bonner et Freivalds à l'apprentissage quantique est la suivante.

Définition 4.2 (Quantum learning, Bonner et Freivalds 01). *The quest of quantum learning ... is to produce tractable quantum algorithms in situations, where tractable classical learning situations do not exist, or are not known to exist.*

Cette définition énonce que le but principal de l'apprentissage quantique est de développer des algorithmes quantique d'apprentissage pour les situations où il n'existe pas (ou il ne semble pas exister) d'algorithmes classiques d'apprentissage efficaces. Parmi ces situations, on peut retrouver les travaux de Bonner et Freivalds sur les automates finis quantiques [38] qui prouvent qu'il existe des classes de fonctions totales récursives qui peuvent être appris par un automate fini quantique, mais pas par un automate fini classique, qu'il soit déterministe ou probabiliste.

L'apprentissage quantique a été le sujet d'un séminaire[7] donné par Meyer de l'uni-

---

inaperçu aussi bien auprès de la communauté d'informatique quantique que d'apprentissage machine. En effet, notre article intitulé "quantum clustering algorithms" [9] est le seul à citer ce survol.

[4] En lisant la note de l'éditeur des actes des deux premiers ateliers (QCL'99 et QCL'00), on peut se rendre compte que là aussi les articles concernaient seulement un des deux domaines. Les actes de QCL'99 et QCL'00 sont accessibles en ligne par les sites web des ateliers, soit respectivement `http://www.mdh.se/ima/personal/rbr01/courses/riga99.html` et `http://www.mdh.se/ima/personal/rbr01/courses/sundbyholm.html`.

[5] L'arXiv (`http://arxiv.org/`) est un répertoire public où il est possible de déposer des versions préliminaires d'articles pour qu'elles soient en libre accès. La section quant-ph (`http://arxiv.org/archive/quant-ph` est très populaire auprès de la communauté d'informatique quantique et de nombreux chercheurs, dont beaucoup parmi les plus renommés du domaine, y contribuent régulièrement.

[6] `http://citeseerx.ist.psu.edu/`

[7] `http://math.ucsd.edu/~dmeyer/teaching/seminar02fall.html`

versité de San Diego durant l'automne 2002. Le but de ce séminaire était d'explorer les connections entre l'informatique quantique et l'apprentissage machine, mais il semble avoir surtout focalisé sur la partie théorique de l'apprentissage machine (cf. section 4.2). L'expérience de conduire un séminaire sur l'apprentissage quantique ne semble pas avoir été renouvelée par Meyer depuis. Au meilleur de ma connaissance, la seule thèse de doctorat directement reliée à l'apprentissage quantique est due à Atıcı et s'intitule "Advances in quantum computational learning theory" [15]. Elle a été soutenue durant l'été 2006 à l'université de Columbia (New-York) et comme son titre l'indique elle porte aussi l'aspect théorique de l'apprentissage (section 4.2).

Au regard des conférences influentes en apprentissage machine, trois articles au moins ont été présentés à la conférence NIPS (*Neural Information Processing Systems*) : un sur l'*entraînement des réseaux de neurones quantiques* [163] (cf. section 4.3.1), un autre sur un *algorithme de catégorisation s'inspirant de la mécanique quantique* [116] (cf. section 4.4) et le dernier sur une *règle de Bayes pour mettre à jour notre connaissance des matrices densité* [185] (cf. section 4.7). À ICML (*International Conference of Machine Learning*), notre article portant sur les algorithmes quantiques de catégorisation [9] (chapitre 5) semble être le seul qui a un lien direct avec l'apprentissage quantique. Nous pouvons donc voir que le "quantique" ne semble pas être un thème de prédilection pour l'instant auprès de la communauté d'apprentissage machine.

Du côté de l'informatique quantique et de QIP (*Workshop on Quantum Information Processing*), qui est la principale rencontre annuelle du domaine, les sujets qui ont été présentés en rapport avec l'apprentissage quantique sont les *bornes en complexité de la communication quantique se basant sur des notions d'apprentissage machine* [97] (cf. section 4.5), l'*estimation de systèmes et processus quantiques* (cf. section 4.6), les *cryptosystèmes basés sur des problèmes d'apprentissage supposés difficiles même pour un ordinateur quantique* [162] (cf. section 4.8) et *apprendre à généraliser sur des POVMs* [1] (cf. section 4.9).

Au lieu de simplement se restreindre à utiliser l'informatique quantique pour améliorer l'apprentissage classique (comme le fait la définition 4.2), je suggère d'adopter une définition plus ouverte qui inclut toutes les interactions possibles entre les deux domaines, telles que les situations où la mécanique quantique sert d'inspiration à des algorithmes classiques d'apprentissage ou encore celles où on utilise des idées et des notions provenant

d'apprentissage machine pour résoudre des problèmes quantiques.

Définition 4.3 (Apprentissage quantique). *L'apprentissage quantique étudie toutes les interactions et rencontres possibles entre l'informatique quantique et l'apprentissage machine.*

Cette définition est volontairement large pour ne pas restreindre le domaine simplement à des variantes quantiques d'algorithmes d'apprentissage (définition 4.1) ou à chercher à améliorer sur ce qui classiquement est déjà possible en apprentissage (définition 4.2). Cette définition est illustrée dans ce chapitre par un tour d'horizon à travers les travaux d'apprentissage quantique suivants :

- résultats fondamentaux en théorie de l'apprentissage,
- variantes quantiques d'algorithmes d'apprentissage,
- algorithmes classiques d'apprentissage s'inspirant de la mécanique quantique,
- bornes en complexité de la communication quantique s'appuyant sur des notions d'apprentissage machine,
- estimation de processus et de systèmes quantiques,
- calcul bayésien pour les matrices densité,
- cryptosystèmes basés sur les problèmes d'apprentissage dit difficiles
- apprendre à généraliser sur des mesures.

Mon souhait est qu'un jour, il existe un atelier ou une conférence annuelle relié à l'apprentissage quantique, qui offrirait un endroit naturel où les deux communautés pourraient se rencontrer.

4.2 Résultats fondamentaux en théorie de l'apprentissage

Parmi les premières recherches en apprentissage quantique figurent des travaux menés dans le domaine de la *théorie calculatoire de l'apprentissage*[8] (nommée *computational learning theory* en anglais) afin de comparer l'apprentissage dans un contexte classique

[8]La théorie calculatoire de l'apprentissage peut être vue comme la contrepartie théorique de l'apprentissage machine. Alors que l'apprentissage machine adopte en généralement une vue plutôt empirique de l'apprentissage où un algorithme d'apprentissage doit être validé sur un ensemble de données réelles pour prouver son efficacité, la théorie calculatoire de l'apprentissage s'intéresse plutôt à démontrer des résultats théoriques dans des modèles d'apprentissage plus abstraits. La conférence annuelle CoLT est la conférence principale du domaine.

et dans un contexte quantique. Deux modèles ont été généralisés au monde quantique : le modèle d'*apprentissage PAC* (pour Probably Approximately Correct) de Valiant [181] et le modèle d'*apprentissage exact à partir de requêtes* de Angluin [13].

Dans ces modèles, le but est d'inférer une fonction f qui est calculée par une boîte noire/un oracle en faisant des requêtes sur celle-ci. La complexité de l'apprentissage est généralement définie par rapport aux nombres de requêtes nécessaires pour apprendre (approximativement ou exactement) la fonction f. La principale différence entre les versions classiques et quantiques de ces modèles est que dans les variantes quantiques, nous pouvons interroger l'oracle en utilisant une superposition de questions au lieu d'une seule question à la fois. Ainsi, ce type de modèle est très proche du modèle de boîte noire considéré habituellement en informatique quantique (cf. section 5.3.1). La principale différence est qu'on souhaite ici apprendre (approximativement ou exactement) une fonction f alors qu'en informatique quantique, on cherche d'habitude plutôt à déterminer une des propriétés de la fonction (comme constante ou balancée pour l'algorithme de Deutsch-Jozsa [70] ou une entrée x particulière tel que $f(x) = 1$ pour Grover [103]).

4.2.1 Apprentissage PAC

Dans le modèle PAC [181], l'algorithme d'apprentissage a accès à un oracle O_{PAC}^{cl} qui ne prend pas de valeur en entrée mais qui à chaque fois qu'il est invoqué produit une paire $(x, f(x))$ choisie aléatoirement selon une certaine distribution de probabilités p_n sur les entrées $x \in \{0,1\}^n$, pour $n \geq 1$. Typiquement, f est une fonction booléenne choisie parmi une classe de fonctions possibles F. Dans le modèle PAC, le but est de produire une hypothèse qui *approxime* la fonction calculée par l'oracle.

Définition 4.4 (Apprentissage PAC classique). *Un algorithme classique A est un algorithme d'apprentissage permettant d'apprendre une classe de fonctions F dans le sens PAC du terme, si pour toutes les distributions de probabilités p et pour tout $n \geq 1$, $0 < \epsilon$, $\delta < 1$, $f \in F_n$, si A connaissant n, ϵ et δ et ayant accès à l'oracle O_{PAC}^{cl}, peut produire avec probabilité au moins $1 - \delta$, une hypothèse h tel que la probabilité que $h(x) \neq f(x)$ sous la distribution de probabilités p soit au plus ϵ. La complexité classique en terme d'appels à O_{PAC}^{cl} pour que A puisse apprendre f dans le modèle PAC classique est appelée $T_{PAC}^{cl}(n, \epsilon, \delta)$.*

Typiquement, la complexité $T^{cl}_{PAC}(n, \epsilon, \delta)$ pour une fonction particulière représente le nombre de requêtes que l'algorithme d'apprentissage va devoir demander *en pire cas* à l'oracle avant de pouvoir avec probabilité non-négligeable approximer la fonction calculée par celui-ci. Pour certaines fonctions, cette complexité va être polynomiale dans la taille de l'entrée, alors que pour d'autres comme une fonction totalement aléatoire, elle va être de $O(2^n)$.

La variante quantique du modèle PAC est due originalement à Bshouty et Jackson [52]. Leur motivation était de définir une extension quantique d'un oracle calculant une forme normale disjonctive (ou *Disjunctive Normal Form* (DNF) en anglais), problème pour lequel il n'existe pas d'algorithme classique PAC efficace, afin de prouver ensuite que quantiquement cette classe de fonctions DNF pouvait être apprise sous la distribution uniforme à partir de cet oracle O^{qu}_{PAC}. Dans ce modèle, l'action de l'oracle O^{qu}_{PAC} consiste à prendre comme entrée un registre quantique qui a la forme $|0^{n+1}\rangle$ et de retourner en sortie la superposition $\sum_{x\in\{0,1\}^n} \sqrt{p(x)}\,|x, f(x)\rangle$, où $\sqrt{p(x)}$ constitue l'amplitude d'une paire d'entrée/sortie particulière $|x, f(x)\rangle$. Formellement, nous pouvons définir l'apprentissage quantique dans le modèle PAC de la manière suivante.

Définition 4.5 (Apprentissage PAC quantique [169]). *Un algorithme d'apprentissage quantique A pour F dans le sens PAC du terme est une famille de circuits quantiques indexés par $n \geq 1$ et $0 < \epsilon, \delta < 1$, tel que pour tout $f \in F_n$ en ayant accès à O^{qu}_{PAC}, A va pouvoir apprendre, avec probabilité au moins $1 - \delta$, une hypothèse classique h telle que la probabilité que $h(x) \neq f(x)$ sous la distribution de probabilité p soit au plus ϵ. La complexité quantique en terme de requêtes pour que A puisse apprendre f dans le modèle PAC quantique est appelée $T^{qu}_{PAC}(n, \epsilon, \delta)$.*

Comme l'énonce le lemme suivant, il est trivial à partir de l'oracle quantique O^{qu}_{PAC} de simuler l'oracle classique O^{cl}_{PAC}.

Lemme 4.1 (Simulation de l'oracle PAC classique à partir de l'oracle quantique). *Le nombre de requêtes quantiques $T^{qu}_{PAC}(n, \epsilon, \delta)$ nécessaires pour apprendre une fonction est plus petit ou égal au nombre de requêtes classiques nécessaires $T^{cl}_{PAC}(n, \epsilon, \delta)$.*

Démonstration. Soit l'oracle quantique O^{qu}_{PAC} qui étant donné un registre initialisé à $|0^{n+1}\rangle$ produit en sortie la superposition $\sum_{x\in\{0,1\}^n} \sqrt{p(x)}\,|x, f(x)\rangle$. Pour simuler O^{cl}_{PAC},

il suffit de mesurer directement ce registre, ce qui conduit à observer une paire $(x, f(x))$ choisie selon la distribution de probabilités $p(x)$. Une fois cette simulation effectuée, il suffit d'exécuter l'algorithme classique permettant d'apprendre la fonction dans le sens PAC. □

Servedio et Gortler [169] ont caractérisé quel est l'avantage que peut procurer pour apprendre une fonction dans le sens PAC du terme d'avoir accès à un oracle quantique comparativement à un oracle classique. Le théorème suivant formalise les résultats de leurs recherches pour le modèle PAC.

Théorème 4.1 (Relation entre l'apprentissage PAC classique et quantique [169]). *Soit une classe de fonctions F qui peut être apprise avec $T^{qu}_{PAC}(n, \epsilon, \delta)$ requêtes à un oracle quantique O^{qu}_{PAC}, alors cette même classe de fonctions peut être apprise en faisant $T^{cl}_{PAC}(n, \epsilon, \delta) \in O(nT^{qu}_{PAC}(n, \epsilon, \delta))$ requêtes à la version classique de l'oracle O^{cl}_{PAC}.*

Ce théorème démontre qu'il existe une relation linéaire en terme du nombre de requêtes entre la version quantique et la version classique de l'apprentissage PAC. Cependant, cette relation n'exclut pas que pour certains problèmes le nombre de requêtes quantiques nécessaires puisse être constant alors que classiquement il faudrait un nombre linéaire de requêtes. Il est important aussi de réaliser que ce théorème *énonce la quantité d'information* (mesurée par le nombre de requêtes) qu'il est nécessaire de demander à la boîte noire, classiquement et quantiquement, mais qu'il *ne dit rien par rapport au temps de calcul nécessaire pour mener à bien cet apprentissage*. Ainsi, Servedio et Gortler [169] ont prouvé que sous certaines hypothèses cryptographiques (comme la difficulté de factoriser les entiers de Blum), il était possible d'apprendre efficacement (en temps polynomial) quantiquement une certaine classe de concepts (grâce à l'algorithme de Shor [172]) mais pas classiquement[9].

4.2.2 Apprentissage exact

Dans le modèle d'apprentissage exact à partir de requêtes [13], le but est découvrir *exactement* la fonction calculée par l'oracle O^{cl}_{exact}. Par contraste avec l'apprentissage PAC, dans ce modèle on peut spécifier l'entrée x désirée à l'oracle O^{cl}_{exact} qui produit

[9] À moins qu'un algorithme classique efficace de factorisation soit découvert dans le futur

alors $f(x)$. Le modèle d'apprentissage exact est plus puissant que le modèle PAC, puisqu'il permet de choisir exactement l'entrée de l'oracle O^{cl}_{exact} et donc la paire $(x, f(x))$ qu'on va observer, alors que dans le modèle PAC cette paire est choisie aléatoirement par l'oracle O^{cl}_{PAC} selon la distribution de probabilités p_n.

Définition 4.6 (Apprentissage exact classique). *Un algorithme classique A est un algorithme d'apprentissage permettant d'apprendre F dans le sens exact du terme, si pour tout $n \geq 1$, $f \in F_n$, si A connaissant n et ayant accès à l'oracle O^{cl}_{exact}, peut produire avec probabilité au moins $\frac{2}{3}$, une hypothèse h tel que $h(x) = f(x)$ pour toutes les entrées $x \in \{0,1\}^n$. La complexité classique en terme d'appels à O^{cl}_{exact} pour que A puisse apprendre f dans le modèle d'apprentissage exact à partir de requêtes est appelée $T^{cl}_{exact}(n)$.*

Servedio et Gortler [169] ont défini la généralisation quantique de ce modèle. La variante quantique de l'oracle pour l'apprentissage exact $O^{cl}_{quantique}$ prend en entrée une superposition d'états $\sum_{x\in\{0,1\}^n} \alpha_x |x, b\rangle$ et produit la superposition correspondante de sorties $\sum_{x\in\{0,1\}^n} \alpha_x |x, b \oplus f(x)\rangle$. Cette description de l'oracle correspond exactement à la manière standard de décrire l'action d'une boîte noire en algorithmique quantique (cf. section 5.3.1).

Définition 4.7 (Apprentissage exact quantique [169]). *Un algorithme d'apprentissage quantique A pour F dans le sens exact du terme est une famille de circuits quantiques indexés par $n \geq 1$ et qui a une architecture indépendante de la fonction f particulière choisie, tel que pour tout $f \in F_n$, en ayant accès à O^{qu}_{exact}, A va pouvoir apprendre, avec probabilité au moins $\frac{2}{3}$, une hypothèse classique h tel que $h(x) = f(x)$ pour toutes les entrées $x \in \{0,1\}^n$. La complexité quantique en terme de requêtes pour que A puisse apprendre f dans le modèle exact à partir de requêtes est appelée $T^{qu}_{exact}(n)$.*

Théorème 4.2 (Relation entre l'apprentissage exact classique et quantique [169]). *Soit une classe de fonctions F qui peut être apprise avec $T^{qu}_{exact}(n)$ requêtes à un oracle quantique O^{qu}_{exact}, alors cette même classe de fonctions peut être apprise en faisant $T^{cl}_{exact}(n) \in O(nT^{qu}_{exact}(n))^3$ requêtes à la version classique de l'oracle O^{cl}_{exact}.*

Ce résultat démontre que tout comme dans le modèle PAC, l'apprentissage classique et quantique dans le modèle d'apprentissage exact à partir de requêtes sont polynomialement reliés en terme du nombre du requêtes. Nous pouvons remarquer cependant que

ce théorème n'exclut pas que la complexité en terme de requêtes puisse être 1 dans le cas quantique contre n classiquement (ce qui est le cas par exemple pour l'algorithme de Deustch-Jozsa [70]). De plus, Servedio et Gortler ont construit une classe de concept qui peut être apprise quantiquement en temps polynomial dans le modèle d'apprentissage exact à partir de requêtes mais pas classiquement sous l'hypothèse qu'il existe des fonctions à sens unique. Leur construction se base sur une extension d'un algorithme quantique dû à Simon [175].

4.3 Variantes quantiques d'algorithmes d'apprentissage

Pour n'importe quel algorithme d'apprentissage (supervisé ou non-supervisé), il est naturel d'essayer d'imaginer à quoi ressemble un analogue quantique de cet algorithme et de se questionner sur les avantages que cette variante quantique peut offrir sur sa contrepartie classique. En apprentissage quantique, le modèle où le plus de travaux ont été réalisés dans ce sens est celui des *réseaux de neurones quantiques* (section 4.3.1).

Définition 4.8 (Quantisation). *Processus qui part d'un protocole ou algorithme classique et qui le convertit (partiellement ou totalement) en un algorithme quantique afin d'en améliorer les performances, par exemple en le rendant plus rapide dans le cas d'un algorithme ou en économisant sur le coût de communication dans le cas d'un protocole distribué.*

À noter qu'en anglais, quantisation se traduit par "quantization" qui peut prendre d'autres sens que celui évoquer dans la définition, comme diviser un espace continu en morceaux discrets (qui se traduirait en français par *quantification*). Il existe des versions quantisées des machines à vecteurs de support [14], de certains algorithmes d'apprentissage par renforcement [76] ainsi que d'algorithmes d'apprentissage non-supervisé (chapitre 5).

Anguita, Ridella, Riviecco et Zunino [14] ont proposé un algorithme quantique permettant d'accélérer l'entraînement des machines à vecteurs de support [182]. Leur méthode se base sur l'algorithme pour trouver le minimum d'une fonction de Dürr et Høyer [78] afin d'accélérer l'entraînement de la machine à vecteurs de support. L'entraînement de la machine à vecteurs de support peut s'exprimer comme un problème

d'optimisation quadratique et l'algorithme de Dürr et Høyer est utilisée comme sous-routine à l'intérieur de l'algorithme d'optimisation. Au niveau de l'apprentissage par renforcement, Dong, Chen et Chen ont défini une variante quantique de l'apprentissage par renforcement [76], dans laquelle le robot maintient un état quantique interne qui évolue au fur et à mesure des interactions avec l'environnement et qui lui sert à décider quelle action accomplir en fonction de quelle situation[10]. Un champ de recherche qui semble prendre de l'essor en apprentissage quantique depuis quelques années est celui où la machine qui réalise l'apprentissage est quantique et elle apprend à réaliser une certaine tâche de nature quantique elle-aussi (ce qui correspond à la définition 4.1 de Chrisley). Par exemple, deux articles récents [16, 95] parus sur l'arXiv et qui ont l'expression "quantum learning" dans leurs titres tombent dans cette catégorie.

4.3.1 Réseaux de neurones quantiques

Un *réseau de neurones artificiels* [36] est un algorithme d'apprentissage s'inspirant du fonctionnement des réseaux de neurones qui se trouvent dans le cerveau humain. Plusieurs variétés de réseaux de neurones existent mais typiquement leur architecture globale partagent les mêmes caractéristiques générales. Un réseau de neurones artificiels se compose de plusieurs couches de neurones artificiels reliés entre eux par des liens comportant des poids, appelés *synapses* comme leur contrepartie biologique. Chaque neurone est une petite unité de calcul qui prend en entrée les valeurs renvoyées par les neurones de la couche précédente pondérées par les poids des synapses, applique une fonction sur les valeurs d'entrée (par exemple la fonction *sigmoïde* ou la fonction *tanh*) et retourne en sortie la valeur résultante. Parmi les couches du neurones figurent au moins la *couche d'entrée* et la *couche de sortie*, et souvent aussi une couche intermédiaire appelée *couche cachée*. Un des résultats importants concernant les réseaux de neurones est le *théorème de l'approximation universelle* [90] qui précise qu'il est possible d'approximer de manière uniforme n'importe quelle fonction continue sur les nombres réels en utilisant un réseaux de neurones avec une seule couche cachée[11].

[10] J'avoue de mon côté être assez sceptique quant à leur approche qui semble volontairement floue sur certains détails.

[11] Cependant, il se peut que le nombre de neurones nécessaires dans la couche cachée pour réaliser cette approximation soit exponentiel par rapport au nombre de neurones de la couche d'entrée. De plus, ce théorème d'approximation ne donne aucune garantie en terme de généralisation. En se basant sur les

Beaucoup de propositions ont été faites durant ces quinze dernières années afin d'imaginer la contrepartie quantique des réseaux de neurones[12]. Aucun consensus ne semble pourtant se détacher pour l'instant et les modèles proposés sont très différents les uns des autres et semblent incomparables. La première difficulté provient de trouver une définition sur ce qu'est exactement un réseau de neurones. En effet, un réseau de neurones quantique pourrait être :

- une représentation quantique d'un réseau de neurones classiques,
- un type de POVM permettant de classifier de l'information quantique,
- un algorithme quantique pour entraîner plus rapidement un réseau de neurones classique

Ezhov et Berman ont écrit un livre d'introduction sur le sujet [83] intitulé "*Introduction to quantum neural technologies*" qui recense et discute certains des modèles proposés. Un des modèles décrits dans ce livre représente un réseau de neurones quantiques sous la forme d'un registre quantique qui serait en superposition des différentes entrées x_i possibles du réseau et des sorties y_i correspondantes à ces entrées. Lors de la classification d'un nouvel état inconnu $x_?$, le POVM qui va être appliqué sur le circuit va dépendre directement de $x_?$ et va produire en sortie $y_?$ le motif associé à cette entrée. L'espoir est que si le "réseau" est bien conçu, il puisse retrouver même sur une entrée bruitée non vue auparavant la sortie correspondante (ce qui constitue une forme de généralisation). Les auteurs interprètent ce modèle comme un neurone unique qui serait en superposition de plusieurs entrées/sorties. L'avantage du modèle est qu'il permet de représenter un réseau de neurones avec un nombre exponentiellement plus faible de ressources que classiquement mais l'inconvénient est que la phase d'"entraînement" où il s'agit de mettre en place la superposition dans le registre quantique peut elle-même demander un temps exponentiel. Une autre application discutée dans le livre est celle où les réseaux de neurones quantiques pourraient être utilisés pour implémenter une approximation de n'importe quel fonction réalisée par un circuit quantique, un peu de la même manière que l'approximation universelle réalisée par les réseaux classiques.

---

travaux réalisés ces dernières années dans le contexte des *réseaux profonds* [113], qui comportent plusieurs couches cachées, il semble au contraire qu'un réseau de neurones à plusieurs couches, s'il est correctement entraîné, a un meilleur pouvoir de généralisation que les réseaux à une seule couche. Une question ouverte intéressante est de développer une variante quantique de ces réseaux profonds.

[12] Une manière de s'en convaincre est de faire une requête sur "*quantum neural network*" sous Google.

Un article de Ricks et Ventura discute le problème de l'entraînement [163] dans un modèle de réseau quantique où chaque noeud du réseau ainsi que chaque synapse serait représenté sous la forme d'un registre quantique spécifique. Pour la phase d'entraînement, les auteurs proposent une méthode inspirée de Grover pour initialiser le poids des synapses. Un des problèmes de leur méthode est qu'elle aussi requiert un temps exponentiel dans sa version exacte[13]. Les auteurs suggèrent une version probabiliste de l'entraînement qui semble plus efficace d'après les simulations effectuées mais pour laquelle ils n'offrent aucune garantie théorique quant à la vitesse de convergence. Cependant, leur article a le mérite d'être l'un des seuls qui fait une comparaison empirique entre la performance d'un réseau de neurones quantiques et d'un réseau de neurones classique sur un ensemble de données réel.

4.4 Algorithmes de catégorisation s'inspirant de la mécanique quantique

La physique sert souvent d'inspiration à des algorithmes d'intelligence artificielle ou d'apprentissage machine. Horn et Gottlieb ont inventé un algorithme de catégorisation qui puise son inspiration dans la mécanique quantique [116, 117]. Leur algorithme relie deux problèmes qui sont *a priori* très différents : celui de la catégorisation et celui de l'équation de Schrödinger. L'équation de Schrödinger est communément utilisée en mécanique quantique pour décrire l'état d'un système. Les solutions de cette équation (aussi appelées *eigenfonctions*) ont souvent un sens physique réel et intéressent au plus haut point les physiciens.

Dans leurs papiers, les auteurs s'intéressent au cas où l'on cherche à représenter les points de données comme ayant été générés par une distribution de noyaux gaussiens $\psi(x) = \sum_{i=1}^{n} e^{\frac{-(x-x_i)^2}{2\sigma^2}}$, où n correspond au nombre de points de données dans un espace euclidien de dimension d, x_i est le vecteur représentant le $i^{\text{ème}}$ centre de la gaussienne et $\psi(x)$ est la distribution de probabilités. L'algorithme de catégorisation de Horn et Gottlieb reformule $\psi(x)$ comme étant une composante d'une équation de Schrödinger l'état d'un système dans un espace de Hilbert. La recherche des minima de cette équation permet indirectement de déterminer le centre des catégories. L'algorithme possède un seul

[13]Le problème de l'*initialisation d'une superposition arbitraire* dans un registre quantique est un problème difficile pour lequel il n'existe pas d'algorithme générique efficace.

paramètre qui est σ, la largeur des noyaux gaussiens. En faisant varier ce paramètre, il est possible de diminuer ou d'augmenter le nombre de catégories trouvées. Une fois que les centres des catégories ont été identifiés, une descente de gradient permet de déterminer à quelles catégories appartiennent chacun des points de données. En utilisant des techniques provenant de l'analyse en composantes principales, l'algorithme peut devenir indépendant de d, le nombre de dimensions. Dans ce cas-là, le temps de calcul est de $\Theta(n^2)$ car l'équation de Schrödinger est évaluée uniquement sur les points de données.

Horn et Gottlieb ont testé leur approche sur deux ensembles de données et rapportent des résultats convaincants, en particulier comparé à une méthode plus classique se basant sur un estimateur de fenêtre de Parzen [155]. Leur algorithme semble particulièrement approprié pour faire de la catégorisation en tirant parti de l'information géométrique. On retrouve exprimé dans l'équation de Schrödinger deux effets inhérents aussi à la catégorisation qui ont une action opposée l'un à l'autre sur la fonction d'onde. Le potentiel représente la force attractive qui essaye de concentrer la distribution autour de son minimum (intra-similarité des catégories) alors que le Laplacien au contraire va contribuer à éparpiller la distribution (inter-dissimilarité des catégories).

4.5 Bornes en complexité de la communication quantique s'appuyant sur des notions d'apprentissage machine

Les *empreintes quantiques* [53] (appelées *quantum fingerprinting* en anglais) sont une technique développée en complexité de la communication quantique [43, 189] qui permet de calculer l'égalité de manière distribuée efficacement. Supposons qu'Alice et Bob possèdent chacun une chaîne de bits de longueur n, soit $x, y \in \{0,1\}^n$, et qu'Alice veut savoir si sa chaîne est identique à celle de Bob. Dans ce cas-là, Alice peut décider d'encoder sa chaîne de bits x dans une empreinte quantique de taille $O(\log n)$ qubits et l'envoyer à Bob. Celui-ci peut ensuite encoder sa propre chaîne y dans une empreinte et utiliser le Control-Swap test pour déterminer si les deux empreintes sont similaires ou non (voir section 6.1.4 pour plus de détails). Si les deux empreintes sont différentes (autrement dit $x \neq y$), ce test a une certaine probabilité non négligeable de le détecter. Ainsi, si Alice et Bob répètent cette procédure un nombre constant de fois et qu'ils observent toujours que les deux empreintes sont identiques, ils peuvent être sûr avec

une bonne probabilité que $x = y$. Classiquement, le même problème requiert un coût de communication de $\Omega(\sqrt{n})$ [150] dans le modèle où Alice et Bob ne sont pas autorisées à partager des variables aléatoires à l'avance.

Gavinsky, Kempe et de Wolf [97] ont cherché à caractériser les situations distribuées où développer un protocole basé sur les empreintes quantiques permet d'économiser comparé au coût de communication classique. Ils ont trouvés une connection intéressante entre les empreintes quantiques et la *marge*, un concept provenant de l'apprentissage machine classique. Intuitivement, la marge peut être vue comme la distance entre un point de données et la surface de décision d'un classifieur. Avoir une marge importante conduit généralement à une bonne généralisation et certains classifieurs comme les machines à vecteurs de support [182] cherchent explicitement à maximiser la marge durant leur apprentissage (voir section 3.2.3 pour un peu plus de détails). La connection entre la marge et les empreintes quantiques fonctionne dans les deux sens. Ainsi, des protocoles d'apprentissage performant pour un problème d'apprentissage particulier impliquent des bornes inférieures sur la marge réalisée pour ce problème, qui impliquent en retour des bornes supérieures sur les protocoles utilisant les empreintes quantiques pour résoudre ce problème. De la même manière, trouver un protocole distribué utilisant les empreintes quantiques pour un certain problème implique qu'il existe une classifieur avec une marge importante pour ce même problème. Le même type de résultat connectant la marge et la complexité de communication a été trouvé indépendamment par Linial et Shraibman [137].

4.6 Estimation de systèmes et de processus quantiques

La *tomographie* est un procédé qui, à partir d'un nombre de copies identiquement préparées d'un état quantique, permet d'identifier de façon unique cet état grâce à une série de mesures. Si le nombre de copies dont on dispose est infini, l'état peut être parfaitement identifié. En pratique, cependant cette hypothèse n'est jamais vérifiée et le nombre de copies est toujours fini,ce qui conduit à un estimé de l'état qui sera imparfait. Ainsi, la tomographie permet d'obtenir une connaissance complète de l'état mais pour obtenir un estimé fiable, elle requiert un nombre de copies qui grandit linéairement avec la dimension de l'espace de Hilbert si l'état est pur, et quadratiquement si l'état est

mélangé, donc exponentiellement dans les deux cas avec le nombre de qubits servant à décrire l'état. L'*estimation de la matrice densité* d'un état pur ou d'un état mélangé peut être réalisé grâce à une tomographie de cet état.

Exemple 4.1 (Tomographie d'un état pur). *Supposons un état quantique inconnu* $|\psi\rangle$ *sur lequel on n'a aucune connaissance a priori si ce n'est qu'il s'agit d'un état pur dont toutes les copies ont été préparées de la même manière*[14]. *Si cet état est défini sur d qubits, la dimension de l'espace de Hilbert dans lequel il vit sera de* 2^d *et l'état pourra être décrit par le même nombre d'amplitudes. Chaque amplitude étant un nombre complexe pouvant être représenté par deux réels, il faudra de l'ordre de* $\Theta(2^{d+1})$ *mesures différentes pour estimer ces amplitudes avec une bonne précision.*

Exemple 4.2 (Tomographie d'un état mélangé). *Supposons que l'état quantique inconnu* ρ *qu'on cherche à estimer est un mélange statistiques (et non pas un état pur) qui est défini sur d qubits. Cet état* ρ *peut être représenté par une matrice densité de taille* 4^d *où chaque entrée est une valeur réelle positive. Il faudra donc de l'ordre de* $\Theta(4^d)$ *mesures différentes pour pouvoir caractériser précisément cet état.*

De part leur nature, l'estimation d'états quantiques et l'estimation de processus quantiques, tel que la caractérisation de portes ou de canaux quantiques, sont intrinsèquement reliés. Ainsi lorsqu'on estime un état quantique, on fait souvent une série de mesures sur cette état, pouvant être implémentées par une transformation unitaire suivi d'une mesure dans la base de calcul. De la même façon, lorsqu'on a devant soi un canal quantique inconnu ξ, il est possible d'estimer son action sur un qubit le traversant à travers des observations faites sur des paires d'entrée/sortie du canal. Une technique consiste par exemple à utiliser un ensemble d'états linéairement indépendants qui formeront une base vectorielle de tous les opérateurs physiques possibles. Le problème avec cette deuxième technique est qu'elle requiert l'utilisation de d^2 états de test différents, où d est la dimension de l'espace de Hilbert considéré[15]. Si jamais les données collectées sont *incomplètes* (parce qu'on ne dispose pas de tous les d^2 états de test différents) ou *incohérentes* (parce

[14] De plus, on suppose qu'on est dans un modèle sans bruit, où toutes les opérations sur le système peuvent être réalisées de manière parfaite.

[15] Une autre façon de procéder est d'utiliser une paire d'états intriqués (comme $|\Psi^-\rangle$), dont on fait passer la moitié dans le canal et avant de réaliser ensuite une tomographie complète sur les deux moitiés de la paire.

que on n'a pas collecté un nombre suffisants de statistiques ou que du bruit vienne affecter le processus d'estimation), il est possible que la méthode de reconstruction caractérise l'action du canal ξ comme ne correspondant pas à une opération physique valide. La même situation peut survenir lors de l'estimation d'états ou portes quantiques.

Plusieurs familles de méthodes d'estimation existent, chacune se basant sur approche très différente du problème, et pouvant être pertinente dans des situations différentes. La plupart de ces méthodes sont aussi utilisés en apprentissage machine classique pour la tâche d'estimation de densité où le but est d'apprendre une fonction de densité qui modélise les données observées.

L'approche du *maximum de vraisemblance* se base sur le principe que le meilleur estimé est celui qui maximise la probabilité des données observées. La vraisemblance d'un modèle est défini comme la probabilité que le modèle aurait pu prédire les données observées avant même que les mesures n'aient été effectuées. En utilisant le logarithme de la vraisemblance comme critère d'optimisation, on obtient une fonction convexe à optimiser qui admet un seul maximum global. Un des défauts de cette méthode est qu'elle peut retourner des valeurs propres qui sont négatives ce qui physiquement ne correspond à aucune opération valide. De plus, l'approche par maximum de vraisemblance assigne toujours une probabilité zéro à tous les événements non-observées. Ziman, Plesch, Bužek et Štelmachovič [199] ont proposé une autre solution à ce problème d'estimation faisant appel au principe du *maximum de vraisemblance*. Leur idée est d'essayer de trouver parmi toutes les représentations possibles du canal correspondant à des actions physiquement réalisables, celle qui approxime le mieux les données observées. Autrement dit, cela revient à chercher le modèle qui est le plus vraisemblable par rapport aux observations faites comme on le fait couramment en estimation de densité.

D'autres approches sont possibles, elles aussi communément utilisées en apprentissage machine classique dans le cadre de l'estimation de densité, comme l'approche *bayésienne* ou le principe d'entropie maximale [54].

4.7 Calcul bayésien pour les matrices densité

Warmuth, qui est à l'origine principalement chercheur en théorie calculatoire de l'apprentissage, est littéralement tombé amoureux[16] du formalisme des matrices densité (cf. section 2.3.4) et a développé un calcul bayésien [185, 186] les concernant (appelé en anglais *Bayesian probability calculus for density matrices*). Son calcul peut être vu comme une généralisation des distributions de probabilités habituellement utilisée en apprentissage machine, où chaque événement correspond maintenant à un état quantique. Sans rentrer dans les détails, la règle de Bayes définie dans son calcul permet de mettre à jour notre connaissance *a priori* d'un système quantique (représentée sous forme d'une matrice densité) suite au résultat d'une mesure effectuée sur ce système. Cette règle est une généralisation de la règle de Bayes traditionnelle (qui est retrouvée comme cas particulier lorsqu'on se restreint à des états classiques). Elle a comme caractéristique principale de maintenir l'incertitude dans la direction de variance maximum de la matrice densité. Bien que Warmuth cherche encore à interpréter les connections entre son formalisme de calcul et l'informatique quantique, il l'a utilisé pour développer des applications pour l'apprentissage machine classique tel qu'une version en ligne de l'analyse en composantes principales [187]. Ainsi, on pourrait voir les travaux de Warmuth comme des algorithmes d'apprentissage puisant leur inspiration dans l'informatique quantique, un peu comme l'algorithme de catégorisation présentée dans la section 4.4.

4.8 Cryptosystèmes basés sur des problèmes d'apprentissage difficiles

L'existence même de l'algorithme de Shor [172] menace beaucoup de cryptosystèmes classiques, tel que RSA [164], qui sont basés sur la difficulté classique de résoudre un certain problème (comme la factorisation de deux grands nombres premiers ou le logarithme discret). Une question naturelle à se demander est quels sont les cryptosystèmes pour lesquels on ne connaît par encore d'algorithmes quantiques efficaces pouvant les briser et qui pourrait donc potentiellement résister à la "vague quantique". En théorie calculatoire de l'apprentissage (classique), Kearns et Valiant avaient déjà réalisé en 1994 que certaines classes de concepts étaient difficiles à apprendre dans le modèle PAC sous

[16] Extrait et traduit d'une communication personnelle avec Warmuth.

certaines hypothèses cryptographiques [127]. Pour être plus précis, ils ont démontré que s'il est possible d'apprendre efficacement ces classes de concepts alors il est possible de briser certaines hypothèses cryptographiques en temps polynomial. Le dual de cela est de développer un cryptosystème qui fonde sa sécurité sur la difficulté d'un certain problème d'apprentissage. Regev a mis au point un cryptosystème [162] qui se base sur la difficulté dans le pire cas d'une situation d'apprentissage reliée au plus court vecteur et aux plus courts vecteurs indépendants dans un treillis. Ce cryptosystème est totalement classique en soit, mais il base sa sécurité sur un problème considéré difficile, même à approximer, pour un ordinateur quantique. En effet, la preuve de sécurité de ce cryptosystème utilise une réduction quantique entre deux problèmes, et pouvoir le briser implique automatiquement avoir un algorithme quantique efficace pour le plus court vecteur et les plus courts vecteurs indépendants dans un treillis. En se basant sur les travaux de Regev, Klivans et Sherstov [130] ont établis la difficulté d'apprendre efficacement certains familles de circuits arithmétiques de taille polynomial et de profondeur 2 (dont une variante des réseaux de neurones utilisant des portes du type majorité) à moins qu'il existe un algorithme quantique efficace pour briser le cryptosystème de Regev.

4.9 Apprendre à généraliser sur des mesures

Aaronson a défini l'équivalent de l'apprentissage dans un contexte où nous avons accès à un nombre fini de copies d'un état quantique inconnu $\rho_?$, et nous voulons prédire comment cet état réagira lorsqu'il est mesuré par des POVMs choisis selon une distribution fixe, mais possiblement inconnue et continue [1]. Contrairement à la situation d'apprentissage considérée dans le chapitre 6, l'ensemble d'entraînement D_n n'est donc pas constitué d'états quantiques mais plutôt de n POVMs, et est décrit par un ensemble $D_n = \{E_1, \ldots, E_n\}$, où chaque E_i est un POVM. La question posée par Aaronson est similaire en esprit à celle de l'apprentissage PAC (section 4.2.1) et consiste à se demander sur combien de POVMs nous devons testé $\rho_?$ (autrement dit qu'elle doit être la taille minimum de D_n) avant de produire une hypothèse qui nous permet de prédire avec une bonne précision comment $\rho_?$ va réagir sur les autres POVMs que nous n'avons pas observé. Là encore, si nous pouvions faire une tomographie de l'état inconnu $\rho_?$, nous aurions une connaissance complète de l'état et il devient possible de prédire exactement

son comportement sur n'importe quel POVM. Aaronson a prouvé qu'à toute fin pratique il suffit d'avoir un ensemble d'entraînement D_n qui *grandit linéairement* (contre exponentiellement dans le cas de la tomographie) avec le nombre de qubits[17] sur lesquels $\rho_?$ est défini afin de produire avec une bonne probabilité une hypothèse qui pourra prédire avec une bonne précision le comportement de $\rho_?$ sur des POVMs non observés (ce qui est une forme de généralisation). La preuve du théorème de Aaronson se base sur une succession de résultats provenant de la théorie calculatoire de l'apprentissage, dont la dimension de Vapnik-Chernovenkis (VC) [37] qui est un concept central de la théorie de l'apprentissage.

[17]Le théorème exact de Aaronson inclut une borne sur la taille de l'échantillon qui dépend aussi de paramètres comme la précision de l'hypothèse générée ou la probabilité de succès de l'apprentissage.

CHAPITRE 5

ALGORITHMES QUANTIQUES D'APPRENTISSAGE NON-SUPERVISÉ

5.1 Introduction

Considérons le scénario suivant d'une tâche de catégorisation particulièrement ambitieuse.

Scénario 5.1 (Catégorisation de tous les habitants de la Terre). *Imaginez que vous êtes un employé du département de statistiques des Nations Unies. Votre patron vient vous voir avec les données démographiques de tous les habitants de la Terre et vous demande d'analyser ces données en utilisant un algorithme de catégorisation avec l'espoir que cette analyse pourra révéler des groupes intéressants parmi la population. En voyant que vous semblez un peu perdu devant l'immensité du travail à accomplir, votre patron vous dit de ne pas vous inquiéter car pour vous aider à réaliser cette tâche, il a réussi à "emprunter" le prototype d'un ordinateur quantique complètement opérationnel au centre de la sécurité des télécommunications.*

Interrogation : est-il possible d'utiliser cet ordinateur quantique pour résoudre la tâche de catégorisation plus rapidement ?

Les algorithmes d'apprentissage non-supervisé sont fréquemment utilisés en *forage de données* [188] (ou *data mining* en anglais) pour des applications où la taille des données à traiter est gigantesque tel que l'astronomie, la bio-informatique ou encore le traitement de données issues de réseaux de grande taille comme Internet. Disposer d'algorithmes rapides et performants est vital pour ces applications, et dans ce contexte il ne suffit plus que l'algorithme fonctionne en temps polynomial pour qu'il soit considéré efficace. Ainsi, même un algorithme fonctionnant en temps quadratique pourrait être totalement inutilisable en pratique s'il demandait plusieurs mois avant de se terminer, par rapport à un algorithme linéaire qui aurait un temps d'exécution plus raisonnable. Cet exemple souligne l'importance de développer les algorithmes les plus efficaces possibles. En particulier, comme nous allons le voir dans ce chapitre, le paradigme offert par l'informatique quantique permet d'accélérer certains algorithmes classiques d'apprentissage non-supervisé.

Ainsi que nous l'avons vu dans la section 4.3, des algorithmes d'apprentissage se basant sur des primitives quantiques ont déjà été développés en apprentissage supervisé et apprentissage par renforcement. Cependant, au niveau de l'apprentissage non-supervisé, le terrain était resté quasiment vierge, à l'exception d'un algorithme quantique pour l'arbre couvrant minimal dont l'usage peut être détourné pour faire de la catégorisation (voir section 5.4 pour plus de détails). La plupart des algorithmes quantiques d'apprentissage non-supervisé développés dans ce chapitre se basent sur des variantes de l'algorithme de Grover afin d'accélérer le temps d'exécution de l'algorithme comparativement à sa variante classique. Les versions quantisées des algorithmes de catégorisation offrent un temps d'exécution plus rapide que leur contreparties classiques, mais en revanche elles n'améliorent pas la qualité de la catégorisation générée. Ainsi, si un problème de catégorisation particulier est NP-ardu [96] pour la recherche de la solution optimale à sa fonction de coût, les algorithmes quantiques décrits ici, bien que plus rapides que leurs homologues classiques, n'améliorent pas la qualité de la solution retournée[1].

Le plan de ce chapitre est le suivant. La section 5.2 rappelle le principe de l'algorithme de Grover et présente brièvement quelques-unes de ses variantes. Certaines d'entre elles sont détaillées plus tard dans la section 5.3.2 car elles seront utilisées comme sous-routines d'algorithmes d'apprentissage. La section 5.3 décrit le modèle de boîte noire[2] adapté pour la tâche de catégorisation. Une recette explicite est aussi donnée pour construire, à partir de la description classique de l'ensemble de données, un circuit pouvant jouer le rôle de l'oracle. Les versions quantisées des algorithmes de catégorisation basée sur *l'arbre couvrant minimal*, de *catégorisation divisive*, et des *k-médianes* (dans sa version standard et distribuée) sont respectivement décrites dans les sections 5.4, 5.5 et 5.6. Ensuite, la section 5.7 décrit une série d'algorithmes pouvant être utilisés comme outils durant l'étape de prétraitement d'autres algorithmes d'apprentissage non-supervisé : *construction d'un graphe de voisinage*, *détection d'anomalies* et *initialisation des centres de catégories*. Finalement, la section 5.8 conclut ce chapitre par une discussion et quelques perspectives

[1] Dans le modèle *boîte noire* (voir section 5.3.1), il a été prouvé par Bennett, Bernstein, Brassard et Vazirani [22] que si la seule connaissance qu'on peut avoir d'un problème NP-complet est accessible par un oracle alors le quantique offre au mieux un avantage quadratique sur le classique. Autrement dit, si la recherche de la solution demandait un temps $O(2^n)$ classiquement, le mieux que l'on pourrait espérer faire quantiquement dans ce modèle est de l'ordre de $O(2^{\frac{n}{2}})$.

[2] Les expressions "boîte noire" et "oracle" sont utilisées de manière interchangeable dans ce chapitre, bien qu'ils puissent avoir des sens différents dans la littérature.

futures, dont la proposition d'une version quantique d'Isomap, un algorithme de réduction de dimensionnalité.

5.2 L'algorithme de Grover et ses variantes

Dans la version originale de l'algorithme de Grover [103], une fonction booléenne f est donnée sous la forme d'une boîte noire avec la promesse additionnelle qu'il existe un unique x_0 tel que $f(x_0) = 1$. Classiquement, trouver ce x_0, quand la fonction f n'offre aucune structure particulière, demande de faire en moyenne $n/2$ requêtes à la boîte noire, où n est le nombre de points du domaine de f. L'algorithme de Grover résout le même problème en faisant approximativement seulement $\sqrt{n}$ accès à la boîte noire. Contrairement à la version classique, ces accès sont faits sous forme de superposition quantique.

L'algorithme de Grover commence par appliquer une tour de portes de Walsh–Hadamard à l'état initial composé uniquement d'une suite de zéros afin de créer une superposition égale de toutes les entrées possibles. L'algorithme procède ensuite en répétant un nombre adéquat de fois *l'itération de Grover* qui se compose de deux étapes : un appel au circuit quantique décrit dans la figure 5.1, qui inverse la phase d'un état inconnu x tel que $f(x) = 1$ (c'est à dire l'état "cible") par retour de phase, et *une inversion par rapport à la moyenne*, qui est une opération unitaire définie indépendamment de la fonction f considérée. Cette itération doit être répétée approximativement $\frac{\pi}{4}\sqrt{n}$ fois, d'où le temps d'exécution de l'algorithme de l'ordre de $O(\sqrt{n})$. Une application de l'itération de Grover a pour effet d'accroître légèrement l'amplitude de l'état recherché, tout en faisant diminuer les amplitudes des autres états. Après le nombre adéquat d'itérations de Grover, l'amplitude de l'état recherché sera très proche de 1, ainsi si on mesure le registre quantique à ce moment-là, on va observer l'état recherché avec quasi-certitude.

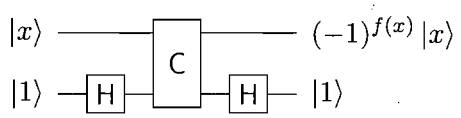

FIG. 5.1 – Calculer une fonction par retour de phase.

À partir de l'idée originale de Grover, plusieurs généralisations de son algorithme ont

été développées pour les cas où il y aurait plus d'un x tel que $f(x) = 1$. Ainsi, pour tout $t > 0$, où t est le nombre de solutions, un nombre d'applications de l'itération de Grover de l'ordre de $\frac{\pi}{4}\sqrt{n/t}$ sera suffisant pour trouver une de ces solutions [41]. Si le nombre de solutions t est inconnu, le même article montre qu'il reste néanmoins possible de trouver une solution parmi les t possibles en un temps proportionnel à $\sqrt{n/t}$. Il existe aussi des extensions à l'algorithme de Grover qui permettent de *compter* (exactement ou approximativement) le nombre de solutions [46,47].

D'autres applications de l'algorithme de Grover permettent de trouver le minimum (ou le maximum) d'une fonction (ainsi que sa position) [78] ou les c plus petites (ou plus grandes) valeurs dans son image [77] après $\Theta(\sqrt{n}\,)$ et $\Theta(\sqrt{cn}\,)$ appels à la fonction, respectivement. D'autres variantes permettent d'approximer la médiane ou des statistiques qui lui sont reliées [149] avec un gain quadratique comparativement au meilleur algorithme classique possible. Enfin, la généralisation de l'algorithme de Grover connue sous le nom *d'amplification d'amplitude* [46] permet d'accélérer n'importe quel algorithme classique probabiliste par un facteur quadratique en terme du nombre de requêtes effectuées à la boîte noire[3].

5.3 Quantisation d'algorithmes de catégorisation

Pour rappel (définition 4.8), la *quantisation* consiste à convertir partiellement ou totalement un algorithme classique en algorithme quantique afin d'améliorer ses performances. Dans cette section, nous allons développer un modèle et des outils permettant de quantiser des algorithmes d'apprentissage non-supervisé, dont principalement des algorithmes de catégorisation. Bien que reliée, la tâche de quantisation d'algorithme de catégorisation ne devrait pas être confondue avec la mise au point d'algorithmes de catégorisation classiques inspirés de la mécanique quantique (cf. section 4.4) ou la tâche consistant à faire de la catégorisation directement sur de l'information quantique (cf. section 6.3.1).

[3]Pour être précis, l'amplification d'amplitude peut même être utilisée pour accélérer avec un gain quadratique tout algorithme quantique qui n'effectuerait pas de mesure en cours de route.

5.3.1 Modèle de boîte noire

Traditionnellement, en catégorisation on considère un ensemble d'entraînement composé de n points de données dénoté par $D_n = \{x_1, \ldots, x_n\}$, où chaque point de données x correspond à un vecteur de d attributs. Le but d'un algorithme de catégorisation est de partitionner l'ensemble D_n en sous-ensembles de points appelés *catégories* (*clusters* en anglais), de manière à ce que les objets similaires soient regroupés dans la même catégorie (*intra-similarité*) alors que les objets dissemblables soient placés dans des catégories différentes (*inter-dissimilarité*). Une des hypothèses est qu'il existe une notion de *distance* (ou une *mesure de similarité/dissimilarité*) pouvant être évaluée afin de comparer chaque paire de points. Cette métrique sera utilisée par l'algorithme pour former les catégories.

Le modèle considéré dans ce chapitre diffère de ce cadre traditionnel et correspond plutôt au *modèle de boîte noire*. Dans ce modèle, notre connaissance de l'ensemble de données provient uniquement d'une boîte noire (aussi appelée "oracle"), qu'il est possible d'interroger pour apprendre la distance entre deux points. Aucune hypothèse n'est faite *a priori* sur les propriétés de cette distance, si ce n'est qu'elle est symétrique et non-négative[4]. (En particulier, l'inégalité du triangle n'a pas besoin d'être respectée[5]). Ce modèle est généralement utilisé afin de dériver des bornes inférieures pour des problèmes pour lesquels il est difficile de prouver de telles bornes dans un contexte plus général. À noter que dans ce modèle, la complexité d'un algorithme en terme du nombre de requêtes/appels à l'oracle constitue une borne inférieure de sa complexité en terme de temps de calcul.

Ce modèle de boîte noire est comparable à celui imaginé par Angluin [13], qui est utilisé en théorie calculatoire de l'apprentissage pour étudier la complexité en terme de requêtes pour apprendre *exactement* une fonction donnée sous la forme d'une boîte noire. Un analogue quantique du modèle de Angluin a été défini par Servedio [168] (voir la section 4.2.2 pour plus de détails). Dans ce chapitre, notre but est de faire de la catégorisation, et non pas d'apprendre la fonction calculée par l'oracle. Au meilleur

[4]Si la distance n'est pas symétrique, les algorithmes présentés dans ce chapitre peuvent être facilement modifiés pour prendre cela en compte sans modifier le temps de calcul de façon conséquente.

[5]Dans le cas où la propriété de symétrie ou l'inégalité du triangle ne sont pas respectées, le terme *mesure de dissimilarité* serait plus approprié que le terme distance si on veut être rigoureux mathématiquement.

de ma connaissance, il n'existe pas de travaux antérieurs étudiant la complexité de la catégorisation dans le modèle d'Angluin, que ce soit dans sa variante classique ou quantique. Cependant, un problème similaire a été considéré [144] dans la version classique du modèle PAC (*Probably Approximately Correct*). Dans ce travail, le but était de caractériser le nombre de requêtes qu'il est nécessaire de demander à l'oracle pour apprendre (dans le sens PAC du terme) une catégorisation spécifique parmi une classe de catégorisations possibles.

Dans la version classique du modèle boîte noire, une requête à l'oracle correspond à demander la distance entre deux points x_i et x_j en lui donnant les index i et j de ces points. La boîte noire quantique correspondante est illustrée dans la figure 5.2, où O signifie "oracle". Afin d'obéir aux principes de la mécanique quantique, O doit être unitaire (et donc réversible). En pratique, cela n'est pas vraiment une restriction car il est possible de transformer n'importe quel circuit classique irréversible en un circuit réversible pour un coût "raisonnable" [20]. Il suffit donc de donner la description d'un circuit classique irréversible réalisant l'oracle, pour pouvoir le convertir en circuit réversible et ainsi pouvoir potentiellement l'implémenter quantiquement.

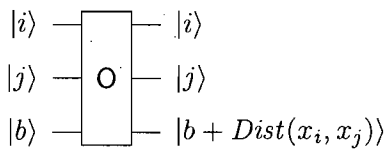

FIG. 5.2 – Illustration de l'oracle de distance : i et j sont les index de deux points de D_n et $Dist(x_i, x_j)$ représente la distance entre ces deux points. L'addition $b + Dist(x_i, x_j)$ est calculée dans un groupe fini de taille appropriée entre le registre ancillaire b et la distance $Dist(x_i, x_j)$.

Si la boîte noire est quantique, elle peut être interrogée en superposition d'entrées. Ainsi, si tous les qubits d'entrée sont initialement dans l'état $|0\rangle$ et qu'on applique la porte de Walsh–Hadamard sur chacun d'entre eux (à l'exception du registre ancillaire contenant $|b\rangle$ qu'on laisse inchangé à $|0\rangle$), l'entrée aura été transformée en une superposition égale de toutes les paires d'index des points de données[6]. Dans ce cas précis, la sortie résultante est une superposition possible de tous les triplets $|i, j, Dist(x_i, x_j)\rangle$.[7]

[6] Pour simplifier l'analyse et sans perte de généralité, nous supposerons que le nombre d'index possibles est une puissance de deux.

[7] À ne pas confondre avec simplement un superposition des toutes les distances entre les paires de points, ce qui n'aurait pas de sens quantiquement d'une manière générale.

L'hypothèse que l'occurrence particulière à catégoriser est donnée seulement sous la forme d'une boîte noire n'est pas vraiment réaliste en pratique, même s'il s'agit du paradigme usuellement considéré en informatique quantique ou dans le modèle d'Angluin. Nous verrons dans la section 5.3.3 comment abandonner cette hypothèse en donnant une recette explicite permettant de construire l'oracle O à partir de la description classique de l'ensemble de données D_n.

5.3.2 Sous-routines quantiques

Cette section décrit trois sous-routines quantiques qui seront utilisées pour accélérer des algorithmes classiques de catégorisation. Toutes ces sous-routines sont basées sur des variations de l'algorithme de Grover. En particulier, les deux premières sont des applications directes des travaux antérieurs de respectivement, Dürr et Høyer [78] et Dürr, Heiligman, Høyer et Mhalla [77]. La troisième sous-routine est une nouvelle, bien que simple, application de l'algorithme de Grover.

L'algorithme quant_trouver_max décrit ci-dessous (algorithme 3) s'inspire directement de l'algorithme de Dürr et Høyer pour rechercher le minimum[8] d'une fonction [78]. Il permet de trouver la paire de points les plus éloignés de l'ensemble de données (la distance entre ces deux points est appelé *diamètre* de l'ensemble de points). Un algorithme similaire peut être utilisé pour trouver le point de données qui est le plus éloigné d'un point spécifique.

Algorithme 3 quant_trouver_max(D_n)

Choisir au hasard deux index initiaux i et j
Initialiser $d_{max} = Dist(x_i, x_j)$
Répéter
  En utilisant l'algorithme de Grover, trouver deux index i et j tels que
  $Dist(x_i, x_j) > d_{max}$ si jamais ils existent
  Mettre à jour $d_{max} = Dist(x_i, x_j)$
Jusqu'à ce qu'aucun nouveau i, j soient trouvés
Retourner i, j

L'algorithme commence par choisir uniformément au hasard deux index i et j.

[8] L'algorithme de Dürr et Høyer dans sa version originale a été formulé pour rechercher le minimum, mais il peut être facilement adapté pour rechercher le maximum à la place tout en gardant la même complexité de calcul.

Une première estimation grossière pour le diamètre est obtenue en prenant simplement $d_{max} = Dist(x_i, x_j)$. En se basant sur le circuit effectuant le changement de phase décrit dans les figures 5.3 et 5.4, l'algorithme de Grover permet de trouver une nouvelle paire de points (i, j), si elle existe, telle que $Dist(x_i, x_j) > d_{max}$. Si un telle paire n'existe pas, on a alors calculé le diamètre et l'algorithme se termine. Sinon, la valeur de la distance d_{max} est mise à jour à $Dist(x_i, x_j)$ et la procédure est répétée jusqu'à ce que l'algorithme converge à la distance maximale.

Théorème 5.1 (Convergence quant_trouver_max). *Avec probabilité élevée, l'algorithme* quant_trouver_max *retourne les index i et j de la paire de points les plus éloignés après un nombre espéré de requêtes de l'ordre de $\sqrt{p}$, où $p = n^2$ est le nombre de paires de points de données, d'où un nombre total de requêtes qui est de $O(n)$. Pour le cas le plus simple où on souhaiterait trouver le point le plus éloigné d'un point spécifique, l'algorithme correspondant prend un temps de $O(\sqrt{n})$.*

Démonstration. La preuve de convergence de quant_trouver_max est similaire à l'analyse de Dürr et Høyer [78] pour la version permettant de trouver le minimum d'une fonction. □

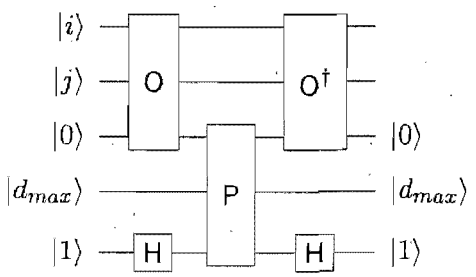

FIG. 5.3 – Circuit réalisant le changement de phase dans la version de l'algorithme de Grover qui permet de trouver la paire de points à distance maximum (algorithme 3). La transformation unitaire $O^\dagger$ est la transposée conjuguée de O. La sortie est identique à l'entrée, à l'exception de la phase globale de $|i\rangle\,|j\rangle$ qui sera inversée si et seulement si $Dist(x_i, x_j) > d_{max}$. (Voir la figure 5.4 pour la définition de P.)

Dürr, Heiligman, Høyer et Mhalla ont développé un algorithme qui permet de chercher les c valeurs minimales d'une fonction [77] en un temps espéré de $O(\sqrt{cn})$, pour n le nombre de points dans l'image de la fonction[9]. Si on choisit cette fonction comme étant

[9]Rechercher le minimum d'une fonction peut être vu comme un cas particulier de cet algorithme pour $c = 1$.

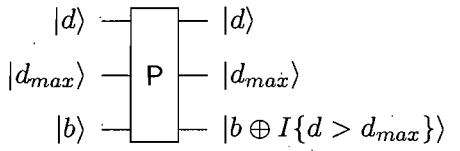

FIG. 5.4 – Sous-circuit P utilisé dans la figure 5.3 pour réaliser le retour de phase, où $I\{\cdot\}$ est la *fonction indicatrice* qui vaut 1 si son argument est vrai et 0 sinon, et "$\oplus$" représente le ou-exclusif.

la distance entre un point fixé et tous les autres points, on obtient la seconde sous-routine quant_trouver_c_plus_proches_voisins qui est l'application directe de l'algorithme de Dürr, Heiligman, Høyer et Mhalla [77] adapté pour trouver les c plus proches voisins d'un point.

Théorème 5.2 (Convergence quant_trouver_c_plus_proches_voisins [77]). *Avec probabilité élevée, l'algorithme* quant_trouver_c_plus_proches_voisins *permet de trouver les c plus proches voisins d'un point (au lieu des c valeurs minimales d'une fonction) en un temps de* $O(\sqrt{cn})$.

À noter que l'algorithme pour trouver les c valeurs minimales [77] est plus complexe que simplement appliquer c fois l'algorithme pour trouver le minimum, qui donnerait un temps d'exécution de l'ordre de $O(c\sqrt{n})$ et non pas $O(\sqrt{cn})$.

La troisième et dernière sous-routine est un nouvel algorithme, baptisé quant_trouver_mediane, qui permet de calculer la médiane parmi un ensemble de m points $Q_m = \{z_1, \ldots, z_m\}$.

Définition 5.1 (Médiane). *La* médiane *est le point parmi un ensemble dont la somme des distances à tous les autres points (ou la distance moyenne) est minimale.*

La notion de médiane est particulièrement intuitive dans le sens de la norme L_1 mais peut être généralisée à d'autres situations[10] (voir par exemple le survol [176]). Dans notre cas, on peut définir formellement la médiane d'un ensemble de points Q_m comme étant le point z_i tel que :

$$\text{mediane}(Q_m) = \arg\min_{z_i \in Q_m} \sum_{j=1}^{m} Dist(z_i, z_j) \tag{5.1}$$

Trouver la médiane peut être réalisé classiquement en calculant pour chaque point dans l'ensemble, la somme de ses distances (ou la distance moyenne) avec tous les autres

[10] Dans le cas de points définis sur plusieurs dimensions, le terme *médoïde* est parfois utilisé à la place du terme médiane.

points et en prenant le minimum. Ce processus requiert un temps de l'ordre de $O(m^2)$, où m est le nombre de points considérés. Dans le cas général où il n'y a aucune restriction sur la distance utilisée ou aucune structure parmi l'ensemble de points pouvant être exploitée, il n'existe pas d'approche plus efficace que l'algorithme naïf présenté ci-dessus[11].

Quand les z_i correspondent simplement à des nombres, ou plus généralement, quand tous les points sont colinéaires, un algorithme quantique dû à Nayak et Wu [148,149] peut être utilisé pour *approximer la médiane* en un temps de $\Theta(\sqrt{m})$. Cependant dans le cas plus général considéré dans ce chapitre, le but est de *calculer exactement la médiane* en ayant seulement comme information la distance entre chaque paire de points (de plus, l'inégalité du triangle peut ne pas être vérifiée), ce qui correspond à une situation où l'algorithme de Nayak et Wu ne s'applique pas.

Pour résoudre ce problème de manière quantique, il suffit de construire le circuit S illustré par la figure 5.5, qui prend $|i\rangle$ comme entrée, avec $1 \leq i \leq m$, et calcule la somme des distances entre z_i et tous les autres points dans Q_m. Pour cela, il suffit d'appliquer la boîte noire décrite dans la figure 5.2 successivement pour chaque valeur de j, $1 \leq j \leq m$. (On suppose que $Dist(z_i, z_i) = 0$.) Ceci prend un temps de $\Theta(m)$, mais pourrait être possiblement amélioré (voir la question ouverte 5.1).

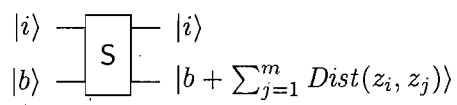

FIG. 5.5 – Calcul de la somme des distances entre z_i et tous les autres points de l'ensemble $Q_m = \{z_1, \ldots, z_m\}$. L'oracle S peut être obtenu en répétant m fois l'oracle O décrit dans la figure 5.2 pour j allant de 1 jusqu'à m.

L'algorithme de Dürr et Høyer pour trouver le minimum [78] peut ensuite être utilisé pour trouver la somme minimale (ce qui est équivalent à trouver la distance minimale moyenne) parmi tous les z_i possibles en utilisant $\Theta(\sqrt{m})$ applications du circuit de la figure 5.5.

Lemme 5.1 (Convergence de quant_trouver_mediane). *Avec probabilité élevée, l'algorithme* quant_trouver_mediane *trouve la médiane parmi un ensemble de m points en temps*

[11] Une des façons de s'en convaincre est de construire un scénario où tous les points sont à la même distance les uns des autres, à l'exception de deux points qui sont plus proches l'un de l'autre. Ces deux points sont les médianes de cet ensemble de points. Dans ce cas-là, classiquement, il faudra demander quasiment toutes les distances entre paires de points à la boîte noire avant d'identifier l'une des deux médianes. Ceci résulte en une borne inférieure de $\Omega(m^2)$ appels à l'oracle.

de $O(m^{3/2})$.

Démonstration. Chaque application du circuit prend un temps de $O(m)$ et de plus, trouver la somme minimale parmi m possibles requiert $O(\sqrt{m})$ applications du circuit de la figure 5.5 en utilisant l'algorithme pour trouver le minimum [78]. Le temps total pour calculer la médiane sera donc de $O(m\sqrt{m}) = O(m^{3/2})$. □

L'algorithme quant_trouver_mediane qui permet de calculer la médiane de m points en un temps $O(m^{3/2})$ pourrait ne pas être optimal, ce qui conduit à la formulation de la question ouverte suivante.

Question ouverte 5.1 (Borne inférieure du calcul de la médiane parmi un ensemble de points). *Est-ce que calculer* exactement *la médiane parmi un ensemble de* m *points nécessite* $\Omega(m^{3/2})$ *appels à l'oracle (comme le fait l'algorithme* quant_trouver_mediane*) ou est-ce que* $O(m)$ *appels à l'oracle sont suffisants ?*

Répondre à cette question déterminerait si le calcul de la médiane d'un ensemble de points est équivalent ou plus complexe en terme du nombre d'appels à l'oracle que de simplement trouver la paire de points qui sont à distance minimale ou maximale. Cette réponse aurait aussi une influence directe sur la version quantisée de l'algorithme de k-médianes qui sera présentée dans la section 5.6.

5.3.3 Construire explicitement l'oracle

Les sous-routines décrites dans la section précédente présupposent que l'oracle est fourni directement comme ressource à l'exécution de l'algorithme et que la connaissance que nous pouvons obtenir sur l'ensemble de données provient uniquement de requêtes effectuées à cet oracle. Ce modèle, dit de boîte noire, est couramment utilisé en informatique quantique[12], car il permet de dériver des bornes inférieures ou de décrire de façon relativement abstraite des algorithmes sans avoir besoin de préciser les détails d'implémentation. En pratique[13] cependant, cette approche n'est pas vraiment réaliste. En effet dans la vie de tous les jours d'un spécialiste de l'apprentissage quantique, il semble improbable que

[12]Pensons par exemple, à la formulation originale de Grover en terme de recherche d'un aiguille dans une botte de foins.

[13]L'expression "en pratique" s'applique à un monde où l'ordinateur quantique de taille raisonnable est disponible :-).

quelqu'un nous fournisse directement une boîte noire quantique[14] pouvant être utilisée pour interroger la distance entre des paires de points en superposition. À la place, il est plus plausible que nous recevions un ensemble de données D_n qui contient la description (classique) des n points de données.

À partir de cet ensemble de données, il est important de connaître une recette explicite permettant de construire un circuit quantique efficace réalisant la même fonctionnalité que la boîte noire. Il existe une borne inférieure triviale de $\Omega(nd)$ sur tout algorithme qui prendrait D_n comme entrée classique et qui produirait un circuit quantique jouant le rôle de l'oracle, car cet algorithme devrait parcourir au moins une fois chaque point de données de l'ensemble, et observer pour chaque point les valeurs de tous ses attributs. L'approche constructive décrite ci-après permet de réaliser un tel circuit en $O(nd+n\log n)$ opérations (la taille du circuit obtenu est proportionnelle au temps de calcul nécessaire pour construire le circuit), où n est le nombre de points de l'ensemble de données et d est la dimension de l'espace dans lequel vivent ces points (c'est-à-dire le nombre d'attributs les décrivant). Si nous nous restreignons sans perte de généralité, à un circuit logique composé uniquement de portes d'arité ou de sortance qui est au maximum c, pour c une petite constante (par exemple $c = 2$), on peut définir sa *taille* et sa *profondeur* de la manière suivante[15].

Définition 5.2 (Taille d'un circuit). *La* taille d'un circuit *est le nombre de portes logiques qui composent ce circuit.*

Définition 5.3 (Profondeur d'un circuit). *La* profondeur d'un circuit *est la distance maximum (en terme de portes) depuis une entrée du circuit jusqu'à une de ses sorties.*

Grâce aux travaux de Bennett sur la réversibilité du calcul classique [20], il suffit de décrire un circuit classique (non nécessairement réversible) permettant de réaliser une tâche pour ensuite le transformer en circuit réversible sans augmenter de façon significative sa profondeur et sa taille, avant de pouvoir ensuite l'implémenter quantiquement.

[14] On pourrait cependant imaginer un scénario où la boîte noire quantique nous serait donnée par une entité (comme un fournisseur de données par exemple). Pour des raisons de confidentialité, cette entité ne souhaiterait pas divulguer la description complète de l'ensemble de données mais serait prêt à fournir une implémentation quantique permettant d'apprendre de l'information sur des relations entre les points (comme leurs distances). Ce type de scénario est particulièrement vraisemblable dans le domaine du *forage de données préservant la confidentialité* [11] (appelé *privacy-preserving data mining* en anglais).

[15] Les portes FANOUT qui produisent deux copies identiques du bit d'entrée doivent elles-aussi être comptabilisées parmi les portes logiques composant le circuit.

La première étape consiste à décomposer l'oracle O décrit précédemment dans la figure 5.2 en deux sous-circuits ; un sous-circuit E, qui à partir de l'index i d'un point de l'ensemble de données, produit sa description x_i, et un sous-circuit D qui prend la description de deux points en entrée et calcule la distance entre ces deux points.

Lemme 5.2 (Encodage d'un ensemble de données sous forme de circuit). *Le sous-circuit* E, *qui produit la description d'un point à partir de son index, peut être réalisé avec un circuit quantique de profondeur* $O(\log(nd))$ *et de taille* $O(nd + n\log n)$. *De plus, l'algorithme qui construit explicitement ce circuit à partir de la description classique de* D_n, *l'ensemble de données, requiert un temps lui aussi de* $O(nd + n\log n)$.

Démonstration. Considérons une implémentation naïve du circuit E qui encode directement l'ensemble de données sous forme de circuit. Pour chaque bit individuel $i_1, \ldots, i_{\log n}$ du registre servant à décrire l'index i d'un point de données (qui est de taille logarithmique en n), on peut en produire n copies en utilisant un arbre binaire composé de FANOUT. Chaque arbre de FANOUT est de profondeur $O(\log n)$ et de taille $O(n)$. La forêt d'arbres binaires de FANOUT (figure 5.6) qui copie tous les bits du registre d'index est donc de profondeur $O(\log n)$ et de taille $O(n\log n)$.

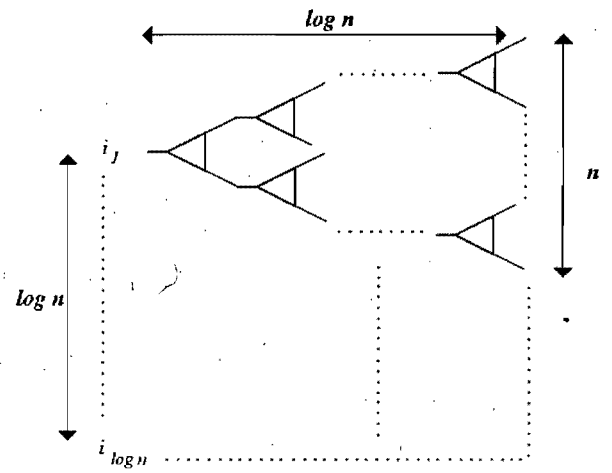

FIG. 5.6 – Forêt d'arbres binaires de FANOUT réalisant la copie des bits d'entrées. L'action de la porte FANOUT est simplement de produire deux copies identiques du bit d'entrée.

Pour chacun des n index possibles des points, on détermine ensuite si l'index i passé en entrée au circuit est égal à cet index ou non. De façon équivalente, on veut calculer

pour tous les index possibles $j \in \{1, \ldots, n\}$, la fonction indicatrice $I\{j = i\}$ dont la valeur booléenne est 1 uniquement si l'index j examiné actuellement correspond à l'index i passé en entrée au circuit, et 0 pour toutes les autres valeurs possibles d'index. Cette fonction indicatrice peut être implémentée par un arbre de AND de profondeur $O(\log \log n)$ et de taille $O(\log n)$ (figure 5.7). Au premier niveau, avant d'utiliser l'arbre de AND, on applique la porte NOT sur chacun des bits de i pour lesquels le bit de j correspondant vaut 1. Si le bit correspondant de j vaut 0 alors on ne fait rien et on laisse le bit intact. Comme on calcule ensuite le AND de tous les $\log n$ bits résultants du premier niveau, la valeur booléenne finale est de 1 uniquement si $i = j$, c'est-à-dire si tous les bits de l'index j courant sont égaux à ceux de l'index i passé en entrée. Comme il y a n fonctions indicatrices au total, cette partie du circuit est de taille $O(n \log n)$ pour une profondeur $O(\log \log n)$.

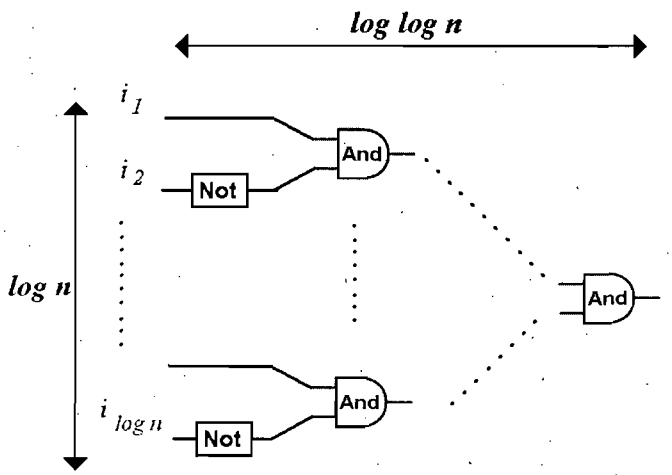

FIG. 5.7 – Circuit réalisant la fonction indicatrice $I\{i = j\}$ pour $i, j \in \{1, \ldots, n\}$, où i est l'index passé en entrée au circuit et j l'index d'un point spécifique. Ce circuit est répété n fois en parallèle pour réaliser les fonctions indicatrices des n index possibles. Dans l'exemple courant de la figure, il s'agit de déterminer la valeur de la fonction indicatrice $I\{i = 01^{\log \frac{n}{2}}\}$.

Ensuite, supposons sans perte de généralité que chaque attribut est codé sur un nombre fini et constant de bits. Comme chaque point est décrit par d attributs, on fait $O(d)$ copies de chacun des n bits obtenus à la sortie des fonctions indicatrices lors de l'étape précédente, c'est-à-dire un nombre constant de copies pour chacun des d attributs. Pour cela, on utilise une forêt d'arbres binaires de FANOUT (figure 5.8) comme durant

la première étape. Cette forêt est de profondeur $O(\log d)$ et de taille $O(nd)$.

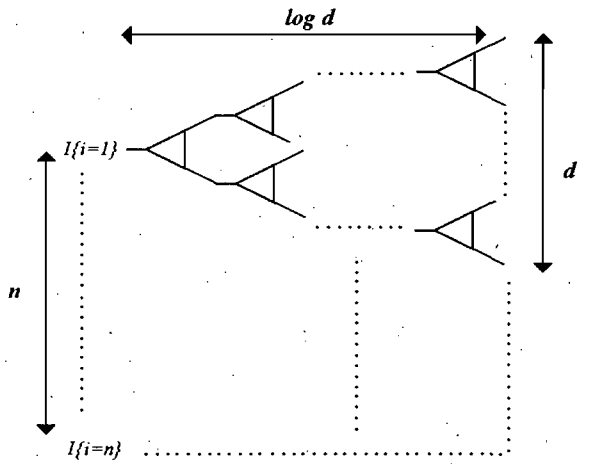

FIG. 5.8 – Forêt d'arbres binaires de FANOUT réalisant la copie des n fonctions indicatrices. Chaque fonction indicatrice sera copiée $O(d)$ fois, pour d le nombre d'attributs.

Enfin pour chacune des n sorties des fonctions indicatrices et chacun des bits de chaque attribut d, on réalise la construction suivante (figure 5.9). Tout d'abord, cette construction effectue un AND entre le résultat de la fonction indicatrice pour l'index j courant et la valeur du bit dans la description de x_j pour l'attribut actuel. En faisant ensuite un OR sur tous les sorties des AND du premier niveau, en utilisant là encore un arbre binaire mais cette fois composé de OR, on obtient en sortie la valeur du bit de l'attribut courant de x_i. Cette forêt d'arbres est de taille $O(nd)$ et de profondeur $O(\log n)$.[16]

Le tableau 5.1 récapitule la taille et la profondeur des différentes parties du circuit. Pour résumer, le circuit final classique irréversible obtenu est de taille $O(nd + n \log n)$, de profondeur $O(\log(nd))$ et utilise uniquement des portes d'arité ou de sortance 2 tel que FANOUT, AND et OR, ainsi que des portes NOT. Ce circuit peut être transformé en une version réversible en remplaçant chacune de ces portes[17] par une ou deux applications d'une porte universelle réversible, telle que la porte de Toffoli [180] ou la porte de

[16] Toutes les fois où la valeur du bit de l'attribut pour un point d'index spécifique vaut 0, on pourrait remplacer le AND du premier niveau par le bit constant 0 sans affecter le résultat final. Bien que ne changeant pas la complexité finale du circuit, cette technique permet d'éviter de rajouter des portes inutilement.

[17] Sauf la porte unaire NOT qui est déjà intrinsèquement réversible.

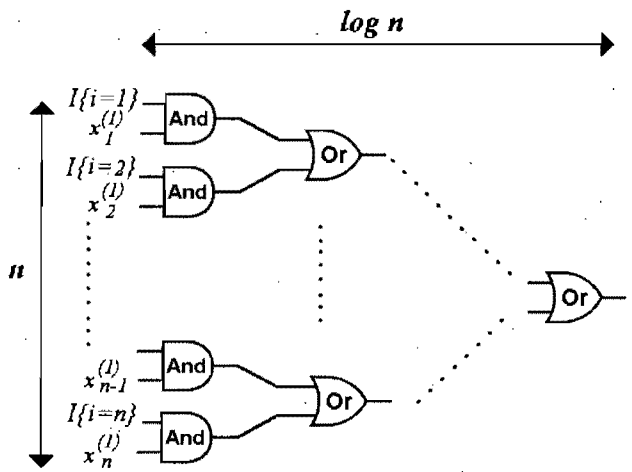

FIG. 5.9 – Circuit encodant la valeur d'un bit d'un attribut pour chacun des n index possibles des points de données. Au premier niveau, ce circuit réalise un AND entre chacune des fonctions indicatrices et un bit de l'attribut pour le point d'index correspondant (dans la figure le premier bit du premier attribut est utilisé comme exemple). La valeur qui se trouve en sortie de l'arbre binaire de OR est la valeur d'un bit d'un attribut du point x_i, où i est l'index passé en entrée au circuit. Ce circuit sera répété $O(d)$ fois en parallèle (soit un nombre constant par attribut) afin de générer la description complète des d attributs composant x_i.

Fredkin [88] (voir chapitre 4 de [42]). Ainsi au final, le circuit réversible classique pour E obtenu est d'une taille qui est du même ordre de grandeur que sa version irréversible. Comme n'importe laquelle de ces deux portes peut être implémentée quantiquement par un nombre constant de portes quantiques unaires et binaires (voir [152], page 182), cela implique qu'il existe un circuit quantique implémentant E qui est de taille $O(nd+n\log n)$ et de profondeur $O(\log(nd))$. De plus, le temps de construction de ce circuit est directement proportionnel à sa taille. □

Une fois qu'on a construit le sous-circuit E, il reste encore à construire le sous-circuit D qui prend en entrée la description de deux points et calcule leur distance de façon réversible.

Remarque 5.1 (Profondeur d'un circuit classique implémentant une distance). *Pour la plupart des métriques communément utilisées en apprentissage machine, comme la distance euclidienne, la distance de Manhattan ou toute variante de la distance de Minkowski, le circuit classique irréversible calculant cette distance aura une profondeur de*

Étape	Taille	Profondeur
Copies des bits d'entrée	$O(n \log n)$	$O(\log n)$
Implémentation des fonctions indicatrices	$O(n \log n)$	$O(\log \log n)$
Copies des bits des fonctions indicatrices	$O(nd)$	$O(\log d)$
Encodage des attributs	$O(nd)$	$O(\log n)$
Total pour le circuit E	$O(nd + n \log n)$	$O(\log(nd))$

TAB. 5.1 – Tableau résumant la taille et profondeur des différentes parties formant le circuit E qui fait l'encodage de l'ensemble de données.

$O(\log d)$ *et une taille de l'ordre de* $O(d)$.

Exemple 5.1 (Circuit classique implémentant la distance euclidienne). *La distance euclidienne entre deux points* x_a *et* x_b *décrits par* d *attributs est égale à*

$$Dist_{L_2}(x_a, x_b) = \sqrt{\sum_{i=1}^{d} \left(x_a^{(i)} - x_b^{(i)} \right)^2} \tag{5.2}$$

où $x_a^{(i)}$ *et* $x_b^{(i)}$ *sont les valeurs du* $i^{ème}$ *attribut, pour respectivement les points* x_a *et* x_b. *Pour calculer cette distance, il suffit pour chaque attribut, de soustraire la valeur pour cet attribut du premier point et du second point et de mettre le résultat obtenu au carré. Ensuite, on additionne les* d *résultats obtenus (un par attribut) avant de calculer la racine carrée de la somme. La soustraction, la multiplication, l'addition ainsi que le calcul de la racine carrée d'un nombre sont toutes des opérations arithmétiques pouvant être réalisées par des circuits de taille polynomiale et de profondeur logarithmique par rapport à la taille de l'entrée, et qui n'utilisent que des portes d'arité 2 (ce qui correspond à la classe de complexité* NC^1 *[184]). Ainsi, le circuit implémentant la distance euclidienne est de taille* $O(d)$ *et de profondeur* $O(\log d)$.

Lemme 5.3 (Profondeur d'un circuit quantique implémentant une distance). *Tout circuit classique irréversible implémentant une distance qui utilise uniquement des portes d'arité 2 tel que* FANOUT, AND *et* OR *peut être implémenté par un circuit quantique (et donc réversible) de la même taille et profondeur.*

Démonstration. Comme vu à la fin de la preuve du lemme 5.2, pour rendre le circuit classique réversible, il suffit de remplacer les portes FANOUT, AND et OR d'arité 2 par une ou deux applications de la porte de Toffoli. De plus, chaque porte de Toffoli peut

être simulée par un nombre constant de portes quantiques binaires et unaires, résultant ainsi en un circuit quantique implémentant la distance qui est de même profondeur et de même taille. □

Théorème 5.3. *Si la distance utilisée est calculable par un circuit de taille $O(d)$ et de profondeur $O(\log d)$, il existe un circuit quantique implémentant physiquement l'oracle de distance* O *(figure 5.2) qui est de taille $O(nd + n \log n)$ et de profondeur $O(\log(nd))$, où n est le nombre de points de l'ensemble de données et d le nombre d'attributs servant à les décrire (on suppose les attributs de taille constante). De plus, ce circuit peut être construit en un temps proportionnel à sa taille.*

Démonstration. Le circuit qui jouera le rôle de l'oracle de distance O va être construit à partir de 2 sous-circuits ; un sous-circuit E qui à partir de l'index i d'un point produit sa description x_i, et un circuit D qui à partir de la description de deux points x_i et x_j calcule leur distance $Dist(x_i, x_j)$. Si on met en entrée deux index i et j, et qu'on applique successivement deux copies de E, puis D et enfin deux copies de $\mathsf{E}^\dagger$ (la transposée conjuguée de E)[18] on obtient la superposition $|i, j, Dist(x_i, x_j)\rangle$, ce qui correspond exactement à réaliser l'oracle O (figure 5.10). De par le lemme 5.2, il est possible de construire

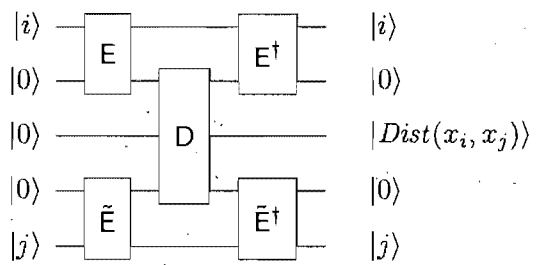

FIG. 5.10 – Circuit implémentant l'oracle de distance O. Le circuit E encode l'ensemble de données et D est le circuit calculant la distance. $\tilde{\mathsf{E}}$ est simplement le circuit E avec les deux registres d'entrée qui ont été échangés alors que $\mathsf{E}^\dagger$ et $\tilde{\mathsf{E}}^\dagger$ sont les transposées conjuguées de respectivement E et $\tilde{\mathsf{E}}$.

un circuit E faisant l'encodage de l'ensemble de données qui est de taille $O(nd + n \log n)$ et de profondeur $O(\log(nd))$. De plus, la combinaison de la remarque 5.1 et du lemme 5.3 implique que pour la plupart des distances couramment utilisées en apprentissage machine, il existe un circuit quantique D calculant la distance entre deux descriptions de points qui est de taille $O(d)$ et de profondeur $O(\log d)$. Le circuit quantique final qui

[18] $\mathsf{E}^\dagger$ peut être obtenu en "déroulant" le circuit E à l'envers si c'est physiquement possible.

jouera le rôle de l'oracle O et qui se compose des sous-circuits E et D aura donc une taille de $O(nd + n \log n)$ et une profondeur de $O(\log(nd))$. Le temps nécessaire pour construire ce circuit est directement proportionnel à sa taille. □

Cette construction est très efficace car relativement proche de la borne inférieure de $\Omega(nd)$ sur la taille du circuit et le temps nécessaire pour le réaliser. De plus, le circuit est lui aussi efficace car de profondeur logarithmique en n et d. Lors de l'analyse du temps d'exécution des algorithmes d'apprentissage non-supervisé dans les prochaines sections, nous passerons sous silence ces facteurs logarithmiques afin de faciliter l'analyse et parce qu'ils influencent peu le temps d'exécution final. Il faut cependant être conscient que ces facteurs logarithmiques sont toujours présents, ce qui conduit à formuler les trois questions ouvertes suivantes.

Question ouverte 5.2 (Borne inférieure de la construction du circuit O). *Est-il possible de construire un circuit quantique pour* O *en un temps équivalent à parcourir l'ensemble de données* $O(dn)$ *?*

Question ouverte 5.3 (Profondeur optimale pour le circuit O). *Est-il possible de réduire la dépendance de la profondeur du circuit* O *sur* n *(le nombre de points) ou* d *(le nombre d'attributs) afin de réduire la profondeur du circuit, et cela sans faire exploser sa taille ?*

Question ouverte 5.4 (Taille optimale pour le circuit O). *Est-il possible d'atteindre la borne inférieure en réduisant la taille du circuit* O *à* $O(dn)$ *?*

5.4 Catégorisation par arbre couvrant minimal

Soit $G = \langle S, A \rangle$ un graphe connexe orienté, où S est l'ensemble des n sommets du graphe et A l'ensemble des arêtes. Chaque arête comporte un poids qui correspond à une valeur réelle positive.

Définition 5.4 (Arbre couvrant). *Un* arbre couvrant *est un sous-ensemble de* $n - 1$ *arêtes* $T \subseteq A$ *tel que* $\langle S, T \rangle$ *forme un graphe connexe.*

Définition 5.5 (Arbre couvrant minimal). *Un* arbre couvrant minimal *est un arbre couvrant dont la somme des poids des arêtes est minimale parmi tous les arbres couvrants possibles.*

Une des plus anciennes techniques de catégorisation [196] se base directement sur l'*arbre couvrant minimal*. En effet, supposons que chaque point de données x_i de l'ensemble d'entraînement constitue le sommet d'un graphe et que chaque paire de sommets (x_i, x_j) est reliée par une arête dont le poids est proportionnel à une certaine mesure de distance $Dist(x_i, x_j)$. Une fois qu'un arbre couvrant minimal de ce graphe a été construit, il est facile de grouper les points de données en k catégories en enlevant tout simplement les $k-1$ plus longues arêtes de cette arbre.

La catégorisation basée sur l'arbre couvrant minimal maximise un critère qui tient compte de la distance minimum entre chaque catégorie.

Définition 5.6 (Espacement). *Soit C_1 et C_2 deux catégories disjointes, l'*espacement[19] *entre C_1 et C_2 est défini comme :*

$$espacement(C_1, C_2) = Dist_{min}(C_1, C_2) = \min_{x \in C_1, y \in C_2} Dist(x, y) \tag{5.3}$$

c'est-à-dire la distance entre la paire de points (x, y) les plus proches, pour x appartenant à la première catégorie et y appartenant à la seconde catégorie.

Définition 5.7 (k-catégorisation d'espacement maximum). *Une* k-catégorisation d'espacement maximum *de l'ensemble de données D_n est un ensemble de k catégories disjointes $C_1, C_2, \ldots, C_k$ formant une partition de D_n, et tel que :*

$$k\text{-}categorisation\_espacement\_maximum(D_n) = \arg\max_{C_1, C_2, \ldots, C_k} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} espacement(C_i, C_j) \tag{5.4}$$

La catégorisation basée sur l'arbre couvrant minimal est précisément celle qui maximise ce critère de catégorisation d'espacement maximum [101]. Il s'agit donc d'une situation où il existe un algorithme polynomial permettant de maximiser le critère de catégorisation utilisé, ce qui n'est pas le cas de nombreuses situations de catégorisation.

Classiquement, lorsque le graphe est représenté sous forme de matrice d'adjacence, ce problème de catégorisation se résout directement en utilisant l'algorithme de Prim [159] qui a un temps d'exécution de $O(n^2)$, où n est le nombre de sommets du graphe (ou

[19] L'espacement est aussi parfois appelé *distance minimale entre deux catégories*, ou aussi *single linkage* en anglais.

de façon équivalente le nombre de points de l'ensemble de données). Cet algorithme est optimal puisque dans le cas d'un graphe complet et d'une métrique arbitraire, tous les algorithmes classiques possibles requièrent un temps de $\Omega(n^2)$.

Le premier algorithme de catégorisation quantique, bien qu'il n'ait pas été développé dans ce but, est dû à Dürr, Heiligman, Høyer et Mhalla [77]. Ces chercheurs ont étudié la complexité quantique en terme de requêtes de certains problèmes de graphes, dont l'arbre couvrant minimal. Les modèles considérés dans leurs travaux sont les variantes quantiques de la *matrice d'adjacence* et de la *liste d'adjacence*. Le modèle de matrice d'adjacence est comparable à celui de boîte noire décrit dans la section 5.3.1. Ainsi dans ce modèle, on précise les index i et j de deux sommets et on reçoit en sortie le poids de l'arête entre ces deux sommets (ce qui, pour notre application, correspond à la distance).

En pratique, il faut absolument éviter de construire un circuit quantique qui garde en mémoire explicitement toutes les valeurs possibles de la matrice d'adjacence. En effet, un algorithme qui prendrait cette matrice d'adjacence de taille n par n aurait un temps d'exécution d'au moins $\Omega(n^2)$ car il devrait lire au moins une fois chaque entrée de la matrice, ce qui ne serait pas mieux que d'utiliser directement l'algorithme de Prim. Intuitivement, les situations où il est intéressant d'utiliser l'algorithme de Grover (ou une de ses variantes), sont celles où il est possible de construire un circuit quantique définissant l'espace de recherche en moins de temps qu'il est nécessaire classiquement pour rechercher un élément.

L'algorithme de Dürr, Heiligman, Høyer et Mhalla [77] est une quantisation d'un algorithme classique pour l'arbre couvrant minimal dû à Borůvka [40].

Théorème 5.4 (Algorithme quantique pour l'arbre couvrant minimal [77]). *Avec probabilité élevée, la quantisation de l'algorithme de Borůvka due à Dürr, Heiligman, Høyer et Mhalla est capable de trouver l'arbre couvrant minimal d'un graphe en un temps de $\Theta(n^{3/2})$, où n est le nombre de sommets du graphe. De plus, cet algorithme est optimal, c'est-à-dire que dans le cas d'un graphe complet il n'existe pas d'algorithme quantique pouvant trouver l'arbre couvrant minimal en moins de $\Omega(n^{3/2})$ appels à l'oracle.*

Le théorème 5.4 implique qu'en utilisant la version quantisée [77] de l'algorithme pour l'arbre couvrant minimal de Borůvka, on peut trouver la catégorisation qui minimise le critère de k-catégorisation d'espacement maximum en un temps de $\Theta(n^{3/2})$ (contre $\Theta(n^2)$

classiquement), pour n le nombre de points de l'ensemble de données.[20]

5.5 Catégorisation divisive

Une des manières les plus simples de construire une hiérarchie de catégories est de commencer par assigner tous les points à la même super-catégorie. La *catégorisation divisive* sépare ensuite cette super-catégorie en deux sous-catégories. Pour cela, deux points de données sont choisis pour former les *germes* des deux nouvelles sous-catégories. Un choix pertinent de germes est de prendre les deux points qui sont les plus éloignés dans l'ensemble. Une fois ce choix effectué, tous les autres points sont attachés à leur germe le plus proche. Cette technique de division est ensuite appliquée récursivement sur les sous-catégories obtenues jusqu'à ce que tous les points contenus dans une catégorie soient considérés suffisamment similaires ou qu'un critère d'arrêt soit atteint. Voir l'algorithme 4 pour plus de détails.

La catégorisation divisive cherche à maximiser à chaque étape de la récursion le critère de distance maximale entre les catégories.

Définition 5.8 (Distance maximale entre deux catégories). *La* distance maximale entre deux catégories *est la distance entre la paire de points (x, y) les plus éloignés, pour x appartenant à la première catégorie et y appartenant à la seconde catégorie.*

$$Dist_{max}(C_1, C_2) = \max_{x \in C_1, y \in C_2} Dist(x, y) \tag{5.5}$$

La partie la plus coûteuse de cet algorithme est de trouver les deux points qui sont les plus éloignés dans l'ensemble initial des n points (ce qui est équivalent à calculer le diamètre de cet ensemble de données). Si les points de données sont des vecteurs dans $\mathbb{R}^d$ pour une haute dimension arbitraire d, ce processus requiert généralement $O(n^2)$ comparaisons[21], et ce même si on est prêt à se contenter d'une approximation [86]. Quantiquement, il est possible d'utiliser quant_trouver_max comme sous-routine dans cet algorithme

[20] L'analyse originale de Dürr, Heiligman, Høyer et Mhalla considère implicitement que l'oracle à un coût d'appel constant puisque c'est la complexité en terme de requêtes pour les problèmes de graphes que ces chercheurs ont caractérisé. Dans notre cas, comme précisé dans la section 5.3.3, chaque appel à l'oracle à un coût "caché" qui est logarithmique en n et d.

[21] Cependant, si d est petit (comme $d = 1, 2$ ou 3) et qu'on utilise une métrique telle que la distance euclidienne, des algorithmes linéaires ou sous-quadratiques existent [31, 158].

Algorithme 4 quant_categorisation_divisive(D)

Si les points dans D sont suffisamment similaires **alors**
  Retourner D comme une catégorie
Sinon
  Trouver les deux points les plus éloignés x_a et x_b dans D en utilisant quant_trouver_max
  Pour chaque $x \in D$ **faire**
    Rattacher x au point de données le plus proche entre x_a et x_b
  fin pour
  Soit D_a l'ensemble de tous les points rattachés à x_a
  Soit D_b l'ensemble de tous les points rattachés à x_b
  Appeler quant_categorisation_divisive(D_a)
  Appeler quant_categorisation_divisive(D_b)
fin si

pour trouver les deux points les plus éloignés en temps $O(n)$.

Théorème 5.5 (Catégorisation divisive quantique). *Avec probabilité élevée, le gain en complexité de temps de l'algorithme* quant_categorisation_divisive, *qui permet de réaliser la catégorisation divisive d'un ensemble de n points, comparé à sa version classique est entre* $O(\frac{n}{\log n})$ *et* $O(n)$.

Démonstration. Supposons qu'à chaque appel récursif, l'algorithme sépare l'ensemble de données en deux sous-catégories d'approximativement la même taille. Ceci conduit à la construction d'un arbre équilibré et l'algorithme aura un temps global d'exécution $T(n)$ caractérisé par la récurrence asymptotique $T(n) = 2T(n/2) + O(n)$, qui est de $O(n \log n)$. Classiquement, la récurrence pour le même cas est de $T(n) = 2T(n/2) + O(n^2)$ à cause du temps nécessaire pour trouver les deux points les plus éloignés dans l'ensemble de points, ce qui revient à $O(n^2)$. Le gain entre le classique et le quantique est donc de $O(\frac{n^2}{n \log n}) = O(\frac{n}{\log n})$. À l'inverse dans le cas déséquilibré où l'algorithme produit une catégorie qui contient un petit nombre de points et une autre qui concentre la masse globale, l'arbre généré sera déséquilibré et de profondeur n. Dans ce cas-là, quantiquement il faudra un temps de calcul global de $O(n^2)$ contre $O(n^3)$ classiquement, ce qui conduit là aussi à un gain de $O(\frac{n^3}{n^2}) = O(n)$. Pour n'importe quelle situation entre ces deux extrêmes, le gain entre les version classiques et quantiques de l'algorithme de catégorisation divisive sera donc entre $O(\frac{n}{\log n})$ et $O(n)$. □

En pratique, si jamais les catégories générées ne sont pas équilibrées, cela peut vouloir dire que l'ensemble de données contient des *anomalies* (appelés *outliers* en anglais). Dans ce cas, la technique usuelle consiste à détecter et enlever ces anomalies avant de lancer la catégorisation divisive. Ceci permettra d'éviter normalement la formation de sous-catégories trop déséquilibrées en taille. Voir par exemple la section 5.7.2 pour la version quantisée d'un algorithme permettant de détecter les anomalies.

5.6 k-médianes

Cette section décrit deux versions quantiques de l'algorithme des k-médianes : la version "standard", où toutes les données sont physiquement rassemblées au même endroit, et la version *distribuée* où les données sont partagées entre deux ou plusieurs participants.

5.6.1 Version standard

L'algorithme des k-médianes, parfois appelé k-médoïdes [125], est un cousin de l'algorithme des k-moyennes. Il s'agit d'un algorithme itératif, où chaque itération se décompose en deux étapes. Lors de la première étape, chaque point de données est rattaché à son centre le plus proche. Durant la seconde étape, le centre de chaque catégorie est mis à jour comme étant le point médian parmi tous les points composant cette catégorie (c'est-à-dire le point qui est à distance totale minimale des autres points de la catégorie). L'algorithme s'arrête lorsque les centres des catégories sont stabilisés (ou quasi-stabilisés). Les centres des catégories sont souvent initialisés comme étant k points choisis au hasard parmi les n points de l'ensemble de données, où k est un paramètre de l'algorithme, qui correspond au nombre de catégories désirées. Nous verrons cependant dans la section 5.7.3 un algorithme quantique permettant d'initialiser les catégories de manière plus "intelligente".

L'algorithme des k-médianes cherche à partitionner les données en k catégories qui minimisent un critère de distance entre les points d'une catégorie et le centre de la catégorie.

Définition 5.9 (Critère des k-médianes). *Soit un ensemble de k catégories disjointes $C_1, C_2, \ldots, C_k$ qui ont pour centres respectifs les points $\mu_1, \mu_2, \ldots, \mu_k \in D_n$. Ces*

catégories sont optimales par rapport au critère de catégorisation des k-médianes si :

$$k\text{-}medianes(D_n) = \underset{(C_1,\mu_1),(C_2,\mu_2),\ldots,(C_k,\mu_k)}{\arg\min} \sum_{i=1}^{k} \sum_{x \in C_i} Dist(x, \mu_i) \tag{5.6}$$

Contrairement aux critères utilisés dans les algorithmes de catégorisation présentés précédemment, ce critère est NP-ardu à calculer dès que $k \geq 2$ [153]. Ainsi, on ne connaît pas d'algorithme polynomial permettant de trouver la solution optimale à ce problème, c'est-à-dire un ensemble de catégories qui minimise le critère 5.6. La version quantique de l'algorithme des k-médianes correspond donc à une version quantisée d'une heuristique servant à résoudre un problème NP-ardu, et non pas à un algorithme permettant de résoudre ce problème exactement.

La différence principale entre k-moyennes et k-médianes est que l'algorithme des k-moyennes utilise comme représentant d'une catégorie un point virtuel appelé *centroïde*, qui correspond à la moyenne des points à l'intérieur de la catégorie. Par contraste, l'algorithme des k-médianes est restreint à utiliser un point "réel" de l'ensemble de données comme centre d'une catégorie. Alors que l'algorithme des k-moyennes est sûr de converger vers un assignation des centres des catégories qui est stable après un nombre fini d'itérations, l'algorithme des k-médianes pourrait arriver dans une situation où il oscille entre deux ou plusieurs configurations. Cette différence de comportement de convergence découle du fait que la moyenne d'un ensemble de points est toujours définie de façon unique, alors qu'il pourrait y avoir plusieurs médianes valides pour un même ensemble de points. Cependant, un des avantages des k-médianes sur les k-moyennes est qu'on peut l'utiliser même lorsque la seule information disponible sur les points est leurs distances les uns des autres, ce qui rend impossible le calcul explicite de la moyenne, et donc l'application de l'algorithme des k-moyennes. Par rapport à son cousin plus connu, l'algorithme des k-médianes offre aussi l'avantage d'être généralement plus robuste au bruit et moins sensible à la présence d'anomalies dans les données (voir la section 5.7.2 pour une brève explication).

Théorème 5.6 (k-médianes standard quantique). *Avec probabilité élevée, l'algorithme* quant_k−medianes *permet de réaliser la catégorisation d'un ensemble de n points en un temps de* $O(\frac{1}{\sqrt{k}} n^{3/2} t)$ *et avec un gain de* $O(\sqrt{\frac{n}{k}})$ *comparé à sa version classique, où k*

Algorithme 5 quant_$k-$medianes(D_n,k)

Choisir k points uniformément au hasard comme étant les centres initiaux des catégories
Répéter
  Pour chaque point de données dans D_n **faire**
    Rattacher ce point à son centre le plus proche
  fin pour
  Pour chaque catégorie Q **faire**
    Calculer la médiane de cette catégorie en utilisant quant_trouver_mediane(Q) et en faire son nouveau centre
  fin pour
Jusqu'à (quasi-)stabilisation des catégories
Retourner les catégories trouvées et leurs centres

est le nombre de catégories retournées et t le nombre d'itérations de l'algorithme.

Démonstration. Afin d'analyser l'efficacité d'une itération de l'algorithme, supposons que les catégories sont approximativement de tailles comparables, c'est-à-dire n/k. Si les médianes étaient calculées de façon classique, chacune demanderait un temps de calcul de $O((\frac{n}{k})^2)$, avec un total de $O(\frac{1}{k}n^2)$ pour trouver les centres des k catégories. De façon quantique, il est possible de trouver la médiane d'une catégorie de taille n/k en un temps de $O(\frac{n}{k}\sqrt{\frac{n}{k}})$ en utilisant la sous-routine quant_trouver_mediane. Ceci résulte en un temps de $O(\frac{1}{\sqrt{k}}n^{3/2})$ pour une itération de la version quantique de k-médianes, ce qui est $O(\sqrt{n/k})$ fois plus rapide que l'approche classique. Pour le cas déséquilibré, considérons par exemple le scénario où quasiment toutes les catégories sont de taille c, pour c une petite constante, alors qu'une unique catégorie concentre toute la masse des points. Dans ce scénario, trouver la médiane demanderait $O(n^2)$ requêtes à l'oracle classique contre $O(n^{3/2})$ quantiquement. Le gain entre le classique et le quantique persiste donc même dans le cas déséquilibré. À noter que d'utiliser la version quantique ou classique de l'algorithme n'influence pas la vitesse de convergence de l'algorithme et que celle-ci dépend de t, le nombre d'itérations[22]. □

Entre deux itérations de l'algorithme de k-médianes, il peut arriver que les catégories restent relativement stables, c'est-à-dire que seul un petit nombre de points changent

[22]En pratique cependant, le nombre d'itérations et la qualité de la catégorisation retournée peuvent être améliorés par une initialisation "intelligente" du centre des catégories (voir section 5.7.3).

de catégories. Dans ce cas-là, il est possible qu'une structure de données (classique) appropriée puisse garder la trace des catégories et de leurs médianes afin d'accélérer la recherche des points médians d'une itération à une autre. Ainsi, en pratique on pourrait d'abord utiliser la version quantique de k-médianes pour les premières itérations avant d'utiliser ensuite la variante classique basée sur la structure de données appropriée.

Il est aussi possible que la version quantisée de k-médianes puisse être encore améliorée en développant un algorithme quantique qui estimerait la somme d'un ensemble de valeurs plutôt que simplement les ajouter une par une comme décrit dans la figure 5.5. Les algorithmes existants actuellement pour estimer la moyenne [104] ne semblent pas pouvoir être utilisés directement à cause de problèmes liés à leur précision, mais d'autres méthodes basées sur l'*estimation d'amplitude* semblent prometteuses [46]. La convergence de l'algorithme, mesuré par le nombre d'itérations t, pourrait être aussi accélérée comme l'énonce la question ouverte suivante.

Question ouverte 5.5 (Accélération quadratique du nombre d'itérations). *Peut-on quantiser encore plus en profondeur l'algorithme des k-médianes afin de réduire le nombre d'itérations t qui sont nécessaires à l'algorithme (par exemple par un facteur quadratique) ?*

5.6.2 Version distribuée

De même qu'on peut parfois accélérer le temps de calcul d'un algorithme en utilisant le paradigme de l'informatique quantique, il est possible dans certains cas de réduire le coût de communication de sa version distribuée. Dans une situation d'apprentissage distribué, l'ensemble de données D_n n'est pas localisé dans un seul endroit mais est distribué aux mains de deux ou plusieurs participants[23]. Leur but est de faire fonctionner un algorithme d'apprentissage sur cet ensemble de données au complet. Ce problème pourrait être résolu très simplement en rapatriant toutes les données sur un site central et en faisant fonctionner une version standard de l'algorithme d'apprentissage. Dans ce cas-ci, le coût de communication du protocole résultant serait de $\Theta(dn)$ bits, ce qui peut être très lourd en pratique si la taille de l'ensemble de données est importante. Tout le

[23] De manière équivalente, on peut voir cette situation comme si chaque participant i avait son propre ensemble de données D_i et le but est de faire fonctionner un algorithme d'apprentissage sur l'union de leurs ensembles de données.

but de la complexité de la communication [132] est d'essayer de trouver des protocoles efficaces requérant moins de bits de communication que nécessaires pour communiquer l'entrée.

Dans le cas de l'apprentissage distribué quantique, on permet aux participants de s'échanger des qubits au lieu de bits ou encore de partager de l'intrication au préalable afin de les aider dans leur tâche. Peut-on gagner quelque chose à utiliser de l'information quantique lors de la réalisation de tâches distribuées, par exemple en économisant sur le coût de la communication pour certains protocoles ? C'est à cette question que s'intéresse la *complexité de la communication quantique* [43, 189]. Comme vu dans la section 2.3, du côté des résultats *a priori* négatifs, l'existence du théorème de Holevo [115] exclut la possibilité de transmettre plus de n bits d'information en utilisant n qubits. Si jamais les participants partagent préalablement de l'intrication sous la forme de n paires EPR alors il est possible de transmettre $2n$ bits classiques en utilisant le codage superdense [26], mais il s'agit-là du mieux qu'on peut faire [59]. De même, il est impossible d'utiliser l'intrication pour communiquer car cela signifierait pouvoir communiquer plus rapidement que la vitesse de la lumière et violerait le principe de causalité. Par contre, ces limites n'interdisent pas qu'il soit possible, en utilisant de l'information quantique, d'économiser significativement (quadratiquement ou exponentiellement) sur le coût de communication de certaines fonctions distribuées [43, 189] ou encore de réaliser des tâches impossibles classiquement dans un contexte où la communication entre les participants n'est pas permise [45].

Supposons que l'ensemble de données D_n soit partagé entre deux participants, Alice et Bob. Pour la simplicité de l'analyse, faisons l'hypothèse qu'Alice possède exactement la moitié des n points de données, et Bob l'autre moitié, pour n un nombre pair[24]. Pour que la version quantique distribuée de k-médianes soit intéressante, il faut que son coût de communication mesuré en qubits échangés soit inférieur au protocole trivial qui consisterait à envoyer classiquement toutes les données sur un seul site (et qui aurait un coût de $\Theta(dn)$ bits). Les points de D_n peuvent être réordonnés de manière à ce que les

[24]Si cette hypothèse n'est pas vérifiée, il est possible d'adapter sans difficulté l'algorithme à cette situation. Toutefois si les tailles des ensembles de données de Alice et de Bob sont vraiment déséquilibrées l'une par rapport à l'autre (par exemple si D_a ou D_b est de taille $O(\sqrt{n})$), le mieux à faire est que celui qui a le plus petit ensemble envoie directement toutes ses données à l'autre participant et que celui-ci utilise la version standard de l'algorithme sur l'ensemble de ces données.

$\frac{n}{2}$ premiers points correspondent à l'ensemble de données d'Alice $D_a = \{x_1, \ldots, x_{\frac{n}{2}}\}$, et le reste des points à l'ensemble de données de Bob $D_b = \{x_{\frac{n}{2}+1}, \ldots, x_n\}$. En utilisant la même prescription que dans la preuve du lemme 5.2, Alice peut produire, à partir de son ensemble de données D_a, un circuit quantique E_a de taille $O(nd + n\log n)$ et de profondeur $O(\log(nd))$ qui encode son ensemble de données. Ce circuit E_a prend en entrée l'index i d'un point de son ensemble, pour $i \in \{1, \ldots, \frac{n}{2}\}$, et produit la description x_i de ce point. Bob fait de même avec son ensemble de données D_b et construit un circuit E_b, qui, à partir d'un index $j \in \{\frac{n}{2}+1, \ldots, n\}$, retourne la description x_j de ce point. De plus, si Alice dispose d'une implémentation quantique du circuit D permettant de calculer la distance de deux points passés en entrée, elle peut l'utiliser en conjonction avec E_a pour implémenter l'oracle $\mathsf{S_a}$ qui calcule la somme totale des distances entre un point et tous les autres points de son ensemble D_a (même remarque pour Bob).

Afin de réaliser l'algorithme des k-médianes de manière distribuée, Alice et Bob vont devoir implémenter de façon distribuée l'itération de Grover utilisée dans la sous-routine quant_trouver_mediane. Durant ce protocole, ils vont utiliser et échanger différents registres quantiques. L'indice A et l'indice B sont rattachés aux registres quantiques pour indiquer qui, de respectivement Alice ou Bob, en a le contrôle à un moment particulier du protocole. Soit i_a l'index du point de l'ensemble de données D_a dont la somme des distances avec les points de D_n est minimale (i_b est défini de manière symétrique D_b), c'est-à-dire :

$$i_a = \underset{x_i \in D_a}{\arg\min} \sum_{j=1}^{n} Dist(x_i, x_j) \tag{5.7}$$

L'itération de Grover distribuée (que nous nommerons iteration_mediane_distribuee) lorsqu'on cherche i_a peut être réalisée par l'algorithme 6.

Lemme 5.4. *Le protocole* iteration_mediane_distribuee *(algorithme 6) réalise l'itération de Grover de façon distribuée dans la sous-routine* quant_trouver_mediane. *Son coût de communication est de* $O(d + \log(nd_{max}))$ *qubits et sa complexité en terme de calcul de* $O(n)$, *où* d *est le nombre d'attributs servant à décrire les points de données,* n *est le nombre de points de données et* d_{max} *est la distance maximale entre deux points de données.*

Démonstration. La description de l'algorithme 6 étape par étape démontre clairement

Algorithme 6 iteration_mediane_distribuee(D_a,D_b)

[Étape 1] Alice utilise le circuit E_a pour produire la description x_i d'un point à partir de son index i.
$|i\rangle_A |0^{\otimes O(d)}\rangle_A \overset{\mathsf{E}_a}{\mapsto} |i\rangle_A |x_i\rangle_A$
(Le registre d'index (registre 1) est de taille $O(\log n)$ qubits et le registre contenant la description du point x_i (registre 2) est de taille $O(d)$.)
[Étape 2] Alice fait appel au circuit S_a pour calculer la somme des distances entre le point x_i et tous les autres points de D_a.
$|i\rangle_A |x_i\rangle_A |0^{\otimes O(\log(nd_{max}))}\rangle_A \overset{\mathsf{S}_a}{\mapsto} |i\rangle_A |x_i\rangle_A |\sum_{j=1}^{\frac{n}{2}-1} Dist(x_i,x_j)\rangle_A$
(Le registre de distance (registre 3) est de taille proportionnelle au logarithme de la distance maximum entre deux points d_{max}, plus un terme logarithmique en le nombre de points.)
[Étape 3] Alice envoie le registre quantique contenant la description de x_i ainsi que le registre de distance à Bob.
$|i\rangle_A |x_i\rangle_A |\sum_{j=1}^{\frac{n}{2}} Dist(x_i,x_j)\rangle_A \overset{com}{\mapsto} |i\rangle_A |x_i\rangle_B |\sum_{j=1}^{\frac{n}{2}} Dist(x_i,x_j)\rangle_B$
[Étape 4] Bob fait interagir le registre contenant la description du point d'Alice avec la description de ses points en utilisant l'oracle S_b. Ceci permet de calculer la somme des distances entre x_i et tous les autres points de D_n.
$|i\rangle_A |x_i\rangle_B |\sum_{j=1}^{\frac{n}{2}} Dist(x_i,x_j)\rangle_B \overset{\mathsf{S}_b}{\mapsto} |i\rangle_A |x_i\rangle_B |\sum_{j=1}^{n} Dist(x_i,x_j)\rangle_B$
[Étape 5] Soit f la fonction booléenne qui instancie la fonction indicatrice $I\{\sum_{j=1}^{n} Dist(x_i,x_j) < d_{min}\}$ et qui pour un index i est égale à

$$f(i) = \begin{cases} 1 & \text{si } \sum_{j=1}^{n} Dist(x_i,x_j) < d_{min} \\ 0 & \text{sinon} \end{cases} \tag{5.8}$$

Bob applique le changement de phase conditionnel P qui réalise la transformation suivante :
$|i\rangle_A |x_i\rangle_B |\sum_{j=1}^{n} Dist(x_i,x_j)\rangle_B \overset{\mathsf{P}}{\mapsto} (-1)^{f(i)} |i\rangle_A |x_i\rangle_B |\sum_{j=1}^{n} Dist(x_i,x_j)\rangle_B$
[Étape 6] Bob applique $\mathsf{S}_b^\dagger$ et renvoie le registre contenant la description de x_i et le registre de distance à Alice.
$(-1)^{f(i)} |i\rangle_A |x_i\rangle_B |\sum_{j=1}^{n} Dist(x_i,x_j)\rangle_B \overset{\mathsf{S}_b^\dagger + com}{\mapsto} (-1)^{f(i)} |i\rangle_A |x_i\rangle_A |\sum_{j=1}^{\frac{n}{2}} Dist(x_i,x_j)\rangle_A$
[Étape 7] Alice désintrique le registre de distance en appliquant $\mathsf{S}_A^\dagger$.
$(-1)^{f(i)} |i\rangle_A |x_i\rangle_A |\sum_{j=1}^{\frac{n}{2}} Dist(x_i,x_j)\rangle_A \overset{\mathsf{S}_a^\dagger}{\mapsto} (-1)^{f(i)} |i\rangle_A |x_i\rangle_A$
[Étape 8] Alice désintrique son registre de description en appliquant $\mathsf{E}_A^\dagger$.
$(-1)^{f(i)} |i\rangle_A |x_i\rangle_A \overset{\mathsf{E}_a^\dagger}{\mapsto} (-1)^{f(i)} |i\rangle_A$
[Étape 9] Alice applique localement l'inversion par rapport à la moyenne sur son registre d'index qui a l'effet suivant :

$$|x\rangle \mapsto \begin{cases} |x\rangle & \text{si } x = 0^{\otimes \log n} \\ -|x\rangle & \text{sinon} \end{cases} \tag{5.9}$$

que celui-ci réalise la fonctionnalité de l'itération de Grover telle que décrite dans la section 5.2. En particulier, l'effet de la première partie de l'itération de Grover (étapes 1 à 5) est d'inverser la phase de tous les points dans D_a dont la somme des distances avec les autres points est inférieure à la valeur d_{min}. L'inversion par rapport à la moyenne, qui est effectuée en dernière étape de l'itération de Grover, est indépendante de la fonctionnalité considérée et peut être réalisée localement par Alice. En ce qui concerne la communication, seules les étapes 3 et 6 nécessitent un échange d'information entre Alice et Bob. Les tailles des registres échangées sont respectivement de $O(d)$ pour le registre de description et de $O(\log(nd_{max}))$ pour le registre de distance, où d_{max} est la distance maximale entre deux points de D_n. Le coût global de communication est donc de $O(d + \log(nd_{max}))$ qubits. La complexité en temps du calcul de l'itération dépend directement des temps nécessaires à appliquer les circuits E_a, E_b, S_a et S_b, qui sont respectivement de $O(1)$ pour E_a et E_b et $O(n)$ pour S_a et S_b comme vu dans la section 5.3.3 (si on ne tient pas compte des facteurs logarithmiques supplémentaires). Le coût total de l'itération en terme de temps est donc dominé par $O(n)$. □

À chaque itération de l'algorithme des k-médianes et pour chaque catégorie, on appelle tout d'abord la sous-routine quant_trouver_mediane afin de trouver i_a en appliquant le protocole iteration_mediane_distribuee pour simuler l'itération de Grover. Alice initialise d_{min} comme étant la somme des distances entre un point i_0 choisit au hasard parmi Q_a et tous les autres points de D_n. Cette étape d'initialisation requiert qu'Alice communique (classiquement) la description de i_0 à Bob (coût de $O(d)$ bits) et que celui-ci lui envoie la somme des distances entre les points dans D_b et i_0 (coût de $O(d_{max} + \log m)$ bits). Avant d'utiliser iteration_mediane_distribuee pour la première fois, Alice commence par mettre le registre d'index en superposition de tous les index possibles grâce à une tour de Walsh–Hadamard. Une fois que i_a a été déterminé en utilisant quant_trouver_mediane, on détermine i_b en inversant le rôle de Bob et d'Alice. Enfin, pour déterminer la médiane parmi D_n, il suffit de choisir entre i_a et i_b celui des deux points dont la somme des distances avec tous les autres points est minimale. L'algorithme 7 formalise le déroulement de la version distribuée de k-médianes.

Théorème 5.7 (k-médianes distribué quantique). *Avec probabilité élevée, l'algorithme* quant_k−medianes_distribue *permet de réaliser la catégorisation d'un ensemble de n points*

Algorithme 7 quant_k−medianes_distribue(D_a,D_b,k)

Alice et Bob choisissent k points uniformément au hasard parmi D_a et D_b comme étant les centres initiaux des catégories
Répéter
  Pour chaque point de données dans D_a et D_b **faire**
    Rattacher ce point à son centre le plus proche
  fin pour
  Pour chaque catégorie Q **faire**
    Alice recherche i_a en utilisant quant_trouver_mediane(Q) avec le protocole iteration_mediane_distribuee pour simuler l'itération de Grover
    Bob recherche i_b en utilisant quant_trouver_mediane(Q) avec le protocole iteration_mediane_distribuee (les rôles de Alice et Bob sont échangés) pour simuler l'itération de Grover
    médiane(Q) = $\arg\min_{i\in\{i_a,i_b\}} \sum_{x_j\in Q} Dist(x_i, x_j)$
  fin pour
Jusqu'à (quasi-)stabilisation des catégories
Retourner les catégories trouvées et leurs centres

distribués entre deux participants en un temps de $O(t\frac{1}{\sqrt{k}}n^{3/2}))$ *et un coût de communication de* $O(t\sqrt{kn}(d+\log(nd_{max})))$ *qubits, où* n *est le nombre de points de l'ensemble de données,* d *le nombre d'attributs servant à les décrire,* k *le nombre de catégories retournées,* t *le nombre d'itérations de l'algorithme et* d_{max} *est la distance maximale entre deux points de données.*

Démonstration. L'analyse du temps d'exécution de la version distribuée de l'algorithme est exactement similaire à celui de sa version standard (théorème 5.6). De plus, de par le lemme 5.4, on peut observer que la quantité de travail en terme de calcul qui sera demandée à Alice et Bob pour la version distribuée est du même ordre de grandeur que pour la version standard, soit $O(\frac{1}{\sqrt{k}}n^{3/2})$ pour une itération de k-médianes. Si les catégories sont approximativement de même taille, chaque itération de l'algorithme demande à Alice et Bob d'échanger un registre de taille $O(d+\log(nd_{max}))$ un nombre de fois dans $O(k\sqrt{\frac{n}{k}})$ ce qui conduit à un coût de communication de $O(k\sqrt{\frac{n}{k}}(d+\log(nd_{max}))) = O(\sqrt{kn}(d+\log(nd_{max})))$ qubits. Pour t itérations, le coût global de communication est donc de $O(t\sqrt{kn}(d+\log(nd_{max})))$. □

Tant que le nombre t d'itérations de l'algorithme est inférieur à $\sqrt{n}$, la version distribuée de l'algorithme des k-médianes est plus économe en terme de communication

que le protocole naïf où toutes les données sont rassemblées dans un site central. De plus, l'algorithme peut être facilement généralisé au cas multipartite, pour un nombre de participants $m \geq 2$. Pour cela, il suffit que chaque participant j (pour $1 \leq j \leq m$) trouve parmi son ensemble de données D_j le point i_j dont la somme des distances avec tous les autres points de D_n est minimale. On peut réaliser cela en adaptant la version distribuée de l'itération de Grover afin qu'elle fonctionne avec plus de 2 participants. Il suffit ensuite de choisir la médiane comme étant le point parmi $\{i_1, \ldots, i_m\}$ qui minimise la somme des distances avec les autres points. Sans rentrer dans les détails, le coût de communication et la complexité de calcul de la version multipartite sont essentiellement identiques à ceux de la version bipartite.

5.7 Outils quantiques pour algorithmes d'apprentissage non-supervisé

Les algorithmes présentés dans cette section ne constituent pas des algorithmes d'apprentissage non-supervisé par eux-mêmes mais sont couramment utilisés comme outils par d'autres algorithmes d'apprentissage non-supervisé. Des variantes plus rapides de ces algorithmes contribuent donc directement à des variantes plus rapides d'autres algorithmes d'apprentissage non-supervisé.

5.7.1 Construction d'un graphe de voisinage

La construction d'un graphe de voisinage est une étape importante du prétraitement de plusieurs algorithmes d'apprentissage non-supervisé tels que l'algorithme de réduction de dimensionnalité Isomap [179] ou la catégorisation par marche aléatoire [106].

Définition 5.10 (Graphe de voisinage). *Soit un graphe complet dont les sommets correspondent aux points de l'ensemble de données, et où chaque arête entre deux sommets est pondérée en tenant compte de la distance entre ces deux sommets (voir section 5.4). Un* graphe de voisinage *est construit à partir de ce graphe original en gardant pour chaque sommet seulement les arêtes connectant ses k voisins les plus proches*[25].

[25] Une manière différente de définir le graphe de voisinage consiste relier deux points par une arête si et seulement si ils sont chacun dans le plus proche voisinage l'un de l'autre. Cette méthode garantit que le degré maximum du graphe sera égal ou inférieur à k, le nombre de plus proches voisins considérés. Un inconvénient cependant est que certains points pourraient se retrouver isolés (non-connectés) du reste du graphe, ce qui pourrait indiquer qu'il s'agit d'*anomalies* (voir section *5.7.2*).

L'algorithme 8 est une version quantisée d'un algorithme de construction d'un graphe de voisinage.

Algorithme 8 quant_construction_graphe_voisinage(D_n,k)

Pour chaque point de données x_i de D_n **faire**
  Utiliser quant_trouver_c_plus_proches_voisins pour trouver les k plus proches voisins de x_i
  Pour chacun des k plus proches voisins de x_i **faire**
    Créer une arête entre x_i et ce voisin qui est la distance entre ces deux points
  fin pour
fin pour
Retourner le graphe construit

Théorème 5.8 (Algorithme quantique pour la construction d'un graphe de voisinage). *Avec probabilité élevée, l'algorithme* quant_construction_graphe_voisinage *construit le graphe de voisinage d'un ensemble de n points en un temps de $O(\sqrt{k}n^{3/2})$, si on considère pour chaque point seulement un nombre constant k de ses voisins.*

Démonstration. Pour chaque point de données, il est possible de trouver ses k voisins les plus proches en un temps $O(\sqrt{kn})$ en utilisant la sous-routine quant_trouver_c_plus_proches_voisins. Afin de construire le graphe de voisinage global, le coût total sera de $O(\sqrt{k}n^{3/2})$. □

Classiquement, si on utilise une métrique arbitraire et que la seule information dont on dispose est la distance entre les paires de points, il faut un temps de $\Omega(n^2)$ pour générer le graphe de voisinage constitué par les voisins les plus proches de tous les n points. Cependant, si on a explicitement accès pour chaque point de données à tous les d attributs qui le décrivent et si d est de faible dimension [129][26], une structure de données appropriée telle que les *arbres binaires de recherche multidimensionnelle* [27] (aussi connus en anglais sous le nom de *kd-trees*[27]) permet de trouver les k plus proches

[26] L'article [129] décrit une étude comparant différentes méthodes et structures permettant d'accélérer la recherche des plus proches voisins. Cette étude observe empiriquement que dès $d \geq 16$, toutes les méthodes sont sensibles au fléau de la dimensionnalité et prennent un temps pire que linéaire dans le nombre de points pour trouver les plus proches voisins d'un point particulier.

[27] À l'origine, le terme "kd-tree" est une abréviation de l'expression "k-dimensional tree". Si on suivait scrupuleusement la notation de cette thèse, ces arbres auraient été nommés comme étant d-dimensionnels ce qui aurait conduit à les appeler "dd-trees" !

voisins d'un point spécifique en un temps de $\Theta(k \log n)$. La construction d'un kd-tree requiert de trier tous les points pour chaque dimension, ce qui peut être fait en temps $\Theta(dn \log n)$, où d est la dimension de l'espace dans lequel les points vivent et n le nombre de points de données. Le coût global pour construire le kd-tree représentant l'ensemble de données *et* l'utiliser pour trouver les k plus proches voisins de chacun des n points est donc de $\Theta((k+d)n \log n)$ si d est de faible dimension.

5.7.2 Détection d'anomalies

Définition 5.11 (Anomalie). *Une* anomalie *(appelée* outlier *en anglais) est une observation qui diffère de façon significative du reste des données.*

En pratique, le sens exact que prend cette définition dépend de l'application considérée. Dans beaucoup de domaines, les anomalies sont considérées comme étant des points de données générés suite à un phénomène de bruit et doivent être éliminés. Par exemple, une anomalie pourrait être un point de données qui a été corrompu, soit par la modification de certains de ses attributs, soit parce qu'il est étiqueté avec une mauvaise classe. Il se pourrait même qu'il s'agisse d'une observation générée aléatoirement.

Ainsi en classification, on veut détecter et enlever les anomalies présentes dans l'ensemble de données afin d'améliorer la précision du classifieur qui sera appris. De la même manière en catégorisation, pouvoir reconnaître les anomalies permet de ne pas en tenir compte lors de la formation des catégories et d'améliorer la qualité de la catégorisation retournée. Dans certaines applications cependant, comme la détection de fraudes au niveau de l'utilisation des cartes de crédit ou encore la détection d'intrusion, on cherche à détecter les anomalies car elles correspondent à un comportement inhabituel qu'il est particulièrement intéressant d'identifier.

Si on connaît la distribution qui a généré les données (grâce par exemple à un modèle appris par de l'estimation de densité ou encore par une connaissance *a priori*), un test statistique peut permettre d'identifier les points de données qui *dévient de façon statistiquement significative de cette distribution* [193]. Une autre manière de détecter une anomalie est d'inspecter les attributs de chaque objet et de considérer comme étant des anomalies ceux qui *diffèrent très largement de la valeur médiane*[28] [141]. D'autres

[28] On se base en général sur la médiane et non pas sur la moyenne pour mesurer cette déviance car celle-ci

méthodes se basant sur la densité [50] inspectent le voisinage de chaque point et considère les points situés dans des zones peu denses comme des anomalies potentielles. Cette approche donne de bons résultats en pratique mais est généralement coûteuse en temps de calcul. Finalement, une autre approche se base sur une *notion de distance* [12] et considère comme étant des anomalies les points qui sont à une distance élevée de leurs k plus proches voisins. Ainsi, une technique possible [12] consiste à identifier pour chaque point x_i, ses k plus proches voisins et à donner un *score de voisinage* ω_i à ce point qui est égal à

$$\omega_i = \sum_{j \in V_k(x_i)} Dist(x_i, x_j), \tag{5.10}$$

où $V_k(x_i)$ est le sous-ensemble des points dans D_n constitué des k plus proches voisins de x_i. Lorsque ce score a été calculé pour chaque point, les anomalies peuvent être identifiées comme les points dont le score ω est supérieur à un seuil ω_{max} déterminé empiriquement, ou bien comme les c points qui ont le score le plus élevé, où c est une constante bien choisie. L'algorithme 9 est une version quantique permettant de calculer ces scores pour chacun des points. Classiquement, il faudrait un temps de $\Omega(n^2)$ pour déterminer les scores de voisinage de chacun des n points (pour les mêmes raisons que celles évoquées pour la construction d'un graphe de voisinage dans la section 5.7.1).

Algorithme 9 quant_anomalie_detection_distance(D_n,k)

Pour chaque point de données x_i de D_n **faire**
  Utiliser quant_trouver_c_plus_proches_voisins pour trouver les k voisins les plus proches de x_i
  Calculer ω_i comme étant la somme des distances de x_i avec ses k plus proches voisins
fin pour
Retourner les scores calculées pour chaque point

Théorème 5.9 (Algorithme quantique pour la détection d'anomalies basé sur les distances). *Avec probabilité élevée, l'algorithme* quant_anomalie_detection_distance *permet d'identifier tous les points qui sont des anomalies, parmi un ensemble de données de*

est peu influencée par la présence d'anomalies, contrairement à la moyenne qui peut changer radicalement sous l'influence d'une valeur extrême. Ceci explique aussi pourquoi l'algorithme des k-médianes (présenté dans la section 5.6) est généralement considéré comme étant moins sensible à la présence d'anomalies que son cousin, l'algorithme des k-moyennes.

taille n, en un temps de $O(\sqrt{k}n^{3/2})$ si $k < \sqrt{n}$, pour k le nombre de voisins considérés pour chaque point.

Démonstration. Pour chaque point de données, on peut trouver ses k plus proches voisins en temps $O(\sqrt{kn})$ en utilisant quant_trouver_c_plus_proches_voisins et calculer son score de voisinage w en temps $O(k)$. Trouver les voisins et calculer ce score pour tous les n points requiert un temps global de $O(\sqrt{k}n^{3/2} + kn)$ ce qui se simplifie à $O(\sqrt{k}n^{3/2})$ si $k < \sqrt{n}$ (ce qui est typiquement le cas). □

5.7.3 Initialisation des centres des catégories

Traditionnellement, les centres initiaux des catégories dans des algorithmes tels que k-moyennes ou k-médianes sont choisis aléatoirement parmi les points de l'ensemble de données. À partir de deux configurations initiales différentes, l'algorithme a une probabilité non négligeable de converger vers deux catégorisations différentes. Comme la fonction de coût que k-moyennes ou k-médianes essayent d'optimiser est NP-ardu à calculer pour sa valeur optimale [153], il est probable que les deux catégorisations générées correspondront toutes les deux à des minima locaux. Une technique standard utilisée en catégorisation consiste à exécuter l'algorithme plusieurs fois à partir de configurations initiales différentes et à sauvegarder la catégorisation qui minimise la fonction de coût considérée.

Une approche différente consiste à choisir les germes initiaux des catégories de manière "intelligente" plutôt que de les tirer aléatoirement. Un algorithme de type *max-min* [65] commence par choisir le premier centre μ_1 au hasard parmi les points de l'ensemble de données. Ensuite, un second point de données μ_2 est choisi comme étant le point qui est à distance maximum de μ_1. Les centres suivants $\mu_3, \ldots, \mu_k$ sont déterminés en choisissant toujours comme nouveau centre le point de données dont la somme des distances avec les centres précédents est maximale. Formellement, on définit le $i^{\text{ème}}$ centre de catégorie générée μ_i comme étant :

$$\mu_i = \arg\max_{x \in D_n} \sum_{j=1}^{i-1} Dist(x, \mu_j) \tag{5.11}$$

Cette méthode produit des germes initiaux de catégories qui sont espacés les uns par rapport aux autres. L'algorithme 10 est une variante quantique de cette méthode d'ini-

tialisation des centres des catégories.

Algorithme 10 quant_initialisation_categorie(D_n,k)

Choisir aléatoirement un point de données dans D_n qui sera étiqueté comme étant μ_1
Pour $i = 2$ jusqu'à k **faire**
  Utiliser quant_trouver_max pour trouver $\mu_i = \arg\max_{x \in D_n} \sum_{j=1}^{i-1} Dist(x, \mu_j)$
fin pour
Retourner les centres initiaux des catégories $\mu_1, \ldots, \mu_k$

Théorème 5.10 (Algorithme quantique d'initialisation des centres des catégories). *Avec probabilité élevée, l'algorithme* quant_initialisation_categorie *permet d'initialiser de façon "intelligente" les k centres de catégories d'un ensemble de n points en un temps de $O(k^2\sqrt{n})$.*

Démonstration. L'algorithme quant_initialisation_categorie choisit le premier centre comme étant un point au hasard parmi l'ensemble de données. Le deuxième centre est ensuite déterminé comme étant le point le plus éloigné du centre original en utilisant la sous-routine quant_trouver_max, ce qui prend un temps de $O(\sqrt{n})$. Puis pour chacun des centres suivants μ_i, pour $3 \leq i \leq k$, on peut le calculer en mettant en superposition les index de tous les points ainsi que la somme des distances entre ces points et tous les $i-1$ centres déjà identifiés. On peut obtenir cette superposition en appliquant $i-1$ fois l'oracle O, et on applique ensuite la sous-routine quant_trouver_max sur le registre des distances générées pour un coût de $O((i-1)\sqrt{n})$. Le temps d'exécution global de l'algorithme $T(n)$ est donc directement proportionnel à la suite arithmétique $T(n) = \sum_{i=2}^{k} i\sqrt{n} = O(k^2\sqrt{n})$. □

Remarque 5.2 (Algorithme quantique pour une k-catégorisation de diamètre maximum minimal). *L'algorithme* quant_initialisation_categorie *peut aussi être utilisé pour trouver une catégorisation composée de k catégories disjointes où le diamètre maximum, parmi toutes ces catégories, est minimisé. Autrement dit, le but va être de former un ensemble de k catégories où les points sont tous très proches des centres des catégories (ce qui conduit à un diamètre maximum des catégories qui est faible). Formellement, le critère que cet algorithme cherche à minimiser est celui des k-centres qui se définit par*

$$k\text{-}centres(D_n) = \underset{(C_1,\mu_1),(C_2,\mu_2),\ldots,(C_k,\mu_k)}{\arg\min} \max_i \max_{x \in C_i} Dist(x, \mu_i) \tag{5.12}$$

*Ce critère est NP-ardu à optimiser dans sa version exacte, mais une approximation due à González [100] retourne un ensemble de k catégories dont le diamètre maximum est au plus deux fois celui de la k-*catégorisation optimale de diamètre maximum minimal. *L'algorithme de González commence par choisir les k centres des catégories grâce à la méthode* quant_initialisation_categorie *détaillée plus haut, ce qui classiquement requiert un temps de* $O(k^2n)$. *Ensuite, chaque point de l'ensemble de données est attachée à son centre le plus proche. La version quantisée de cet algorithme de catégorisation prend un temps de calcul équivalent à celui de* quant_initialisation_categorie, *soit* $O(k^2\sqrt{n})$.

5.8 Discussion et perspectives futures

Comme nous l'avons vu dans ce chapitre, certains algorithmes d'apprentissage non-supervisé peuvent être accélérés au-delà de ce qui est classiquement possible en quantisant certaines de leurs sous-routines. De plus, il est aussi possible d'économiser sur le coût de communication dans certaines situations si on permet xau participants d'échanger de l'information quantique, comme celle de la version distribuée de l'algorithme de k-médianes (section 5.6.2). Cette section résume les différents résultats présentés dans ce chapitre et ouvre des perspectives d'algorithmes quantiques qui vont au-delà de l'algorithme de Grover et ses variantes.

5.8.1 Comparaison équitable entre algorithmes d'apprentissage quantique et classique et bornes inférieures

Afin de faire une comparaison équitable entre un algorithme de catégorisation classique et sa contrepartie quantique, il est important de toujours considérer aussi bien le meilleur algorithme de catégorisation classique fonctionnant sur les distances entre paire de points que l'avantage qui peut-être obtenu si la description complète des points de données est disponible. Par exemple, dans le cas de la construction d'un graphe de voisinage, comme vu dans section 5.7.1, l'algorithme classique des kd-trees permet de calculer ce graphe de manière si efficace que quantiser l'algorithme classique travaillant sur les distances n'offre pas d'avantage significatif si d correspond à une faible dimension [129]. Une question fondamentale est d'étudier les bornes inférieures qui peuvent être atteintes pour différents scénarios de catégorisation, que ce soit classiquement ou

quantiquement. En particulier, est-il possible d'offrir une caractérisation des situations où la version quantisée pourrait offrir une réelle amélioration sur la version classique ? Par exemple, dans le cas de l'arbre couvrant minimal (section 5.4), Dürr, Heiligman, Høyer et Mhalla ont prouvé que leur algorithme est optimal [77]. Ainsi, si on pouvait aussi réduire la k-catégorisation d'espacement maximum à la construction de l'arbre couvrant minimal, il s'en suivrait que n'importe quel algorithme de catégorisation qui cherche à maximiser le critère de k-catégorisation d'espacement maximum, qu'il soit quantique ou classique, ne pourrait pas faire mieux que $\Omega(n^{3/2})$. Pour l'algorithme des k-médianes, il est envisageable qu'on puisse calculer la médiane en un temps linéaire, ce qui réduirait le temps d'exécution d'autant et aurait aussi un impact conséquent sur la complexité de la communication.

Problème de catégorisation	Classique	Quantique
Arbre couvrant minimal / k-catégorisation d'espacement maximum	$\Theta\left(n^2\right)$	$\Theta\left(n^{3/2}\right)$
Catégorisation divisive	$\Theta\left(n^2\right)$,	$\Omega\left(n\right)$, $O\left(n \log n\right)$
k-médianes (standard)	$\Omega\left(\frac{n^2}{k}\right)$, $O\left(t\frac{n^2}{k}\right)$	$\Omega\left(n\right)$, $O\left(t\frac{1}{\sqrt{k}}n^{3/2}\right)$
k-médianes (distribué), coût de communication	$\Theta\left(dn\right)$	$\Omega\left(d\sqrt{n}\right)$, $O\left(t\sqrt{kn}(d+\log(nd_{max}))\right)$
Construction d'un graphe de voisinage (pour d une dimension moyenne ou élevée)	$\Theta\left(n^2\right)$	$\Omega\left(n\right)$, $O\left(\sqrt{k}n^{3/2}\right)$
Détection d'anomalies (se basant sur le voisinage)	$\Theta\left(n^2\right)$	$\Omega\left(n\right)$, $O\left(\sqrt{k}n^{3/2}\right)$
Initialisation "intelligente" des centres des catégories	$\Omega\left(n\right)$, $O\left(k^2n\right)$	$\Omega(\sqrt{n})$, $O\left(k^2\sqrt{n}\right)$

TAB. 5.2 – Tableau résumant les bornes inférieures et supérieures en nombre de requêtes connues pour les algorithmes d'apprentissage non-supervisé présentés dans ce chapitre. Par souci de clarté et d'espace, le temps de construction de la boîte noire (section 5.3.3) qui est de $O(nd + n \log n)$ n'est pas comptabilisé dans le tableau, ni le facteur logarithmique en n et d que coûte chaque appel à cette même boîte noire.

Le tableau 5.2 résume les résultats présentés dans ce chapitre. Ces bornes sont relativement serrées pour la catégorisation par arbre couvrant minimal, pour la catégorisation divisive ainsi que pour l'initialisation "intelligente" du centre des catégories. Pour la

construction d'un graphe de voisinage et la détection d'anomalies, il semble probable que la vraie borne inférieure soit plus proche de $\Omega(n^{3/2})$ que de $\Omega(n)$ du fait que ces algorithmes semblent résoudre n instances de problèmes de type "Grover" (chaque instance demandant un temps de $O(n)$), ce qui aurait pour conséquence que les algorithmes quantiques présentés dans ce chapitre soient optimaux. Par contre pour la version quantique de l'algorithme des k-médianes, la porte est encore grande ouverte pour essayer de trouver un algorithme qui se rapproche de la borne inférieure.

Une autre question fondamentale est de déterminer quels sont les algorithmes qui admettent une version distribuée pour laquelle transmettre de l'information quantique permet d'économiser sur le coût de communication comparativement à la version classique. Dans ce chapitre, il semble que seul l'algorithme des k-médianes et celui pour l'initialisation "intelligente" des centres des catégories[29] admettent une version distribuée intéressante ayant un coût de communication dans $O(\sqrt{n})$ qubits. Au final, ces protocoles produisent en sortie seulement $O(k)$ bits d'information, qui correspondent à la description des centres des catégories, contre $O(n)$ bits par exemple pour un algorithme tel que celui de l'arbre couvrant minimal. En soit, cette observation n'est pas vraiment surprenante car s'il était possible de calculer de manière distribuée la sortie d'une fonction qui est de taille $O(n)$ bits avec moins de $\Omega(n)$ qubits de communication, cela contredirait le théorème énoncé dans [59].

5.8.2 Quantisation d'Isomap

Isomap [179] est un algorithme de réduction de dimensionnalité qui permet d'apprendre une représentation de faible dimension des données pour des variétés non-linéaires (voir la section 3.3.2 pour plus de détails). Isomap présuppose que les données observées qui sont en haute dimension ont été générées par une courbure de l'espace de faible dimension. L'idée principale de l'algorithme est d'approximer la distance géodésique entre deux points sur cette courbure de l'espace par la longueur du plus court chemin entre ces mêmes deux points sur un graphe de voisinage. Une fois que la distance géodésique a été estimée pour chacune des paires de points de l'ensemble de données, l'échelonnement multidimensionnel [62] est utilisé sur cette matrice de distance afin de générer une représentation

[29] Pour rendre l'algorithme d'initialisation des centres des catégories distribué, il suffit de rendre distribué la version de l'itération de Grover utilisée dans quant_trouver_max.

de faible dimension des données. Le goulet d'étranglement calculatoire d'Isomap, ainsi que de la plupart des algorithmes de réduction de dimensionnalité, réside dans le fait de devoir calculer les vecteurs propres et valeurs propres d'une matrice n par n, ce qui requiert un temps de $O(n^3)$. Le coût de ce calcul de vecteurs propres et valeurs propres domine en général la complexité en temps de calcul des algorithmes, ce qui conduit à formuler la question ouverte suivante.

Question ouverte 5.6 (Algorithme quantique pour le calcul de vecteurs propres et valeurs propres). *Est-il possible d'avoir un algorithme quantique qui permette de déterminer les vecteurs propres* et *valeurs propres d'une matrice de taille n par n plus rapidement que cela n'est possible classiquement ?*

En ce qui concerne la détermination des vecteurs propres d'une matrice, au meilleur de ma connaissance on ne connaît pas pour l'instant d'algorithmes quantiques qui sont plus efficaces que leurs contreparties classiques. Il est néanmoins possible, en se basant sur les algorithmes et outils développés dans ce chapitre, de quantiser certains algorithmes de réduction de dimensionnalité. Ainsi, il existe une version d'Isomap nommée L-Isomap [174] (pour *landmark Isomap* en anglais) qui travaille sur un sous-ensemble de points de l'ensemble de données intelligemment choisis, appelés *landmarks* en anglais, pour calculer les vecteurs propres et valeurs propres nécessaires à la réduction de dimensionnalité. L'algorithme quant_l – isomap est une version quantisée de cet algorithme.

Algorithme 11 quant_l – isomap(D_n,k,l)

[Étape 1] Construire un graphe de voisinage en utilisant quant_construction_graphe_voisinage(D_n,k)
[Étape 2] Choisir les l landmarks en utilisant la version classique de l'algorithme pour initialiser "intelligemment" les centres initiaux des catégories
[Étape 3] Calculer la longueur des plus courts chemins entre les l landmarks et les n points de données en utilisant l'algorithme de Dijsktra (classique) implémenté avec des monceaux de Fibonacci
[Étape 4] Trouver les valeurs propres et vecteurs propres de la matrice de distance générée à l'étape 3 en utilisant une version de l'échelonnement multidimensionnel adaptée aux landmarks [173]

Théorème 5.11 (Convergence de l'algorithme quant_l – isomap). *Avec probabilité élevée, l'algorithme* quant_l – isomap *permet de trouver une représentation en faible dimension*

d'un ensemble de données D_n en un temps de $O(\sqrt{k}n^{3/2} + kln\log n + l^2n)$, où n est le nombre de points de l'ensemble, l le nombre de landmarks choisis pour le représenter et k la taille du voisinage considéré lors de la construction du graphe de voisinage.

Démonstration. La construction du graphe de voisinage lors de l'étape 1 avec l'algorithme quant_construction_graphe_voisinage prend un temps de $O(\sqrt{k}n^{3/2})$. Le reste de l'analyse concerne des algorithmes classiques, ainsi choisir les l landmarks avec la version classique de l'algorithme d'initialisation requiert un temps de $O(l^2n)$. De plus, si l'algorithme de Dijsktra [75] est implémenté à l'aide de monceaux de Fibonacci, il prend un temps de $O(kln\log n)$. La version de l'échelonnement multidimensionnel basée sur les landmarks [173] prend elle un temps de $O(l^2n)$. Au final, l'algorithme quant_l − isomap a donc une complexité globale de $O(\sqrt{k}n^{3/2} + kln\log n + l^2n)$ □

Classiquement, le même algorithme a un temps d'exécution de $O(n^2 + kln\log n + l^2n)$ qui est surtout dominé par le coût de construire le graphe de voisinage. La version quantique offre donc un gain sur le classique qui est de $O(\sqrt{\frac{n}{k}})$ ce qui est significatif tant que $k \ll n$ mais tend vers 1 si k devient très proche de n.

5.8.3 Algorithmes d'apprentissage basés sur le comptage

Certaines extensions de l'algorithme de Grover se focalisent non pas sur la recherche mais plutôt sur le comptage du nombre de solutions possibles [47]. Entre autres, la technique d'*estimation d'amplitude* [46] peut être utilisée pour compter (exactement ou approximativement) le nombre de solutions t avec, là aussi, un gain quadratique par rapport aux algorithmes classiques. Ainsi, si les variantes de l'algorithme de Grover pour la recherche permettent de trouver les plus proches voisins d'un point ou identifier la médiane parmi un ensemble de points, les extensions permettant de compter semblent plus appropriées à des *tâches d'estimation de densité*, comme estimer le nombre de voisins qui se trouve dans le voisinage fixe d'un point ou encore la densité d'une zone de l'espace des données.

Une version de l'algorithme de comptage [46] permet de *compter exactement* t, le nombre de solutions, en un temps de $O(\sqrt{(t+1)(n-t+1)})$, où n est la taille du domaine de f. Dans le cas de la détection d'anomalies (section 5.7.2), on peut construire un algorithme quant_comptage_voisins basé sur cette sous-routine, qui considère comme étant

des anomalies tous les points situés dans des zones peu denses. Pour chaque point, cet algorithme compte le nombre de voisins jusqu'à un seuil t_{min} (servant à déterminer si un point est ou non une anomalie) qui se trouvent dans le ϵ-voisinage de celui-ci, pour ϵ une distance fixe choisie de manière appropriée. Ainsi, si un point est très peu entouré, on peut en conclure qu'il est isolé et constitue très probablement une anomalie. Le temps d'exécution de la sous-routine quant_comptage_voisins peut alors se simplifier à $O(\sqrt{n})$ dès que le seuil t_{min} est choisi comme étant une constante. L'algorithme 12 formalise cette technique permettant d'identifier les anomalies en se basant sur la densité de la zone les entourant.

Algorithme 12 quant_anomalie_detection_densite(D_n,t_{min})

Pour chaque point de données x_i de D_n **faire**
Utiliser quant_comptage_voisins pour estimer si le nombre de voisins situés dans le ϵ-voisinage de x_i est plus petit que t_{min}
Si jamais x_i à un nombre de voisins plus petit que t_{min}, comptabiliser x_i comme une anomalie
fin pour
Retourner les anomalies identifiées

Théorème 5.12 (Algorithme quantique pour la détection d'anomalies basée sur la densité). *Avec probabilité élevée, l'algorithme* quant_anomalie_detection_densite *permet d'identifier tous les points qui sont des anomalies, parmi un ensemble de données de taille n en un temps de $O(n^{3/2})$.*

Démonstration. Pour chaque point de données, on peut compter si le nombre de voisins qui se trouve dans son ϵ-voisinage est plus petit que t_{min} en un temps de $O(\sqrt{n})$ en utilisant quant_comptage_voisins. Compter le nombre de voisins pour tous les n points et identifier les anomalies présentes requiert donc un temps global de $O(n^{3/2})$. □

Remarque 5.3 (Quantisation des méthodes de voisinage en apprentissage supervisé). *La sous-routine* quant_trouver_c_plus_proches_voisins *ainsi que les variantes de l'algorithme de Grover permettant de compter le nombre de solutions peuvent aussi permettre d'accélérer par un facteur quadratique les méthodes de voisinage utilisées en apprentissage supervisé telles que les k-plus proches voisins [71] ou les fenêtres de Parzen [155].*

5.8.4 Autres directions de recherche

L'algorithme de Grover [103], et sa généralisation l'amplification d'amplitude [46], sont applicables à de très nombreuses situations mais offrent un gain qui est au mieux quadratique par rapport au meilleur algorithme classique. Plus récemment, d'autres techniques algorithmiques basées sur des *versions quantiques des marches aléatoires* [4, 128] ou *des chaînes de Markov* [139, 178] ont émergé[30]. Ces techniques semblent prometteuses pour permettre de dépasser les limites intrinsèques de l'algorithme de Grover. En particulier, il existe des analogies et des liens entre certains algorithmes d'apprentissage et certaines techniques algorithmiques quantiques, qui peuvent être exploités pour développer de nouveaux algorithmes quantiques d'apprentissage. Un exemple de candidat sérieux est un algorithme de catégorisation basé sur les marches aléatoires [106].

Un changement de paradigme de calcul et de modèle de représentation, comme adopter le point de vue du modèle basé sur la mesure [122] ou du calcul adiabatique, peut aussi conduire au développement de nouveaux algorithmes d'apprentissage. Ces deux modèles offrent le même pouvoir d'expressivité que le modèle "traditionnel" des circuits [2, 51], mais amènent à adopter une perspective différente du problème et peuvent conduire à la découverte de nouveaux algorithmes qui sont plus naturels à développer dans ces modèles que dans celui des circuits. Ainsi, le modèle basé sur la mesure semble être un cadre naturel pour exprimer des algorithmes parallèles ou distribués alors que le calcul adiabatique semble plutôt approprié pour développer des algorithmes d'optimisation, tels qu'une version quantique du *recuit simulé* (appelé *simulated annealing* en anglais). Par exemple, une série de développements récents inclut un algorithme pour l'évaluation des arbres AND-OR (ou de façon équivalente NAND). Il s'agit d'un problème pour lequel on ne connaissait pas d'algorithmes quantiques efficaces jusqu'à une avancée récente dans le modèle adiabatique [84]. Cet algorithme a été ensuite "traduit" vers le modèle des circuits quantiques [55], puis amélioré et généralisé aux arbres MIN-MAX [61].

[30] Voir par exemple `http://www.iqc.ca/~mmosca/web/papers/algorithms-survey08.pdf` pour un survol récent des algorithmes quantiques.

CHAPITRE 6

APPRENTISSAGE MACHINE DANS UN MONDE QUANTIQUE

6.1 Apprendre dans un monde quantique

En général, l'apprentissage machine est implicitement considéré comme se déroulant dans un monde classique, où les points de données de l'ensemble d'entraînement décrivent des objets classiques et où la machine qui réalise l'apprentissage est un ordinateur classique (par exemple une machine de Turing ou un circuit logique classique). Dans le chapitre 5, nous avons déjà eu l'occasion de voir que les frontières entre le monde classique et quantique ne sont pas figées et qu'il est possible d'utiliser un ordinateur quantique pour accélérer certains algorithmes d'apprentissage. Dans ce chapitre, nous allons étudier la situation d'apprentissage où même l'ensemble de données est quantique, c'est-à-dire qu'il se compose de systèmes quantiques (la machine sera elle aussi quantique). Ce changement de théorie physique sous-tendant l'apprentissage aura un impact direct sur la manière dont l'apprentissage se déroule dans ce nouveau cadre et sur ses limitations.

Considérons par exemple ces deux scénarios qui correspondent tous les deux à des situations où l'on dispose d'un ensemble de données composé d'états quantiques[1].

Scénario 6.1 (Sonde spatiale ayant collecté des états quantiques). *Supposons qu'une sonde spatiale ait été envoyée dans l'espace et qu'elle ait pu ramasser des échantillons de phénomènes quantiques rencontrés à l'autre bout de la galaxie. Est-il possible de faire de l'apprentissage directement sur ces échantillons ?*

Scénario 6.2 (Physicien réalisant des expériences dans son laboratoire). *Supposons qu'un physicien réalise des expériences quantiques dans son laboratoire et qu'il ait pu collecter des données de ces expériences (sous la forme de copies d'états quantiques ou d'observations classiques obtenues suite à des mesures). Que peut-il apprendre à partir de ces données ?*

Mes objectifs principaux en reformulant certains problèmes de la théorie quantique de la détection et de l'estimation comme des tâches d'apprentissage machine sont :

[1] Le lecteur pourra décider par lui-même quel scénario semble le plus réaliste.

- de fournir, en adoptant le paradigme de l'apprentissage machine, une approche constructive permettant de résoudre certains scénarios où ces problèmes apparaissent naturellement,
- de caractériser ces tâches d'apprentissage en terme de la quantité d'information nécessaire pour les mener à bien (mesurée par exemple en nombre de copies des états quantiques),
- de développer un cadre permettant de relier et comparer ces différentes tâches.

Le plan de ce chapitre est le suivant. Tout d'abord dans la section 6.1.1, le cadre d'apprentissage qui correspond à apprendre sur un ensemble d'entraînement quantique est énoncé. Les notions de classe d'apprentissage, de réduction entre tâches d'apprentissage et de matrice de similarité entre états quantiques sont formalisées dans cette même section. Ensuite, dans la section 6.2, l'équivalent quantique de la tâche de classification est défini ainsi que ses différentes versions que sont la classification binaire, la classification binaire pondérée et la classification multiclasse. Les variantes quantiques de la catégorisation, la réduction de dimensionnalité ainsi que de l'estimation de densité sont discutées ensuite dans la section apprentissage non-supervisé quantique (section 6.3). Finalement, la section 6.4 conclut ce chapitre par une discussion et une ouverture sur d'autres perspectives de recherche.

6.1.1 Apprendre avec un ensemble d'entraînement quantique

Dans un monde quantique, un algorithme d'apprentissage a toujours besoin d'un ensemble d'entraînement à partir duquel réaliser son apprentissage, mais cet ensemble contient maintenant des *objets quantiques* au lieu d'*observations classiques sur des objets classiques.*

Définition 6.1 (Ensemble d'entraînement quantique). *Un* ensemble d'entraînement quantique *contenant n états purs, peut être décrit comme $D_n = \{(|\psi_1\rangle, y_1), \ldots, (|\psi_n\rangle, y_n)\}$, où $|\psi_i\rangle$ est le $i^{ème}$ état quantique de l'ensemble d'entraînement et y_i est la classe associée à cet état quantique.*

Exemple 6.1 (Ensemble d'entraînement composé d'états purs définis sur d qubits). *Dans la situation où tous les états purs de l'ensemble d'entraînement vivent dans un espace de Hilbert formé de d qubits et où on s'intéresse à la tâche de classification binaire (cf.*

section 6.2.1); on a $|\psi_i\rangle \in \mathbb{C}^{2^d}$ *et* $y_i \in \{-1, +1\}$.

Dans ce chapitre, nous nous restreignons au cas où les états sont quantiques mais où les classes demeurent classiques, mais une généralisation possible est de considérer que les objets puissent se trouver dans une *superposition quantique de classes*[2]. Une autre extension du modèle est de permettre aux états quantiques d'être des mélanges statistiques[3], et pas seulement des états purs.

Le modèle d'apprentissage mis en avant dans ce chapitre est *complémentaire au modèle proposé par Aaronson* [1] où l'ensemble de données est composé de POVMs, et non pas d'états quantiques. Dans ce modèle (voir section 4.9), nous recevons un nombre fini de copies d'un état quantique inconnu et le but est, en faisant un "entraînement" sur un certain nombre de POVMs avec cet état, de pouvoir généraliser à des POVMs non observés auparavant.

6.1.2 Classes d'apprentissage

Une des difficultés intrinsèques de définir l'apprentissage dans un monde quantique provient des nombreuses façons dont les états quantiques pourraient être spécifiés dans l'ensemble d'entraînement. Par exemple, l'ensemble d'entraînement pourrait contenir un nombre fini de copies de chaque état ou consister en une description classique de ces mêmes états (comme une description explicite de leur matrice densité). Cette dernière situation est la plus "puissante" dans le sens de la théorie de l'information puisqu'en principe, il est toujours possible de produire autant de copies que désiré à partir de la description classique d'un état.

Le concept de *classe d'apprentissage* qui précise la forme de l'ensemble d'entraînement, la sophistication technologique dont l'apprenant dispose et le but de l'apprentissage a été introduit dans [8].

Définition 6.2 (Classe d'apprentissage). *Soit une* classe d'apprentissage $\mathsf{L}_{but}^{contexte}$, *où l'index inférieur "but" réfère au but de l'apprentissage alors que l'index supérieur "contex-*

[2]Pour rappel, être dans une superposition quantique de classes n'est pas équivalent à la notion classique d'un point de données qui appartiendrait à plusieurs classes de manière floue ou probabiliste.

[3]Dans cette thèse, les termes "états mélangés" et "mélanges statistiques" seront utilisés de manière interchangeable.

te" se rapporte à la forme de l'ensemble d'entraînement et/ou à la technologie à laquelle l'apprenant a accès.

Parmi les valeurs possibles du *but* on trouve *cl*, qui incarne l'idée de faire de l'apprentissage avec un but classique en tête, et *qu* pour de l'apprentissage avec une motivation quantique. De façon similaire, l'index supérieur *contexte* peut prendre les valeurs *cl* pour "classique" si tout est classique (avec une possible exception pour le but), et *qu* si "quelque chose de quantique" est en train de se produire. Le *contexte* peut prendre d'autres valeurs lorsque le besoin se fait sentir d'être plus spécifique. Considérons par exemple les deux classes d'apprentissage suivantes qui s'intéressent à l'apprentissage avec un but classique à l'esprit.

Définition 6.3 (Apprentissage machine dans un monde classique). L_{cl}^{cl} *correspond à faire de l'apprentissage machine dans le sens usuel du terme, où on utilise* des moyens classiques pour apprendre à partir d'observations classiques sur des objets classiques.

Définition 6.4 (Apprentissage machine avec l'aide d'un ordinateur quantique). L_{cl}^{qu} *correspond à la classe d'apprentissage où on* peut utiliser un ordinateur quantique pour faciliter l'apprentissage mais où le but reste de réaliser une tâche d'apprentissage classique *; dans ce cas l'ordinateur quantique pourrait être utilisé pour accélérer l'apprentissage (soit le cas considéré dans le chapitre 5).*

Il est important de comprendre que ces deux classes d'apprentissage représentent la quantité d'information dont nous disposons pour réaliser l'apprentissage mais ne prétendent aucunement caractériser le temps de calcul nécessaire pour réaliser cet apprentissage. Ainsi, il se pourrait que les classes L_{cl}^{cl} et L_{cl}^{qu} soient équivalentes (comme semble le suggérer le théorème 4.1 de la section 4.2) sans que cela ne contredise le fait que l'ordinateur quantique puisse permettre d'accélérer significativement l'apprentissage.

Dans ce chapitre, nous nous concentrons sur le cas spécifique où "$but = qu$".

Définition 6.5 (Apprentissage quantique à partir des descriptions classiques des états quantiques). L_{qu}^{cl} *est définie comme la* classe d'apprentissage dans laquelle la description des états quantiques de l'ensemble d'entraînement est donnée classiquement, *c'est-à-dire* $D_n = \{(\psi_1, y_1), \ldots, (\psi_n, y_n)\}$, *où* ψ_i *est la description classique de l'état quantique* $|\psi_i\rangle$.

L'apprentissage devient plus ardu[4] quand l'ensemble de données est disponible uniquement sous forme quantique, auquel cas plus on dispose de copies d'un état, plus on peut potentiellement extraire d'information sur cet état. Ainsi, un corollaire de la borne de Holevo (corollaire 2.1, section 2.3.2) précise qu'il est impossible d'extraire plus de d bits classiques d'information d'un système quantique vivant dans un espace de Hilbert formé par d qubits. De plus, le théorème de non-clonage [115] (théorème 2.2, section 2.3.2) interdit de produire deux copies identiques d'un état quantique inconnu à partir d'une seule copie de cet état. Enfin, le théorème 2.4 [25] (section 2.3.2) énonce qu'il est impossible d'acquérir de l'information quantique sur un état sans avoir en retour une probabilité non-négligeable de le perturber.

Définition 6.6 (Apprentissage quantique à partir d'un nombre fini de copies des états quantiques). $\mathsf{L}_{qu}^{\otimes s}$ *est la classe d'apprentissage* dans laquelle on dispose d'un nombre fini d'au moins s copies de chacun des états quantiques de l'ensemble d'entraînement, *c'est-à-dire* $D_n = \{(|\psi_1\rangle^{\otimes s}, y_1), \ldots, (|\psi_n\rangle^{\otimes s}, y_n)\}$ *; où* $|\psi_i\rangle^{\otimes s}$ *symbolise* s *copies de l'état* $|\psi_i\rangle$.

On peut contraster cette classe avec les classes d'apprentissage du monde classique (comme L_{cl}^{cl}), où recevoir des copies additionnelles d'un objet particulier est inutile puisque celles-ci ne révèlent pas de nouvelle information sur l'objet. La raison principale qui nous a conduit à définir des classes d'apprentissage est pour y placer des ensembles de données quantique ainsi que des tâches d'apprentissage.

Définition 6.7 (Appartenance d'un ensemble d'entraînement à une classe d'apprentissage). *Un ensemble de données* D_n appartient à une classe d'apprentissage L, *si la description de cet ensemble de données respecte la définition de la classe* L. *Par souci de simplification de la notation, on dénotera cette relation d'appartenance par* $D_n \in \mathsf{L}$.

Définition 6.8 (Appartenance d'une tâche d'apprentissage à une classe d'apprentissage). *Une tâche d'apprentissage* A appartient à une classe d'apprentissage L, *si étant donné un*

[4]Il faut remarquer cependant que la description classique d'un état est en générale exponentiellement plus longue à écrire si on la représente classiquement sous forme de bits que l'état quantique correspondant sous forme de qubits. Ainsi, on peut imaginer une situation paradoxale où pour décrire classiquement les 2^{1000} amplitudes d'un état quantique vivant sur 1000 qubits, il faudrait plus de mémoire qu'il n'y a d'atomes dans l'univers, même si chaque atome pouvait être utilisé individuellement comme unité classique de mémoire (c'est-à-dire un bit). Par contraste, si nous pouvions manipuler les atomes de manière cohérente et de les maintenir en superposition, il suffirait de 1000 atomes pour stocker le même état.

ensemble de données D_n qui appartient à la classe L (et donc qui respecte sa définition), il est possible de résoudre cette tâche d'apprentissage A pour cet ensemble de données D_n. Par souci de simplification de la notation, on dénotera cette relation d'appartenance par $A \in \mathsf{L}$.

Les classes d'apprentissage quantiques forment une *hiérarchie au sens de la théorie de l'information*, où plus une classe est haute dans la hiérarchie, plus elle contient d'information permettant de réaliser des tâches reliées aux ensembles de données appartenant à cette classe. La classe L_{qu}^{cl} est au sommet de la hiérarchie car il s'agit de la classe d'apprentissage correspondant à *avoir une connaissance complète des états quantiques* composant l'ensemble de données. Soit $\equiv_l$, $\leq_l$ et $<_l$ les opérateurs dénotant respectivement les relations d'*équivalence*, de plus *faible ou égal* et de *strictement plus faible* à l'intérieur de la hiérarchie. Les propositions suivantes décrivent des relations entre différentes classes d'apprentissage composant la hiérarchie.

Proposition 6.1. $\mathsf{L}_{qu}^{\otimes s} \equiv_l \mathsf{L}_{qu}^{cl}$ *lorsque* $s \longrightarrow \infty$.

Démonstration. Lorsque le nombre de copies tend vers l'infini, il est toujours possible d'estimer un état quantique $|\psi\rangle$ en utilisant la tomographie quantique et en reconstruisant sa description classique avec une précision arbitraire (voir section 6.3.3 pour plus détails). □

Proposition 6.2. $\mathsf{L}_{qu}^{\otimes 1} \leq_l \ldots \leq_l \mathsf{L}_{qu}^{\otimes s} \leq_l \mathsf{L}_{qu}^{\otimes s+1} \leq_l \ldots \leq_l \mathsf{L}_{qu}^{cl}$

Démonstration. Chaque nouvelle copie d'un état donne potentiellement plus d'information sur cet état. Ainsi pour n'importe quel entier positif s, on a $\mathsf{L}_{qu}^{\otimes s} \leq_l \mathsf{L}_{qu}^{\otimes s+1}$, ce qui implique que si une tâche d'apprentissage $A \in \mathsf{L}_{qu}^{\otimes s}$, elle appartient aussi à $\mathsf{L}_{qu}^{\otimes s+1}$. De plus, de par la proposition 6.1, une description classique de l'état est au moins aussi puissante que n'importe quel nombre fini de copies. □

Proposition 6.3. $\mathsf{L}_{qu}^{\otimes s} + \mathsf{L}_{qu}^{\otimes 1} \leq_l \mathsf{L}_{qu}^{\otimes s+1}$, *où "+" dénote la restriction que les s premières copies soient mesurées séparément de la dernière copie.*

Démonstration. En réalisant une *mesure conjointe* (appelée parfois *mesure cohérente*), faisant interagir ensemble $s+1$ copies, il est possible que nous apprenions plus d'information qu'en réalisant simplement une mesure conjointe sur s copies, plus une mesure

séparée sur une autre copie. (Voir [157] pour un exemple spécifique où $s = 1$ et [56] pour des résultats pour s arbitraire.) □

Une question ouverte importante est de savoir si cette hiérarchie est stricte ou non. Elle peut être formulée de la manière suivante.

Question ouverte 6.1 (Hiérarchie stricte des classes d'apprentissage). *Dans l'expression* $\mathsf{L}_{qu}^{\otimes 1} \leq_l \ldots \leq_l \mathsf{L}_{qu}^{\otimes s} \leq_l \mathsf{L}_{qu}^{\otimes s+1} \leq_l \ldots \leq_l \mathsf{L}_{qu}^{cl}$, *est-il possible de remplacer tous les* $\leq_l$ *par* $<_l$ *?*

Il y a de bonnes raisons de croire que la réponse à cette question est positive, car généralement plus nous disposons de copies d'un état quantique, plus il est possible d'apprendre de l'information sur cet état. De plus, il a été prouvé que dans certaines situations des mesures conjointes sont plus informatives que des mesures individuelles [56, 157]. Cependant, cela ne signifie pas que cette information puisse être utilisée de manière constructive pour résoudre des tâches d'apprentissage.

6.1.3 Réductions entre tâches d'apprentissage

La notion de *réduction entre tâches d'apprentissage* [32] a été développée et formalisée durant ces dernières années dans le contexte de l'apprentissage machine classique par Langford[5] et ses co-auteurs.

Définition 6.9 (Réduction d'apprentissage [32]). *Une* tâche d'apprentissage A se réduit à une autre tâche d'apprentissage B *(figure 6.1), si en ayant accès à une boîte noire (un oracle) qui permet de résoudre* B*, il est aussi possible de résoudre* A*.*

Une réduction d'apprentissage peut être vue comme un *énoncé du type théorie de l'information*, affirmant à quel point il est facile de résoudre une tâche d'apprentissage particulière étant donné un algorithme (modélisé abstraitement par un oracle) qui résout une autre tâche. Bien qu'il soit en général souhaitable que cette transformation soit efficace, les réductions d'apprentissage diffèrent des réductions "traditionnelles" utilisées en complexité du calcul (tel que les réductions de type Karp ou Turing fonctionnant en

[5] Voir par exemple l'adresse suivante `http://hunch.net/~jl/projects/reductions/reductions.html` pour le projet de Langford sur les réductions d'apprentissage.

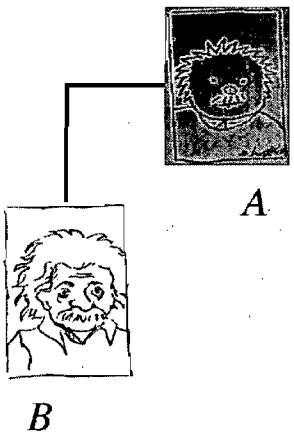

FIG. 6.1 – Illustration du principe de réduction d'apprentissage. Une tâche d'apprentissage A se réduit à une tâche d'apprentissage B si en ayant accès à un oracle pour résoudre la tâche B, il est aussi possible de résoudre la tâche A.
(Tiré des notes d'un tutoriel sur les réductions d'apprentissage par John Langford. Source :`http://hunch.net/~jl/projects/reductions/tutorial/helsinki.ps`.)

temps polynomial). En effet, les réductions d'apprentissage ne cherchent pas à caractériser le temps de calcul nécessaire pour résoudre une tâche particulière. Leur but est plutôt d'offrir une façon de comparer et de relier différentes tâches d'apprentissage dans le sens de la théorie de l'information. Ainsi, si A se réduit à B, cela signifie que si du progrès est réalisé sur la manière de résoudre la tâche B, cela peut être directement transféré pour aider à résoudre A en utilisant la réduction. De plus, si différentes tâches peuvent toutes être réduites à la même *primitive d'apprentissage*, tout nouvel avancement pour résoudre plus efficacement cette primitive aura un impact direct sur toutes les autres tâches d'apprentissage. Par exemple dans les sections 6.2.2 et 6.2.3, nous verrons qu'il est possible de résoudre la version binaire pondérée et la version multiclasse de la classification étant donné un oracle permettant de résoudre la classification binaire standard.

Une bonne réduction offre souvent une garantie en terme de comment la performance de la boîte noire pour résoudre la tâche B implique aussi une bonne performance pour résoudre la tâche A. Ainsi, en classification, cette garantie peut être une borne sur l'erreur réalisée par le classifieur final. Ces bornes relient généralement l'erreur moyenne des

classifieurs générés en appelant l'oracle sur les sous-problèmes B, à l'erreur globale que le classifieur composé réalisera sur le problème général A.

Définition 6.10 (Erreur). *L'*erreur ϵ *(ou* taux d'erreur*) d'un classifieur f est définie comme la probabilité que ce classifieur prédise la mauvaise classe y_i d'un état quantique $|\psi_i\rangle$ choisi aléatoirement parmi les états de l'ensemble de données quantique D_n. Soit formellement :*

$$\epsilon_f = \frac{1}{n}\sum_{i=1}^{n} prob(f(|\psi_i\rangle) \neq y_i) \tag{6.1}$$

Remarque 6.1 (Erreur d'entraînement). *Pour être précis, la définition 6.10 caractérise l'erreur d'entraînement du classifieur mais pas son erreur de généralisation (cf. section 3.2.2). Pour l'instant, nous nous focalisons sur la minimisation de cette erreur d'entraînement mais nous allons revenir sur l'erreur de la généralisation dans la discussion de la section 6.4.2.*

Dans le contexte de l'apprentissage supervisé quantique, en plus du taux d'erreur la notion de *regret* [133] prend une importance particulière.

Définition 6.11 (Regret [133]). *Le* regret r *d'un classifieur f est défini comme la différence entre son taux d'erreur ϵ_f et la plus petite erreur ϵ_{opt} qu'il est possible d'atteindre sur le même problème. Soit formellement :*

$$r_f = \epsilon_f - \epsilon_{opt} \tag{6.2}$$

Le regret d'un classifieur, ainsi que son erreur, peuvent potentiellement prendre n'importe quelle valeur dans l'intervalle entre zéro et un. Le concept de regret est particulièrement pertinent pour les problèmes d'apprentissage dits difficiles, où le taux d'erreur brut ne constitue pas une mesure appropriée pour caractériser la difficulté inhérente de l'apprentissage. Ainsi dans certaines situations d'apprentissage, on peut observer un taux d'erreur élevé mais avoir un regret faible, voir nul. Dans le contexte classique, un taux d'erreur élevé mais un regret faible est généralement la conséquence d'un haut niveau de bruit. La situation est différente dans le monde quantique où un taux d'erreur élevé pourrait être dû à la difficulté physique intrinsèque de distinguer deux classes, mais n'implique pas forcément un haut niveau de bruit. Quelque soit le contexte, si le regret d'un classifieur est zéro cela signifie essentiellement que ce classifieur est optimal.

Quantiquement, une réduction ou une tâche d'apprentissage peut aussi avoir un *coût* qui lui est associée. En effet, chaque appel à l'oracle peut requérir de sacrifier un certain nombre de copies des états quantiques à cause des mesures réalisées par l'oracle durant l'apprentissage. En apprentissage supervisé, ce coût se partage entre le nombre de copies nécessaires durant la *phase d'entraînement/d'apprentissage*, où on construit un POVM f qui jouera le rôle de classifieur, et le nombre de copies dont on a besoin au *moment de faire la classification* (ou *phase de test*) où on utilise f pour classifier un état quantique inconnu $|\psi_?\rangle$.

Définition 6.12 (Coût d'entraînement/d'apprentissage). *Le* coût d'entraînement/d'apprentissage *d'une réduction est proportionnel au nombre d'appels à l'oracle faits par la réduction, multiplié par le nombre de copies de chacun des états quantiques qui sont utilisées à chaque appel. Dans le cas d'une tâche d'apprentissage, le coût est caractérisé directement par le nombre de copies des états quantiques nécessaires pour mener à bien cette tâche.*

Si l'ensemble de données $D_n \in \mathsf{L}_{qu}^{cl}$, alors mener à bien l'apprentissage ne coûte rien en terme d'information car nous disposons déjà d'une connaissance complète des états quantiques.

Définition 6.13 (Coût de classification). *Le* coût de classification *correspond au nombre de copies d'un état inconnu $|\psi_?\rangle$ qui seront utilisés par le classifieur pour prédire la classe $y_?$ de cet état.*

Dans le cas de l'apprentissage non-supervisé quantique, seul le coût d'entraînement/apprentissage est présent car en général il n'y a pas d'équivalent non-supervisé à la phase de test (à l'exception de la tâche d'estimation de densité). Ainsi par exemple en catégorisation, une fois que les catégories ont été découvertes, il n'y généralement pas de seconde étape où de nouveaux états quantiques nous serons donnés[6].

6.1.4 Fidélité, Control-Swap test et matrice de similarité

La *fidélité* est une notion fondamentale en informatique quantique, qui dans le cas de deux états purs est l'analogue quantique (du carré) du produit interne.

[6] À moins que nous soyons dans une situation d'apprentissage en ligne où nous recevons des nouvelles données quantiques périodiquement (voir [91] par exemple).

Définition 6.14 (Fidélité entre deux états purs). *La* fidélité *est une* mesure de similarité entre deux états quantiques, *qui se définit dans le cas de deux états purs* $|\psi\rangle$ *et* $|\phi\rangle$ *par* $Fid(|\psi\rangle, |\phi\rangle) = |\langle\phi|\psi\rangle|^2$. *La fidélité varie entre 0 si les états sont orthogonaux (c'est-à-dire parfaitement distingables) à 1 si les états sont identiques.*

La fidélité est similaire à une mesure communément utilisée en recherche d'information classique, appelée *similarité par le cosinus*[7]. Les propriétés de la fidélité [152, §9.2.2] incluent la *symétrie*, c'est-à-dire $Fid(|\psi\rangle, |\phi\rangle) = Fid(|\phi\rangle, |\psi\rangle)$, ainsi que l'*invariance sous transformation unitaire*, ce qui signifie que si on applique la même transformation unitaire U à deux états quantiques cela ne change pas leur fidélité : $Fid(|U\psi\rangle, |U\phi\rangle) = Fid(|\phi\rangle, |\psi\rangle)$.

Dans sa forme standard, la fidélité ne correspond pas à une métrique car elle n'obéit pas à l'inégalité du triangle[8], mais il est possible de l'adapter pour que ce soit le cas en définissant $Dist(|\psi\rangle, |\phi\rangle) = \arccos Fid(|\psi\rangle, |\phi\rangle)$ au lieu de simplement $Fid(|\psi\rangle, |\phi\rangle)$. Dans ce cas-là, la valeur de $Dist(|\psi\rangle, |\phi\rangle)$ ira de 0 si les états sont identiques à $\frac{\pi}{2}$ s'ils sont orthogonaux. Cette *distance* respecte maintenant l'inégalité du triangle et ainsi $Dist(|\psi\rangle, |\phi\rangle) \leq Dist(|\psi\rangle, |\varphi\rangle) + Dist(|\varphi\rangle, |\phi\rangle)$.

Le Control-Swap test (C-Swap test) [17, 53] est une opération quantique qui permet d'estimer la fidélité entre deux états quantiques inconnus $|\psi\rangle$ et $|\phi\rangle$. La figure 6.2 illustre le circuit du C-Swap test.

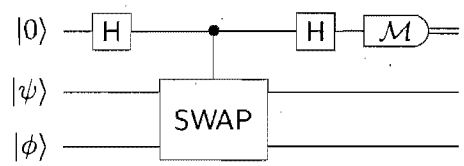

FIG. 6.2 – Circuit réalisant le Control-SWAP test.

Lemme 6.1 (Estimateur de fidélité). *Soit deux états quantiques inconnus* $|\psi\rangle$ *et* $|\phi\rangle$ *dont on dispose de e copies pour chacun de ces états. Le C-Swap test peut être utilisé e fois*

[7] La formule usuelle de la similarité par le cosinus est $cos\_sim(a, b) = \frac{\sum_{i=1}^{d} a_i b_i}{\|a\| \cdot \|b\|}$, où a et b sont tous les deux des vecteurs d'observations de taille d.

[8] Ainsi, par exemple si $|\psi\rangle = \frac{1}{\sqrt{4}}|0\rangle + \frac{\sqrt{3}}{\sqrt{4}}|1\rangle$, alors $Fid(|0\rangle, |\psi\rangle) = \frac{1}{4}$ et $Fid(|0\rangle, |0\rangle) = 1$ donc $Fid(|0\rangle, |0\rangle) > Fid(|0\rangle, |\psi\rangle) + Fid(|\psi\rangle, |0\rangle)$ ce qui montre clairement que l'inégalité du triangle n'est pas respectée.

pour estimer la fidélité $Fid(|\psi\rangle, |\phi\rangle)$.

Démonstration. L'entrée du circuit est $|0\rangle|\psi\rangle|\phi\rangle$. Après avoir appliqué la première porte de Walsh–Hadamard H, l'état a évolué vers la superposition $\frac{1}{\sqrt{2}}|0\rangle|\psi\rangle|\phi\rangle + \frac{1}{\sqrt{2}}|1\rangle|\psi\rangle|\phi\rangle$. L'application de la porte C-Swap échange les états $|\psi\rangle$ et $|\phi\rangle$ si et seulement si l'état sur le fil de contrôle est $|1\rangle$. Ainsi, l'état évolue vers $\frac{1}{\sqrt{2}}|0\rangle|\psi\rangle|\phi\rangle + \frac{1}{\sqrt{2}}|1\rangle|\phi\rangle|\psi\rangle$. Ensuite, la deuxième application de la porte de Walsh–Hadamard H amène l'état vers $\frac{1}{2}|0\rangle(|\psi\rangle|\phi\rangle + |\phi\rangle|\psi\rangle) + \frac{1}{2}|1\rangle(|\psi\rangle|\phi\rangle - |\phi\rangle|\psi\rangle)$. Finalement, la mesure du qubit supérieur donne comme résultat 0 avec probabilité 1 si $|\psi\rangle$ et $|\phi\rangle$ sont identiques. De manière générale, le résultat de la mesure sera 1 avec une probabilité de $\frac{1}{2} - \frac{1}{2}|\langle\phi|\psi\rangle|^2$. Ainsi, le C-Swap test peut être vu comme un *estimateur de la fidélité* entre les états $|\psi\rangle$ et $|\phi\rangle$. Si on dispose de e copies de chacun de ces états, $|\psi\rangle$ et $|\phi\rangle$, il est possible de l'utiliser e fois et d'obtenir un estimé $Fid(|\psi\rangle, |\phi\rangle)$ étant $1 - 2 \times \#|1\rangle/e$ ($\#|1\rangle$ représente le nombre de fois où le résultat $|1\rangle$ a été observé). Un effet additionnel du C-Swap test est de perturber de manière irréversible les états d'entrée, à moins que ceux-ci ne soient effectivement identiques. □

Définition 6.15 (Matrice de similarité d'un ensemble de données quantique)**.** *Une* matrice de similarité[9] S_n *d'un ensemble de données quantique contenant* n *états est une matrice de taille* n *par* n *où chaque entrée* $S(i,j)$ *de la matrice (pour* $i, j \in \{1, \ldots, n\}$*) contient un estimé de la fidélité entre l'état* $|\psi_i\rangle$ *et l'état* $|\psi_j\rangle$*.*

De par la propriété de symétrie de la fidélité, la matrice de similarité est une *matrice symétrique*. Il existe un algorithme efficace pour calculer cette matrice, qui requiert seulement un nombre de copies de chaque état qui est linéaire[10] en n, le nombre d'états composant l'ensemble de données quantique. L'algorithme 13 formalise la méthode permettant de calculer la matrice de similarité d'un ensemble de données quantique D_n.

Théorème 6.1 (Calcul de la matrice de similarité)**.** *L'algorithme* calcul_matrice_similarite *permet de calculer la matrice de similarité d'un ensemble*

[9] La matrice de similarité est souvent appelée *matrice de Gram* dans la littérature, particulièrement en apprentissage machine.

[10] Dans ce chapitre, nous allons mesurer le coût d'apprentissage d'une tâche (ou d'une réduction) par rapport au nombre de copies requis *individuellement* pour chacun des états. Une autre manière de faire aurait été de comptabiliser le nombre de copies requis *globalement* par rapport à la taille de l'ensemble de données. Ainsi par exemple pour le calcul de la matrice de similarité, ce coût global aurait été quadratique en n, et non pas linéaire.

Algorithme 13 calcul_matrice_similarite($D_n \in \mathsf{L}_{qu}^{\otimes\Theta(en)}$)

Pour $i = 1$ à n **faire**
  $S(i,i) = 1$
fin pour
Pour $i < j$ **faire**
  Estimer la fidélité les deux états $|\psi_i\rangle$ et $|\psi_j\rangle$ en utilisant le C-Swap test e fois,
  Mettre à jour $S(i,j) = S(j,i) = Fid(|\psi_i\rangle, |\psi_j\rangle)$
fin pour
Retourner S_n la matrice de similarité calculée

de données quantiques D_n avec une précision ϵ, pour $\epsilon = \frac{1}{e}$, étant donné $\Theta(en)$ copies de chaque état.

Démonstration. Pour chaque paire d'états $(|\psi_i\rangle, |\psi_j\rangle)$ de l'ensemble de données D_n, le C-Swap test permet d'en estimer la fidélité avec précision ϵ, où $\epsilon = \frac{1}{e}$ pour e le nombre de copies utilisées lors du test. Comme la matrice S_n est symétrique, le nombre d'entrées à estimer est de $\Theta(\frac{n(n-1)}{2}) = \Theta(n^2)$. Pour chaque état $|\psi\rangle$, on aura donc besoin d'un nombre de copies de l'ordre de $\Theta(e)$ pour chacun des n C-Swap tests où cet état apparaît, soit $\Theta(en)$ au total. □

Comme nous allons le voir dans les sections suivantes, la matrice de similarité est une construction importante qui contient l'information nécessaire pour réaliser plusieurs tâches aussi bien en apprentissage supervisé que non-supervisé.

6.2 Classification quantique

Scénario 6.3 (Classification quantique). *Supposons que nous recevions un état quantique inconnu qui a été pioché parmi un ensemble d'états purs possibles, où chaque état est étiqueté d'après la classe d'où il est originaire.*
***Interrogation** : pouvons-nous réussir à prédire la classe de cet état inconnu avec une bonne précision ?*

Cette question très générale est souvent référencée dans la littérature sous le terme *discrimination d'états*[11] (quantique) [28] et est étudiée depuis au moins aussi longtemps

[11] Cette tâche est aussi parfois appelée *distinction d'états* ou encore *identification d'état.*

que les travaux de Helstrom [109] dans les années 70 en *théorie quantique de la détection et de l'estimation*. La réponse à cette question dépendra de paramètres tels que :

- la structure et notre connaissance de cet ensemble d'états quantiques,
- la dimension de l'espace de Hilbert dans lequel vivent ces états et
- le nombre de copies de l'état inconnu dont on dispose.

Cette section adopte un point de vue du type apprentissage machine en reformulant le problème de la discrimination d'état quantique en une tâche d'apprentissage baptisée *classification quantique*. Dans les prochaines sections, trois analogues quantiques de la classification seront définies : la *classification binaire*, la *classification binaire pondérée* et la *classification multiclasse*.

6.2.1 Classification binaire

La tâche de *classification binaire* consiste à prédire la classe $y_? \in \{-1, +1\}$ d'un état quantique inconnu $|\psi\rangle_?$ à partir d'*une copie unique de cet état*[12]. Formellement, cette tâche d'apprentissage peut se définir de la manière suivante.

Tâche d'apprentissage quantique 6.1 (Classification binaire).
Entrée *: $D_n = \{(|\psi_1\rangle, y_1), \ldots, (|\psi_n\rangle, y_n)\}$, un ensemble d'entraînement quantique, où $|\psi_i\rangle \in \mathbb{C}^{2^d}$ et $y_i \in \{-1, +1\}$.*
Sortie *: un POVM jouant le rôle d'un classifieur binaire f qui peut prédire la classe $y_?$ d'un état quantique inconnu $|\psi_?\rangle$ étant donné une copie de cet état.*
But *: construire un classifieur binaire f qui minimise l'erreur d'entraînement $\epsilon_f = \frac{1}{n}\sum_{i=1}^{n} prob(f(|\psi_i\rangle) \neq y_i)$.*

Une question naturelle à se poser est quelle est la meilleure probabilité de succès qu'on peut espérer obtenir, ou encore de manière équivalente quel est le plus petit taux d'erreur possible. La situation la plus simple à analyser est lorsque nous avons *connaissance classique complète sur les états de l'ensemble d'entraînement* (c'est-à-dire que $D_n \in \mathsf{L}_{qu}^{cl}$). Cependant même dans ce cas-là, il n'est généralement pas possible de construire un processus qui classifie toujours correctement n'importe quel état inconnu à partir d'une copie unique de cet état. Ceci reste vrai même si on sait d'avance que cet état correspond

[12] Voir cependant les travaux de Sasaki et Carlini [166] pour le cas où nous disposons de plusieurs copies de l'état inconnu $|\psi_?\rangle$.

exactement à un des états de l'ensemble d'entraînement[13]. À partir de la description classique des états quantiques, il est possible de construire analytiquement le POVM optimal qui *minimise l'erreur d'entraînement.* Bien sûr, il faut aussi analyser comment cette approche peut *généraliser face à un nouvel état quantique non présent dans l'ensemble d'entraînement.* Cette question fondamentale sera abordée plus en détails dans la section 6.4.2.

Soit m_- le nombre d'états dans D_n pour lesquels $y_i = -1$ (classe négative), et son complément m_+ le nombre d'états dans D_n pour qui $y_i = +1$ (classe positive), tel que $m_- + m_+ = n$, le nombre total de points de données de D_n. De plus, p_- est la probabilité *a priori* de la classe négative, et p_+ est sa probabilité complémentaire pour la classe positive telles que $p_- + p_+ = 1$.

Définition 6.16 (Mélange statistique de la classe négative). *Le* mélange statistique représentant la classe négative ρ_-, *est défini comme étant* $\frac{1}{m_-}\sum_{i=1}^{n} I\{y_i = -1\} |\psi_i\rangle \langle\psi_i|$, *où* $I\{.\}$ *est la* fonction indicatrice *qui est égale à 1 si son argument est vrai et 0 sinon.*

Définition 6.17 (Mélange statistique de la classe positive). *De la même manière, le* mélange statistique représentant la classe positive ρ_+ *est défini comme étant* $\frac{1}{m_+}\sum_{i=1}^{n} I\{y_i = +1\} |\psi_i\rangle \langle\psi_i|$.

Le problème de classifier un état inconnu $|\psi_?\rangle$ choisi parmi les états de l'ensemble d'entraînement est équivalent à distinguer entre les mélanges ρ_- et ρ_+. Une des manières de s'en convaincre est de considérer le scénario suivant.

Scénario 6.4 (Préparation d'un état d'une classe par un démon[14]). *Imaginons un démon qui est à l'intérieur d'une boîte noire pourvue d'un seul bouton. À chaque fois que le bouton de la boîte noire est pressé, le démon choisit aléatoirement la classe négative ou la classe positive en fonction de leurs probabilités* a priori p_- *et* p_+. *Une fois la classe déterminée, le démon choisit uniformément au hasard un des états appartenant à cette classe et prépare l'état quantique correspondant (on suppose que le démon dans*

[13] À moins qu'on soit dans le cas trivial où tous les états sont mutuellement orthogonaux. Si c'est le cas, une *mesure non-destructive* dans une base où les états de l'ensemble d'entraînement sont des vecteurs propres de la base révélera l'état sans même le perturber.

[14] Il est possible de reformuler ce scénario en remplacant le démon par un algorithme probabiliste. La question se pose alors de savoir quelle quantité de mémoire classique devrait disposer l'algorithme.

son infinie puissance connaît la description classique des états et est capable de préparer parfaitement n'importe lequel d'entre eux). Cet état est ensuite produit en sortie par la boîte noire. Ainsi, trouver la classe de cet état revient à essayer de deviner quelle classe le démon[15] *a choisi lors de la première étape, mais pas forcément à identifier exactement cet état.*

Le taux d'erreur minimal de cette classification est lié au *chevauchement statistique* entre les deux mélanges ρ_- et ρ_+. Ce type de problème a déjà été étudié dans le contexte de la *théorie quantique de la détection et l'estimation* [110], domaine qui précède l'informatique quantique. Certains des résultats de ce domaine peuvent être utilisés pour *borner l'erreur d'entraînement* des algorithmes d'apprentissage quantique.

Théorème 6.2 (Mesure de Helstrom [110]). *Le taux d'erreur lorsqu'on cherche à distinguer entre les deux états mélangés ρ_- et ρ_+ avec un POVM est bornée en dessous par $\epsilon_{Hel} = \frac{1}{2} - \frac{D(\rho_-,\rho_+)}{2}$, où $D(\rho_-,\rho_+) = Tr|p_-\rho_- - p_+\rho_+|$ est une mesure de distance entre ρ_- et ρ_+ appelée* distance de trace *(où p_- et p_+ représentent les probabilités* a priori *des classes ρ_- et ρ_+ respectivement). De plus, cette borne peut être atteinte exactement par le POVM optimal appelé* mesure de Helstrom.

Corollaire 6.1 (Regret de la mesure de Helstrom). *La mesure de Helstrom est un classifieur binaire qui a* un regret nul, *c'est-à-dire $r_{Hel} = 0$.*

Démonstration. Le regret nul de la mesure de Helstrom découle directement de l'optimalité du POVM à distinguer entre les deux classes. □

Remarque 6.2 (Taux d'erreur de la mesure de Helstrom pour des classes équiprobables). *Supposons que la classe négative et la classe positive soient équiprobables. Si jamais ρ_- et ρ_+ sont des matrices densité qui correspondent au même état, la distance de trace $D(\rho_-,\rho_+)$ sera égale à zéro, ce qui veut dire que l'erreur ϵ_{Hel} de la mesure de Helstrom sera de $\frac{1}{2}$. À l'inverse, si les mélanges ρ_- et ρ_+ sont orthogonaux, cela signifie que $D(\rho_-,\rho_+) = 1$, et que donc la mesure de Helstrom aura une erreur $\epsilon_{Hel} = 0$.*

[15] Ici le rôle du démon est simplement de préparer l'état, mais pas de jouer le rôle d'un adversaire qui chercherait à tromper l'apprenant se trouvant à l'extérieur de la boîte.

Le but d'un algorithme d'apprentissage dans le contexte quantique est d'offrir une approche constructive qui permet d'atteindre (ou d'approcher) cette borne. Dans la situation où on dispose d'une description classique des états, cela correspond à trouver une implémentation efficace de la mesure de Helstrom. Si $D_n \in L_{qu}^{\otimes s}$, l'apprentissage devient plus ardu et il est difficile de caractériser le lien exact entre s le nombre de copies de chaque point d'entraînement dont on dispose, d la dimension de l'espace de Hilbert dans lequel vivent les états quantiques, et ϵ_{opt} l'erreur minimale que nous pouvons espérer atteindre. Contrairement à l'apprentissage machine classique, où il est toujours possible (même si non recommandé pour obtenir une bonne généralisation) d'amener l'erreur d'entraînement à zéro (par exemple en utilisant un classifieur qui garderait tout l'ensemble d'entraînement en mémoire comme les 1-plus proches voisins [71]), la situation est différente dans le contexte quantique comme exprimé par le lemme suivant.

Lemme 6.2 (Impossibilité d'atteindre une erreur d'entraînement nulle avec une seule copie d'un état inconnu). *Il est* impossible d'atteindre une erreur d'entraînement nulle dans le cas quantique *étant donné une seule copie d'un état inconnu à moins que les états de l'ensemble d'entraînement ne soient mutuellement orthogonaux entre eux.*

Démonstration. De par le théorème 6.2 et la remarque 6.2, il est impossible d'avoir un POVM permettant de classifier exactement un état inconnu quantique tiré d'un ensemble d'entraînement D_n à moins que tous les états de l'ensemble ne soient mutuellement orthogonaux, ou encore de manière équivalente que la distance entre les deux matrices densité soit $D(\rho_-, \rho_+) = 1$. □

Étant donné un nombre fini de copies de chaque état de l'ensemble d'entraînement, les stratégies d'apprentissage possibles incluent :

(1) l'estimation de l'ensemble d'entraînement par des mesures (conjointes ou non) sur certaines de ces copies afin de construire un POVM qui différencie entre les deux classes,

(2) la mise au point d'un mécanisme de classification qui utilise ces copies seulement quand le moment de classifier l'état inconnu $|\psi_?\rangle$ est arrivé ou

(3) une stratégie hybride basée sur (1) et (2).

Au niveau de la classification, plusieurs stratégies de mesure existent dans le contexte quantique dont celles de :

(a) *maximiser la probabilité de prédire la classe de l'état inconnu* (ce qui correspond à la mesure de Helstrom [110]).

(b) *minimiser la probabilité de faire une mauvaise prédiction.* Cette stratégie s'appelle la *discrimination non-ambiguë* [111] et est possible uniquement si les états quantiques de D_n sont *linéairement indépendants.* Dans ce cas spécifique, une mesure peut être mise au point qui est autorisée à répondre parfois "je ne sais pas", mais qui lorsqu'elle donne une réponse positive concernant l'une des classes offre une confiance de 100% sur le fait que sa réponse est correcte.

(c) *n'importe quelle stratégie entre ces deux extrêmes* (a) et (b). Une *mesure basée sur la confiance*[16] [63] est une mesure qui peut soit identifier la classe d'un état avec une certaine confiance (qui est connue), soit répondre "je ne sais pas" le reste du temps. L'objectif principal lorsqu'on construit une telle mesure, pour une confiance fixée déterminée par l'utilisateur, est de minimiser la probabilité que la mesure puisse répondre "je ne sais pas". Si on choisit de fixer la confiance à 100%, cela correspond à la mesure non-ambiguë, alors que si on ne souhaite jamais répondre de manière inconclusive cela correspond à la mesure de Helstrom. Il est parfois possible d'avoir une mesure basée sur la confiance (avec une confiance supérieure à celle de la mesure de Helstrom) même lorsque la discrimination non-ambiguë est impossible à cause de la forme de D_n (par exemple si les états sont linéairement dépendants).

Dans le reste de ce chapitre, nous nous focalisons uniquement (exception faite de la section 6.2.3.1) sur la stratégie de mesure qui consiste à maximiser la probabilité d'identifier correctement la classe d'un état (stratégie de mesure (a)) en apprenant à partir de l'ensemble d'entraînement un POVM qui jouera le rôle du classifieur (stratégie d'apprentissage (1)). Nous faisons aussi l'hypothèse que nous avons accès à un oracle appelé *oracle de Helstrom* (figure 6.3) qui est capable de résoudre la tâche de classification binaire.

Définition 6.18 (Oracle de Helstrom). *L'*oracle de Helstrom *est une construction abstraite qui prend en entrée : (1) une description classique des matrices densité ρ_- et ρ_+ et des probabilités* a priori *des classes p_- et p_+ (classe d'apprentissage L_{qu}^{cl}), (2) ou un nombre fini de copies de chaque état quantique de D_n (classe d'apprentissage $\mathsf{L}_{qu}^{\otimes\Theta(t_{bin})}$).*

[16] Le terme original en anglais est *maximum-confidence measurement.*

À partir de cette entrée, l'oracle peut être "entraîné" afin de retourner en sortie une implémentation efficace (exacte ou approximative) du POVM de la mesure de Helstrom f_{Hel}, sous forme d'un circuit permettant de distinguer entre les classes ρ_- et ρ_+. Dans le cas de la seconde variante de l'oracle, son coût d'entraînement t_{bin} correspond au nombre minimum de copies de chaque état de l'ensemble d'entraînement que l'oracle doit sacrifier afin de construire f_{Hel}.

FIG. 6.3 – Illustration de l'oracle d'Helstrom. L'oracle de Helstrom prend en entrée la description classique des matrices densité à distinguer ρ_- et ρ_+ (ainsi que leurs probabilités *a priori* p_- et p_+) (version 1) ou un nombre fini s de copies de chaque état quantique (version 2), pour $s \geq t_{bin}$. L'oracle produit en sortie un circuit f_{Hel} représentant une implémentation efficace de la mesure de Helstrom permettant de discriminer entre ρ_- et ρ_+.

Faire l'hypothèse de l'existence de cet oracle de Helstrom permet d'éviter d'avoir à décrire explicitement comment l'algorithme d'apprentissage qui jouerait le rôle de l'oracle fonctionne en réalité (et combien de copies des états quantiques seraient nécessaires pour qu'il puisse réaliser son apprentissage). Construire un algorithme d'apprentissage pour la classification binaire pouvant implémenter en pratique cet oracle est une question ouverte fondamentale.

Question ouverte 6.2 (Construction d'un algorithme d'apprentissage quantique implémentant l'oracle de Helstrom). *Est-il possible d'élaborer un algorithme d'apprentissage quantique qui implémente explicitement l'oracle de Helstrom ? Si oui, quelle serait*

la valeur de t_{bin}, le nombre minimum de copies de chaque état d'entraînement, que cet algorithme demande durant l'apprentissage ?

Quelques pistes possibles d'algorithmes d'apprentissage seront discutées dans la section 6.4.1. Pour l'instant, nous regarderons plutôt quelles autres tâches peuvent être résolues si on a accès à un tel oracle. Si jamais, nous connaissons un algorithme d'apprentissage ayant une erreur faible mais non optimale, il est possible de l'utiliser à la place de l'oracle de Helstrom dans la plupart des réductions décrites dans ce chapitre. Une autre question ouverte importante concerne l'existence ou non d'une implémentation efficace de la mesure de Helstrom.

Question ouverte 6.3 (Implémentation efficace de la mesure de Helstrom). *Quelles sont les situations d'apprentissage (autrement dit quels sont les ensembles de données quantiques) pour lesquelles il est possible d'implémenter efficacement (par exemple avec un circuit de taille polynomial) et de manière* exacte *la mesure de Helstrom ? De plus, quelles sont les situations d'apprentissage où cette implémentation efficace est possible de manière* approximative *?*

Il n'existe aucune garantie *a priori* que la description du POVM (ou de la matrice unitaire) qui correspond à la mesure de Helstrom puisse être réalisé physiquement par un circuit de taille polynomial par rapport au nombre de qubits passés en entrée au circuit. En effet dans le pire des cas, il se peut que ce circuit requiert un nombre de portes qui est exponentiel par rapport à la taille de l'entrée, et ce même dans sa version approximative.

Afin de se réchauffer sur l'usage des réductions d'apprentissage et ce que nous pouvons réaliser si on a accès à un oracle de Helstrom, considérons le cas où on s'intéresse à la tâche d'estimer *la probabilité d'appartenance d'un état inconnu à la classe négative et à la classe positive.*

Tâche d'apprentissage quantique 6.2 (Estimation de la probabilité d'appartenance à une classe). ***Entrée*** *: $D_n = \{(|\psi_1\rangle, y_1), \ldots, (|\psi_n\rangle, y_n)\}$, un ensemble d'entraînement quantique, où $|\psi_i\rangle \in \mathbb{C}^{2^d}$ et $y_i \in \{-1, +1\}$.*
Sortie *: un POVM jouant le rôle d'un estimateur de probabilité f qui peut prédire l'appartenance à la classe négative et positive d'un état quantique inconnu $|\psi_?\rangle$, soit respectivement $prob(y_? = -1||\psi_?\rangle)$ et $prob(y_? = +1||\psi_?\rangle)$.*

***But** : construire un estimateur f, qui à partir de quelques copies d'un état inconnu $|\psi_?\rangle$, peut produire un estimé fiable de sa probabilité d'appartenance à la classe négative et à la classe positive.*

Réduction 6.1 (Réduction de l'estimation de la probabilité d'appartenance à une classe à la classification binaire (via oracle de Helstrom)). *Étant donné l'accès à un oracle de Helstrom, il est possible de* réduire la tâche de l'estimation de la probabilité d'appartenance à une classe à la tâche de classification binaire.
***Coût de l'entraînement** : $\Theta(t_{bin})$ (ou $\Theta(1)$ si $D_n \in \mathsf{L}_{qu}^{cl}$).*
***Coût de classification** : $\Theta(1)$, pour le nombre constant de copies de l'état inconnu qui seront sacrifiées.*

Démonstration. En utilisant l'oracle de Helstrom, nous pouvons produire un circuit jouant le rôle d'un classifieur binaire f qui peut distinguer entre ρ_+ et ρ_-. Ensuite, à partir de ce classifieur binaire f et plusieurs copies d'un état inconnu $|\psi_?\rangle$, nous pouvons *estimer sa probabilité d'appartenance* à la classe positive et négative efficacement. Soit c le nombre constant de copies de l'état inconnu $|\psi_?\rangle$ dont nous disposons, il suffit d'appliquer c fois le classifieur f sur chacune de ces copies pour obtenir un estimé de son appartenance à la classe négative et la classe positive. Le coût de classification de cette réduction est donc de $\Theta(1)$ et le coût d'entraînement de $\Theta(t_{bin})$ si on utilise la deuxième version de l'oracle de Helstrom (et $\Theta(1)$ si on utilise la première version pour le cas où $D_n \in \mathsf{L}_{qu}^{cl}$ car on appelle alors l'oracle une seule fois). □

Corollaire 6.2. *La tâche d'*estimer la probabilité d'appartenance d'un état inconnu à une classe *est dans la classe d'apprentissage $\mathsf{L}_{qu}^{\otimes\Theta(t_{bin})}$ pour son entraînement et dans la classe $\mathsf{L}_{qu}^{\otimes\Theta(1)}$ pour sa classification.*

Dans le monde quantique, nous pouvons résoudre aisément cette tâche d'estimation de probabilités à cause de la *nature probabiliste* inhérente à l'acte de la mesure. La situation est différente en apprentissage machine classique où certains classifieurs sont de *nature déterministe*, voulant dire qu'ils prédisent toujours la même classe si jamais on leur présente plusieurs fois de suite le même exemple. Nous pouvons cependant contourner ce problème :

- soit en utilisant un type de classifieur qui est "naturellement" conçu pour produire un estimé de la probabilité d'appartenance aux différentes classes (comme un classifieur bayésien [135]),
- soit en utilisant une technique générique comme la réduction "*Probing*" [134] qui réduit l'estimation des probabilités d'appartenance à la classification binaire.

6.2.2 Classification binaire pondérée

La tâche de *classification binaire pondérée* est similaire à celle de la classification binaire simple, sauf que chaque point de données a maintenant un *poids* w qui lui est associé indiquant l'importance de classifier correctement cet exemple. Ce poids peut par exemple être choisi en fonction de la pénalité que nous devrions payer si on fait une mauvaise prédiction à son sujet.

Tâche d'apprentissage quantique 6.3 (Classification binaire pondérée). ***Entrée*** *:* $D_n = \{(|\psi_1\rangle, y_1, w_1), \ldots, (|\psi_n\rangle, y_n, w_n)\}$*, un ensemble d'entraînement quantique, où* $|\psi_i\rangle \in \mathbb{C}^{2^d}$*,* $y_i \in \{-1, +1\}$ *et* $w_i \in [0, +\infty)$*.*
Sortie *: un POVM jouant le rôle d'un classifieur binaire* f *qui peut prédire la classe* $y_?$ *d'un état quantique inconnu* $|\psi_?\rangle$*.*
But *: construire un classifieur binaire* f *qui minimise l'erreur d'entraînement pondérée* $\epsilon_f = \frac{1}{n}\sum_{i=1}^{n} w_i prob(f(|\psi_i\rangle) \neq y_i)$*.*

Une fois de plus, si nous nous trouvons dans la situation idéale où on connaît la description classique des états (c'est-à-dire $D_n \in \mathsf{L}_{qu}^{cl}$), leurs poids peuvent être incorporés directement dans la description des matrices densité de leurs classes. La réduction suivante formalise comment résoudre la tâche de classification binaire pondérée dans cette situation étant donné l'accès à un oracle de Helstrom (première version).

Réduction 6.2 (Réduction de la classification binaire pondérée à la classification binaire simple (via oracle de Helstrom)). *Étant donné l'accès à un oracle de Helstrom qui prend en entrée la description des matrices de densité* ρ_- *et* ρ_+ *(et leurs probabilités* a priori *p_- et p_+), il est possible de* réduire la tâche de classification pondérée à la tâche de classification binaire simple.
Coût de l'entraînement *:* $\Theta(1)$*.*
Coût de la classification *:* $\Theta(1)$*.*

Démonstration. Le poids w_i d'un état particulier $|\psi_i\rangle$ peut être converti en une probabilité p_i reflétant son importance en choisissant

$$p_i = \frac{w_i}{\sum_{j=1}^{n} w_j}. \tag{6.3}$$

Soit $\hat{p}_-$, la nouvelle probabilité *a priori* de la classe négative, qui est égale à

$$\hat{p}_- = \sum_{i=1}^{n} p_i I\{y_i = -1\} \tag{6.4}$$

et $\hat{p}_+$, sa probabilité complémentaire telles que $\hat{p}_- + \hat{p}_+ = 1$. Le théorème 6.3 démontre que la mesure de Helstrom qui discrimine entre les matrices densité des classes dans lesquelles on incorpore ces poids est aussi directement le POVM qui minimise l'erreur pondérée. Il suffit donc d'appeler l'oracle de Helstrom en lui donnant comme entrée les matrices densité

$$\hat{\rho}_- = \sum_{i=1}^{n} p_i I\{y_i = -1\} |\psi_i\rangle \langle\psi_i| \tag{6.5}$$

et

$$\hat{\rho}_+ = \sum_{i=1}^{n} p_i I\{y_i = +1\} |\psi_i\rangle \langle\psi_i| \tag{6.6}$$

(avec probabilités *a priori* $\hat{p}_-$ et $\hat{p}_+$). De plus, la réduction utilise un seul appel à l'oracle de Helstrom et requiert seulement une copie de l'état inconnu lors de la classification, d'où un coût d'entraînement et de classification de $\Theta(1)$. □

Théorème 6.3 (Mesure de Helstrom minimisant l'erreur pondérée). *La mesure de Helstrom qui minimise l'erreur de discrimination entre les deux matrices de densité* $\hat{\rho}_- = \sum_{i=1}^{n} p_i I\{y_i = -1\} |\psi_i\rangle \langle\psi_i|$ *et* $\hat{\rho}_+ = \sum_{i=1}^{n} p_i I\{y_i = +1\} |\psi_i\rangle \langle\psi_i|$ *(avec probabilités* a priori $\hat{p}_-$ *et* $\hat{p}_+$*) est aussi la mesure qui minimise l'erreur de classification pondérée de l'ensemble de données* $D_n = \{(|\psi_1\rangle, y_1, w_1), \ldots, (|\psi_n\rangle, y_n, w_n)\}$.

Démonstration. La mesure de Helstrom est le POVM f qui minimise l'erreur de discrimination entre $\hat{\rho}_-$ et $\hat{\rho}_+$. Ce POVM se décompose en deux éléments Π_- et Π_+ qui correspondent tous les deux à des matrices positives semi-définies tel que $\Pi_- + \Pi_+ = \mathsf{I}$,

où I est la matrice identité. On a donc :

$$\epsilon_{Hel} = \min_f(\mathrm{Tr}(\Pi_-\hat{\rho}_+) + \mathrm{Tr}(\Pi_+\hat{\rho}_-)) \tag{6.7}$$

qui peut aussi s'exprimer comme

$$\epsilon_{Hel} = \min_f \left(\sum_{i=1}^{n} p_i I\{y_i = +1\} \mathrm{Tr}(\Pi_- |\psi_i\rangle \langle\psi_i|) + \sum_{i=1}^{n} p_i I\{y_i = -1\} \mathrm{Tr}(\Pi_+ |\psi_i\rangle \langle\psi_i|) \right) \tag{6.8}$$

et se simplifie en

$$\epsilon_{Hel} = \min_f \left(\sum_{i=1}^{n} p_i \mathrm{Prob}(f(|\psi_i\rangle) \neq y_i) \right) \tag{6.9}$$

ce qui revient aussi à minimiser l'erreur pondérée :

$$\epsilon_{opt} = \frac{1}{n} \sum_{j=1}^{n} w_j \times \epsilon_{Hel} = \min_f \left(\frac{1}{n} \sum_{j=1}^{n} w_j \sum_{i=1}^{n} p_i \mathrm{Prob}(f(|\psi_i\rangle) \neq y_i) \right) \tag{6.10}$$

$$\epsilon_{opt} = \min_f \left(\frac{1}{n} \sum_{i=1}^{n} w_i \mathrm{Prob}(f(|\psi_i\rangle) \neq y_i) \right) \tag{6.11}$$

Comme ce POVM est optimal, cela implique aussi qu'il a un regret nul. □

Algorithme 14 echantillonnage_rejet($D_n \in \mathsf{L}_{qu}^{\otimes\Theta(t_{bin})}$)

Choisir une constante c plus grande que n'importe quel poids w
Pour chaque état $|\psi_i\rangle^{\otimes\Theta(t_{bin})}$ **faire**
  Tirer à pile ou face en utilisant une pièce avec un *biais* de $\frac{w_i}{c}$
  (ce qui signifie qu'avec probabilité $\frac{w_i}{c}$, "face" sera observé, et avec probabilité complémentaire, $1 - \frac{w_i}{c}$, "pile" sera observé)
  Si le résultat est "face" **alors**
    Garder les copies de l'état
  Sinon
    Les mettre de côté
  fin si
fin pour
Retourner la nouvelle distribution $\widetilde{D}$ générée

Dans le cas où seul un nombre fini de copies de chaque état quantique est accessible mais nous connaissons une façon de produire un classifieur binaire efficace (comme un

oracle de Helstrom), alors on peut utiliser la réduction "*costing*" [195] pour réduire la classification binaire pondérée à sa version de base. Cette réduction est basée sur un mécanisme d'échantillonnage par rejet (algorithme 14) et sur l'agrégation de plusieurs classifieurs (algorithme 15), et génère un ensemble de T classifieurs binaires, où T est une petite constante choisie indépendamment de D_n.

Algorithme 15 costing_entrainement($D_n \in \mathsf{L}_{qu}^{\otimes\Theta(Tt_{bin})}$)

Pour $j = 1$ à T **faire**
  Appeler echantillonnage_rejet($D_n \in \mathsf{L}_{qu}^{\otimes\Theta(t_{bin})}$) pour obtenir $\widetilde{D}_j$
  Appeler l'oracle de Helstrom sur $\widetilde{D}_j$ pour apprendre le classifieur binaire f_j
fin pour
Retourner le classifieur final $f = \text{majorite}(f_1, \ldots, f_T)$

La sortie finale du classifieur est un vote de majorité sur les sorties des classifieurs individuels[17]. Le nombre d'évaluations faite sur le classifieur final est une constante $\Theta(T)$, qui est proportionnelle au nombre de classifieurs binaires formant le classifieur agrégé (algorithme 16). Il est clair que plus on fait d'évaluations du classifieur final, plus on augmente la précision de la classification, mais aussi plus on a besoin de copies de l'état inconnu $|\psi_?\rangle$.

Algorithme 16 costing_classification($|\psi_?\rangle^{\otimes\Theta(T)}, f = (f_1, \ldots, f_T)$)

Pour $j = 1$ à T **faire**
  Mesurer $y_j = f_j(|\psi_?\rangle)$
fin pour
Retourner $y_? = \text{majorite}(y_1, \ldots, y_T)$

Réduction 6.3 (Réduction de la classification binaire pondérée à la classification binaire simple (via costing [195])). *Étant donné l'accès à un oracle de Helstrom et un ensemble d'entraînement* $D_n \in L_{qu}^{\otimes\Theta(Tt_{bin})}$, *il est possible de* réduire la tâche de classification pondérée à la tâche de classification binaire simple.

Coût de l'entraînement : $\Theta(Tt_{bin})$.

Coût de la classification : $\Theta(T)$.

[17] Cette réduction a une très forte ressemblance avec l'algorithme de bagging [48], qui est une méthode d'ensemble (cf. section 3.2.3) basée elle aussi sur un mécanisme d'échantillonnage et l'agrégation de plusieurs classifieurs.

Démonstration. Lors de l'entraînement, l'algorithme costing_entrainement fait appel à l'oracle de Helstrom T fois, pour T une constante choisie indépendamment de l'ensemble d'entraînement D_n. Le coût de l'entraînement est donc de $\Theta(Tt_{bin})$, c'est-à-dire le nombre d'appels à l'oracle de Helstrom multiplié par t_{bin} le nombre de copies des états d'entraînement requis à chaque appel. Comme chaque appel à l'oracle de Helstrom produit un classifieur, le coût de la classification sera lui de $\Theta(T)$, ce qui revient à utiliser une copie de l'état inconnu $|\psi_?\rangle$ pour chaque classifieur généré. De par l'analyse de la réduction costing [195], il est garanti que la moyenne des erreurs d'entraînement standards que minimisent les classifieurs individuels $f_1, \ldots, f_T$ sur les distributions $\widetilde{D}_1, \ldots, \widetilde{D}_T$ revient aussi indirectement à minimiser approximativement l'erreur d'entraînement pondérée du classifieur global f, c'est-à-dire :

$$\epsilon_f \sim \min_f \frac{1}{n} \sum_{i=1}^{n} w_i \text{prob}(f(|\psi_i\rangle) \neq y_i) \tag{6.12}$$

□

Corollaire 6.3. *La* tâche de classification binaire pondérée *appartient à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(Tt_{bin})}$ *pour la phase d'entraînement et à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(1)}$ *pour la classification.*

La méthode quantique d'échantillonnage par rejet (algorithme echantillonnage_rejet) a comme bénéfice additionnel de permettre "d'économiser" certaines copies d'états quantiques lors de la génération de la distribution biaisée en fonction des poids des états. En effet, les états qui ont un poids faible ont une probabilité plus importante que les autres de ne pas être retenus dans la nouvelle distribution générée. Ainsi, les copies des états qui sont mis de côté pourront être utilisés plus tard, par exemple lors d'une nouvelle étape d'échantillonnage par rejet.

6.2.3 Classification multiclasse

Dans la version multiclasse de la classification, chaque état est étiqueté d'après une classe choisie parmi k, pour $k > 2$. Le but est de construire un classifieur f qui étant donné un nombre fini de copies d'un état inconnu $|\psi_i\rangle$ peut prédire la classe $y_?$ avec une bonne précision.

Tâche d'apprentissage quantique 6.4 (Classification multiclasse).
Entrée *: $D_n = \{(|\psi_1\rangle, y_1), \ldots, (|\psi_n\rangle, y_n)\}$, un ensemble d'entraînement quantique, où $|\psi_i\rangle \in \mathbb{C}^{2^d}$ et $y_i \in \{1, \ldots, k\}$.*
Sortie *: un POVM jouant le rôle d'un classifieur binaire f qui peut prédire la classe $y_?$ d'un état quantique inconnu $|\psi_?\rangle$.*
But *: construire un classifieur binaire f qui minimise l'erreur d'entraînement $\epsilon_f = \frac{1}{n}\sum_{i=1}^{n} prob(f(|\psi_i\rangle) \neq y_i)$.*

Passer du cas binaire au cas multiclasse est loin d'être aisé, et très peu de choses sont connues pour le cas où le nombre de classes k est supérieur à 2. En particulier même à partir de seulement trois classes, la forme exacte du POVM optimal qui permet de distinguer entre ces classes étant donné une seule copie de l'état n'est pas connue. Nous allons cependant voir dans la section 6.2.3.4 que lorsque l'on connaît la description classique des états, on peut construire une mesure (appelée *Pretty Good Measurement* [107] en anglais) dont l'erreur est bornée par la racine carrée de l'erreur du POVM optimal.

Les sections suivantes décrivent différentes stratégies d'entraînement et de classification pour les cas où on dispose d'un nombre de copies de l'état inconnu $|\psi_?\rangle$ à classifier qui est respectivement :

- linéaire en n, le nombre d'états présents dans D_n (section 6.2.3.1).
- linéaire en k, le nombre de classes de D_n (section 6.2.3.2).
- logarithmique en k (section 6.2.3.3).
- une unique copie ou possiblement un nombre constant (section 6.2.3.4).

6.2.3.1 Classification par identification d'état

Une des manières les plus directes de reconnaître la classe d'un état consiste à *identifier exactement cet état.* Une fois cet état identifié, cette information permet aussi de retrouver sa classe (à moins qu'il y ait deux états qui soient identiques mais étiquetés par deux classes différentes). Si nous disposons de $\Theta(n)$ copies de l'état inconnu $|\psi_?\rangle$, nous pouvons utiliser le C-Swap test [17, 53] entre cet état et tour à tour chacun des états de l'ensemble d'entraînement $D_n \in \mathsf{L}_{qu}^{\otimes\Theta(1)}$. Cette méthode ne requiert aucun travail durant l'entraînement, tout le travail s'effectuant au moment de la classification (ce qui correspond à une stratégie d'apprentissage type (2) dans la section 6.2.1). Elle peut

être vue comme l'analogue quantique de la méthode des 1-plus-proches voisins [71]. En effet, on recherche parmi tous les états de l'ensemble de données celui qui est le plus proche/similaire (dans le sens de la fidélité) de l'état inconnu. Pour peu qu'il n'y ait pas deux états quantiques dans D_n qui sont identiques mais étiquetés par deux classes différentes, cette méthode offre une erreur de classification nulle (et donc un regret nul aussi). L'algorithme suivant formalise cette méthode.

Algorithme 17 identification_classification($|\psi_?\rangle^{\otimes\Theta(n)}$, $D_n \in \mathsf{L}_{qu}^{\otimes\Theta(1)}$)

Pour $i = 1$ à n **faire**
Mesurer la fidélité entre $|\psi_?\rangle$ et $|\psi_i\rangle$ en utilisant le C-Swap test ce qui donne un estimé de $Fid(|\psi_?\rangle, |\psi_i\rangle)$
fin pour
Retourner la classe y_j de l'état dont la fidélité avec l'état à classifier est maximal, soit $\arg\max_j Fid(|\psi_?\rangle, |\psi_j\rangle)$

Théorème 6.4 (Classification par identification d'état). *L'algorithme* identification_classification *permet de classifier un état inconnu* $|\psi_?\rangle$ *avec une erreur de classification nulle étant donné* $\Theta(cn)$ *copies de cet état et* $\Theta(c)$ *copies de chaque état de* D_n*, où* c *est une constante qui dépend de la fidélité minimale parmi toutes les paires d'états de* D_n*.*

Démonstration. Chaque C-Swap test utilise un nombre de copies constant c, où c est une constante qui dépend de la fidélité minimale parmi toutes les paires d'états de D_n, et comme on va évaluer la similarité entre $|\psi_?\rangle$ et tous les n états quantiques de D_n, le coût global de identification_classification sera de $\Theta(cn)$ copies de l'état inconnu et de $\Theta(c)$ copies de chaque état de l'ensemble d'entraînement. De plus, s'il n'existe pas deux états dans D_n qui sont identiques mais étiquetés par deux classes différentes, l'algorithme est garanti d'obtenir une erreur de classification nulle (ce qui implique aussi un regret nul). □

Corollaire 6.4. *La* tâche de classification multiclasse *appartient à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(1)}$ *pour la phase d'entraînement et à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(n)}$ *pour la classification.*

Si on souhaite baser la prédiction de la classe de $|\psi_?\rangle$ sur ses k plus proches voisins, et non pas seulement par rapport au plus proche voisin, l'algorithme

identification_classification peut être facilement adapté pour baser sa prédiction sur la majorité des classes de ses k plus proches voisins. Il suffit pour cela d'identifier les k états de D_n qui sont les plus similaires de $|\psi_?\rangle$ et de faire un vote de majorité sur leurs classes (les coûts d'entraînement et de classification restants inchangés). Une question ouverte importante est de déterminer s'il existe un équivalent quantique à des structures de données permettant de faciliter la recherche des plus proches voisins, telles que les kd-trees [27] par exemple. Quantiquement, la raison d'être d'une telle structure serait de permettre de rechercher les plus proches voisins d'un état inconnu en consommant moins de copies qu'avec la méthode naïve directe (par exemple en utilisant un nombre de copies logarithmique en n et linéaire en c le nombre de voisins considérés). On peut formaliser cette question ouverte de la manière suivante.

Question ouverte 6.4 (Structure de données favorisant la recherche des plus proches voisins dans un ensemble de données quantique). *Existe-il une structure de données quantique qui permet de trouver le plus proche voisin (mesuré par la fidélité) d'un état inconnu $|\psi_?\rangle$ parmi un ensemble de n états quantiques plus rapidement que la méthode directe qui requiert un nombre de copies linéaire en n ? Si oui, quel est le coût d'entraînement de cette méthode ? Et son coût de classification pour trouver le plus proche voisin ?*

6.2.3.2 Réduction du type une classe contre tous

Algorithme 18 une_classe_contre_toute_entrainement($D_n \in \mathsf{L}_{qu}^{\otimes\Theta(kt_{bin})}$)

Pour $j = 1$ à k **faire**
  Initialiser $D^{(j)}$ comme étant l'ensemble de données vide
  Pour $i = 1$ à n **faire**
    Ajouter l'exemple $(|\psi_i\rangle^{\otimes\Theta(t_{bin})}, 1 - 2I\{y_i = j\})$ à $D^{(j)}$
  fin pour
  Appeler l'oracle de Helstrom sur l'ensemble de données $D^{(j)}$ pour apprendre un classifieur binaire f_j qui différencie entre la classe j et l'union de toutes les autres classes
fin pour
Retourner l'ensemble des classifieurs binaires $f_1, \ldots, f_j$

L'idée de la réduction d'une classe contre tous [74] (appelée *one-against-all* en anglais) est d'entraîner un classifieur binaire pour chacune des k classes. Chacun de ces classifieurs binaires *différencie entre sa propre classe et l'union de toutes les autres classes.* Cette

réduction peut être adaptée au contexte quantique en construisant pour chacune des classes un POVM jouant le rôle d'un classifieur binaire, qui discrimine entre la matrice densité de cette classe spécifique et un mélange statistique composé des matrices densité des autres classes. Nous disons d'un classifieur qu'il "clique" s'il prédit que l'état qu'il a mesuré $|\psi_?\rangle$ appartient à la classe qu'il incarne, et qu'il ne "clique pas" sinon. Étant donné l'accès à un oracle de Helstrom, il est possible de réduire la classification multiclasse à la classification binaire en utilisant les algorithmes d'entraînement et de classification suivants (algorithmes 18 et 19).

Algorithme 19 une_classe_contre_toute_classification$(|\psi_?\rangle^{\otimes k})$

Pour $j = 1$ à k **faire**
  Appliquer le classifieur binaire f_j sur $|\psi_?\rangle$ pour obtenir la prédiction si cet état appartient à la classe j ou non
fin pour
Si un seul classifieur a "cliqué" **alors**
  Retourner la classe correspondant au classifieur qui a "cliqué"
Sinon
  Si plusieurs classifieurs ont "cliqués" **alors**
    Retourner une classe choisie au hasard parmi tous les classifieurs qui ont "cliqués"
  Sinon
    Retourner une classe choisie uniformément au hasard parmi les k classes
  fin si
fin si

Réduction 6.4 (Réduction de la classification multiclasse à la classification binaire simple (du type une classe contre tous)). *Étant donné l'accès à un oracle de Helstrom et un ensemble d'entraînement* $D_n \in L_{qu}^{\otimes\Theta(kt_{bin})}$, *il est possible de* réduire la tâche de classification multiclasse à la tâche de classification binaire via une réduction du type une classe contre tous.

Coût de l'entraînement : $\Theta(kt_{bin})$.

Coût de la classification : $\Theta(k)$.

Démonstration. L'algorithme une_classe_contre_toute_entrainement fait appel à l'oracle de Helstrom une nombre de fois qui est linéaire dans le nombre de classes k, et chaque appel consomme un nombre de copies des états de D_n de $\Theta(t_{bin})$. Le coût d'entraînement de la réduction est donc de $\Theta(kt_{bin})$ copies. Au niveau de la classification, il faut sacrifier

une copie de l'état inconnu $|\psi_?\rangle$ pour chacun des k classifieurs binaires générés, ce qui conduit à un coût total de $\Theta(k)$.

Au niveau de l'analyse de l'erreur de cette réduction, soit ϵ l'erreur moyenne des classifieurs binaires générés. La pire situation qui puisse arriver est que le classifieur de la "bonne" classe ne réagisse pas (ce qui correspond à un faux négatif). Dans ce cas et si aucun autre classifieur n'a "cliqué", on choisira la classe à prédire uniformément au hasard ce qui conduit à une erreur avec probabilité $\frac{k-1}{k}$. Dans le cas des faux positifs, c'est-à-dire si c classifieurs "cliquent" alors qu'ils ne devraient pas car ils ne correspondent pas à la bonne classe, la probabilité d'erreur sera de seulement $\frac{c}{c+1}$ puisqu'on choisira aléatoirement parmi tous les classifieurs qui ont réagi. Comme chaque classifieur binaire conduit à une erreur de $\frac{k-1}{k}$ dans le pire cas avec probabilité ϵ et qu'il y a k classifieurs, l'erreur globale du classifieur combiné sera dans le pire cas de $\frac{(k-1)k}{k}\epsilon = (k-1)\epsilon$. (Cette réduction ne semble offrir aucune garantie par contre en ce qui concerne le regret.) □

Corollaire 6.5. *La* tâche de classification multiclasse *appartient à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(kt_{bin})}$ *pour la phase d'entraînement et à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(k)}$ *pour la classification.*

Remarque 6.3 (Difficultés des situations d'apprentissage générées par la réduction). *Rien ne garantit* a priori *que les situations d'apprentissage intermédiaires générées par la réduction (ici les k situations de classification binaire) soient faciles à résoudre. Ainsi, même si l'accès à un oracle de Helstrom garantit que le classifieur binaire produit sera optimal pour chacune des k situations binaires, il est possible que l'erreur moyenne observée ϵ soit importante. Dans le cas quantique, cela peut arriver par exemple si la distance de trace entre la matrice densité d'une classe et le mélange composé de l'union des autres classes est faible (ce qui implique qu'il est difficile de les distinguer).*

Il existe une variante pondérée de cette réduction [34] (appelée *weighted one-against-all* en anglais) qui offre une meilleure garantie en terme d'erreur que la version de base. Cette variante exploite le fait que les faux négatifs (autrement dit le fait de manquer de détecter la bonne classe) sont plus dommageables pour l'erreur du classifieur global que les faux positifs (le fait de détecter une mauvaise classe). Il devient possible de faire baisser l'erreur en pire cas du classifieur global en incorporant des poids plus importants aux exemples qui risquent d'amener à observer des faux négatifs. (En pratique, cela veut

dire qu'un point va avoir un poids plus important lors de la construction du classifieur qui correspond à sa classe.) L'algorithme procède en réduisant d'abord la classification multiclasse à la classification binaire pondérée et utilise ensuite la réduction costing [195] pour réduire la classification binaire pondérée à la classification binaire standard. Le principal avantage de cette version pondérée de la réduction est qu'elle offre une garantie sur la borne de l'erreur du classifieur global de $\frac{k}{2}\epsilon$, pour ϵ l'erreur moyenne des classifieurs binaires générés, ce qui est divisée par deux par rapport à la version standard décrite précédemment. Dans ce cas-là, le coût d'entraînement total de la variante pondérée d'une classe contre tous serait de $\Theta(kTt_{bin})$ où k est le nombre de classes, T le nombre constant de classifieurs créés lors de la réduction costing et t_{bin} le nombre de copies utilisées par chaque appel de l'oracle de Helstrom. Le coût de classification sera quant à lui de $\Theta(kT)$. Quantiquement et si on connaît la description classique des états de l'ensemble d'entraînement (soit $D_n \in \mathsf{L}_{qu}^{cl}$), on peut remplacer la réduction de type costing par la réduction via l'oracle de Helstrom (réduction 6.2) ce qui donne un coût d'entraînement de $\Theta(kt_{bin})$ et de classification de $\Theta(k)$.

6.2.3.3 Réduction sous forme d'arbre binaire

Algorithme 20 arbre_binaire_entrainement($D_n \in \mathsf{L}_{qu}^{\otimes\Theta(t_{bin}\log k)}$)

Si tous les points de D_n appartiennent à la même classe **alors**
  Créer une feuille étiquetée d'après cette unique classe
  Retourner
fin si
Choisir aléatoirement deux sous-ensembles de classes Y_a et Y_b parmi D_n tel que $|Y_a| \approx |Y_b|$
Séparer l'ensemble de données D_n en deux sous-ensembles D_a et D_b en fonction des sous-ensembles de classes Y_a et Y_b (soit ρ_a la matrice densité représentant le sous-ensemble D_a et ρ_b la matrice densité représentant le sous-ensemble D_b)
Appeler l'oracle de Helstrom pour apprendre un classifieur binaire $f_{(\rho_a,\rho_b)}$ qui va distinguer entre les deux matrices densité ρ_a et ρ_b
Créer un nœud dans l'arbre binaire dont le test correspond au classifieur $f_{(\rho_a,\rho_b)}$
Appeler arbre_binaire_entrainement(D_a)
Appeler arbre_binaire_entrainement(D_b)

Une autre manière de résoudre le problème de la classification multiclasse est de construire un arbre binaire où *chaque nœud est un classifieur binaire qui différencie*

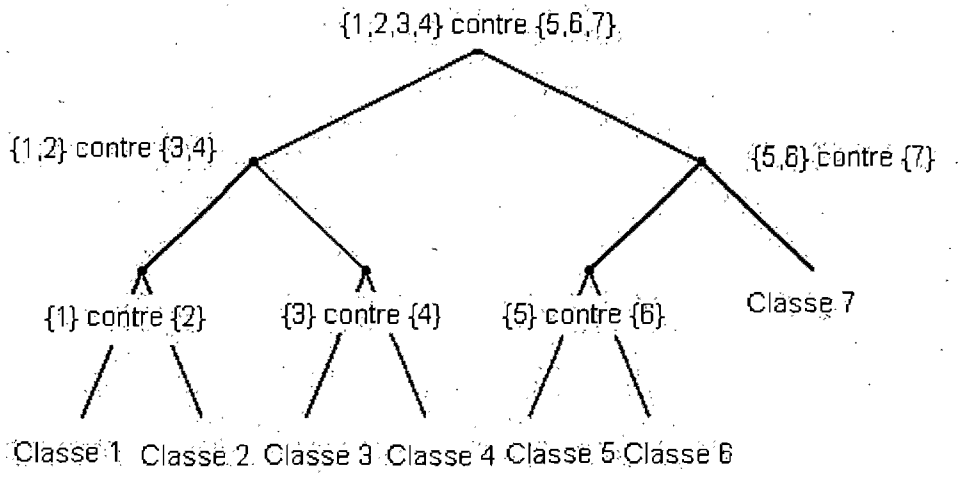

FIG. 6.4 – Illustration d'un arbre de classification binaire. Chaque nœud de l'arbre est un classifieur binaire qui différencie entre deux sous-ensembles de classes. La classification se fait en parcourant l'arbre depuis la racine jusqu'à atteindre une feuille qui est étiquetée d'après une classe. Dans cet exemple, le choix des étiquettes est arbitraire et ne reflète aucunement une relation particulière entre les classes.

entre deux sous-ensembles de classes et où *les feuilles sont étiquetées d'après une classe spécifique* (figure 6.4). La racine contient l'ensemble de toutes les classes et utilise un classifieur binaire pour diviser cet ensemble en deux sous-ensembles de classes de taille approximativement équilibrée par rapport au nombre de classes. Pour classifier un état inconnu, on part de la racine et on descend l'arbre en fonction du résultat du classifieur binaire observé à chaque nœud jusqu'à atteindre une feuille, auquel cas on prédit la classe associée à cette feuille. Il est possible de construire l'arbre binaire de plusieurs façons (par exemple en commençant par le créer depuis les feuilles jusqu'à la racine ou *vice-versa*), qui peuvent donner une erreur globale différente sur le classifieur généré. Les algorithmes 20 et 21 détaillent une manière possible pour construire l'arbre récursivement depuis la racine jusqu'aux feuilles et de l'utiliser ensuite pour faire de la classification.

Réduction 6.5 (Réduction de la classification multiclasse à la classification binaire simple (du type arbre binaire)). *Étant donné l'accès à un oracle de Helstrom et un ensemble d'entraînement* $D_n \in \mathsf{L}_{qu}^{\otimes\Theta(t_{bin}\log k)}$, *il est possible de* réduire la tâche de classification multiclasse à la tâche de classification binaire via une réduction du type arbre binaire.

Algorithme 21 arbre_binaire_classification($|\psi_?\rangle^{\otimes\Theta(\log k)}$, un classifieur f ayant la forme d'un arbre binaire de classification)

Commencer le parcours de l'arbre au noeud qui correspond à la racine
Tant que une feuille n'est pas atteinte **faire**
  Utiliser une copie de l'état $|\psi_?\rangle$ dans le classifieur binaire correspondant au noeud actuel
  Si le classifieur prédit la classe négative **alors**
    Descendre à gauche dans l'arbre
  Sinon
    Descendre à droite dans l'arbre
  fin si
fin tant que
Retourner la classe étiquettée par la feuille

Coût de l'entraînement : $\Theta(t_{bin} \log k)$.

Coût de la classification : $\Theta(\log k)$.

Démonstration. Lors de la construction de l'arbre binaire, l'oracle de Helstrom est appelé un nombre de fois qui est directement proportionnel au nombre de nœuds qui composent l'arbre. Cependant chaque appel à l'oracle sépare l'ensemble de données utilisée pour l'entraînement en deux sous-ensembles (dont la somme des tailles est égale à celle de l'ensemble original), cela implique qu'à chaque niveau de l'arbre le nombre d'états quantiques utilisés au total par l'oracle de Helstrom est de $\Theta(t_{bin})$. Le coût global de l'entraînement est donc de $\Theta(t_{bin} \log k)$ copies. Comme l'arbre est construit de manière à être équilibré, il aura une profondeur maximum de $\log k$, pour k le nombre de classes. En ce qui concerne le coût de la classification, il est directement proportionnel à la profondeur de l'arbre, soit $\Theta(\log k)$.

L'erreur globale du classifieur final que représente l'arbre binaire est au pire de $\epsilon \log k$, pour ϵ l'erreur moyenne des classifieurs binaires générés. En effet, une erreur peut arriver avec probabilité ϵ à chaque nœud traversé ce qui implique une erreur globale de $\epsilon \log k$ dans le pire cas. □

Corollaire 6.6. *La* tâche de classification multiclasse *appartient à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(t_{bin} \log k)}$ *pour la phase d'entraînement et à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(\log k)}$ *pour la classification.*

Théorème 6.5 (Identification d'état). *Soit un ensemble de données quantique D_n composé de n états purs tel qu'il n'y a pas deux états quantiques identiques. Il existe un POVM qui peut identifier l'index d'un état inconnu $|\psi_?\rangle$ choisi parmi un ensemble possible de D_n avec une très bonne précision étant donné $\Theta(\log n)$ copies de cet état.*

Démonstration. La preuve est relativement directe, il suffit simplement de choisir $k = n$, c'est-à-dire d'assigner une classe différente à chacun des n points de l'ensemble de données D_n. Ensuite, on peut directement appliquer la réduction 6.5. □

Si nous sommes dans la situation où nous avons une connaissance complète des états de l'ensemble d'entraînement ($D_n \in \mathsf{L}_{qu}^{cl}$), il est possible de choisir les deux sous-ensembles de classes de manière à maximiser la distance de trace entre les deux matrices densité qui représentent ces deux sous-ensembles. Dans ce cas-là, il est possible de construire l'arbre depuis la racine jusqu'aux feuilles en séparant à chaque fois l'ensemble de données en deux sous-ensembles qui maximisent la distance de trace. Nous pourrions aussi faire croître l'arbre depuis les feuilles jusqu'à la racine en choisissant au premier niveau de mettre ensemble les paires de classes qui sont les plus faciles à distinguer. En particulier, il existe une réduction appelée "*filter tree*" [33] qui réduit la classification multiclasse à la classification binaire (via la classification binaire pondérée et la réduction *costing* [195]). Cette réduction construit un classifieur multiclasse de type arbre binaire en commençant par les feuilles et garantit que l'erreur de ce classifieur est au plus $\epsilon \log k$, pour k le nombre de classes et ϵ l'erreur moyenne des classifieurs binaires générés. Le point fort de cette réduction est qu'elle offre une garantie similaire sur le regret (ce qui n'est pas le cas de l'algorithme arbre_binaire_entrainement présenté précédemment). Ainsi le regret du classifieur multiclasse est au pire de $r \log k$, pour r le regret moyen des classifieurs binaires générés.

6.2.3.4 Pretty Good Measurement

Si nous connaissons la description classique des points de données ($D_n \in \mathsf{L}_{qu}^{cl}$), il existe une stratégie générale de mesure appelée le "*Pretty Good Measurement*"[18] [107] qui permet de construire un classifieur qui, étant donné une seule copie de l'état inconnu

[18] Cette mesure est aussi parfois appelée "*square-root measurement*" dans la littérature à cause de la forme explicite de ce POVM.

$|\psi_?\rangle$, va prédire la classe de cet état avec une précision qui est bornée par la racine carrée de l'erreur du classifieur optimal.

Théorème 6.6 (Taux d'erreur du Pretty Good Measurement [18]). *Étant donné la description classique de k matrices densité $\rho_1, \ldots, \rho_k$, il est possible de construire un POVM, appelé* Pretty Good Measurement, *dont le taux d'erreur ϵ_{PGM} pour distinguer entre ces k matrices de densité, étant donné une seule copie d'un de ces matrices $\rho_?$, est au pire quadratiquement plus élevé que l'erreur ϵ_{opt} que ferait le POVM optimal. De façon équivalente, cela signifie que*

$$\epsilon_{opt} \leq \epsilon_{PGM} \leq \sqrt{\epsilon_{opt}} \tag{6.13}$$

Corollaire 6.7 (Borne sur le regret du Pretty Good Measurement). *Le regret du Pretty Good Measurement est bornée par :*

$$r_{PGM} \leq \sqrt{\epsilon_{opt}} - \epsilon_{opt} \tag{6.14}$$

Montanaro [146] a prouvé que l'erreur du Pretty Good Measurement est toujours plus petite que la stratégie de prédiction qui consiste à choisir la classe aléatoirement sans même mesurer l'état. Plus précisément, il a donné une borne supérieure sur l'erreur du Pretty Good Measurement qui dépend de la fidélité entre chaque paire d'états qui compose l'ensemble d'entraînement. Cette borne est :

$$\epsilon_{PGM} \leq 1 - \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sum_{j=1}^{n} Fid(|\psi_i\rangle, |\psi_j\rangle)} \tag{6.15}$$

À partir de la matrice de similarité S_n d'un ensemble de données quantiques D_n, il est possible de calculer explicitement cette borne. Pour cela, il suffit de remplacer $Fid(|\psi_i\rangle, |\psi_j\rangle)$ dans l'équation 6.15 par l'entrée de la matrice $S(i,j)$ correspondante. Cela implique que cette borne peut être explicitement estimée étant donné un nombre linéaire de copies de chaque état quantique de l'ensemble d'entraînement.

Théorème 6.7 (Borne sur l'erreur du Pretty Good Measurement). *Étant donné un ensemble d'entraînement $D_n \in \mathsf{L}_{qu}^{\otimes\Theta(n)}$, on peut obtenir* une borne supérieure sur l'erreur que le Pretty Good Measurement *réalise sur cet ensemble de données D_n.*

Démonstration. La preuve est directe, il suffit d'appliquer l'algorithme calcul_matrice_similarite et d'évaluer l'équation 6.15 en utilisant l'estimé de la fidélité entre les paires d'états à partir des entrées correspondantes de la matrice de similarité. □

Corollaire 6.8. *La* tâche d'estimer une borne supérieure sur l'erreur du Pretty Good Measurement *appartient à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(n)}$.

Montanaro a trouvé une autre borne sur l'erreur du Pretty Good Measurement 6.15 qui dépend directement des valeurs propres de la matrice de similarité S_n. Soit λ_i, la $i^{\text{ème}}$ valeur propre de la matrice de similarité, l'erreur du Pretty Good Measurement est bornée par dessus par :

$$\epsilon_{PGM} \leq 1 - \frac{1}{n}\left(\sum_{i=1}^{n}\sqrt{\lambda_i}\right)^2 \tag{6.16}$$

Cette borne peut aussi être calculée explicitement à partir de la matrice de similarité S_n. Il suffit pour cela de diagonaliser cette matrice pour en extraire les valeurs propres. En ce qui concerne une borne inférieure sur le Pretty Good Measurement, il existe une borne récente [147] due aussi à Montanaro, qui prouve que cette erreur est bornée par dessous par :

$$\epsilon_{PGM} \geq \sum_{i=1}^{n}\sum_{j=i}^{n} p_i p_j Fid(|\psi_i\rangle, |\psi_j\rangle), \tag{6.17}$$

où p_i et p_j sont les probabilités *a priori* des états $|\psi_i\rangle$ et $|\psi_j\rangle$. Dans la situation où tous les états sont équiprobables, il suffit de remplacer tous les probabilités par $\frac{1}{n}$ dans la formule 6.17. Là aussi cette borne inférieure peut être estimée directement à partir de la matrice de similarité S_n.

Intuitivement, ces bornes semblent indiquer que la fidélité entre paires d'états est une mesure suffisante pour savoir si les états de l'ensemble de données quantique peuvent être discriminés les uns des autres. Cette intuition est fausse, en effet Jozsa et Schlienz [124] ont prouvé qu'il existe des situations où la fidélité entre chaque paire d'états de l'ensemble de données D_n est faible (c'est-à-dire qu'il est aisé de différencier entre ces deux états) mais en même temps il est impossible de distinguer efficacement de manière globale un état de tous les autres états.

Pour résumer, il semble plus facile de borner l'erreur que réaliserait le Pretty Good Measurement que de construire explicitement celui-ci. En effet, nous pouvons borner

l'erreur étant donné un nombre linéaire de copies de chaque état de l'ensemble de données quantique D_n, alors que pour construire explicitement le POVM correspondant à cette mesure les techniques connues actuellement demandent de connaître explicitement la description classique des états (ce qui demande un nombre exponentiel de copies en utilisant la tomographie).

Question ouverte 6.5 (Quantité d'information nécessaire pour "apprendre" le Pretty Good Measurement). *Quel est le nombre minimum de copies t_{PGM} de chacun des états d'un ensemble de données quantique D_n nécessaire afin de pouvoir "entraîner" un POVM qui pourrait implémenter (exactement ou approximativement) le Pretty Good Measurement ?*

Sans rentrer dans les détails, les interrogations et arguments présentés lors de la discussion concernant l'oracle de Helstrom, comme l'existence (ou non) d'un circuit de taille polynomial implémentant le POVM ou le fait d'implémenter celui-ci exactement ou approximativement, s'appliquent aussi au Pretty Good Measurement[19].

6.3 Apprentissage non-supervisé quantique

Pour rappel, la différence fondamentale entre l'apprentissage supervisé et non-supervisé est que dans le deuxième cas, on ne connaît pas *a priori* les classes des différents points de l'ensemble de données. Dans le cadre de l'apprentissage non-supervisé, la fidélité va être une mesure pertinente dans le cadre de nombreuses tâches d'apprentissage, quelle soit utilisée comme mesure de similarité entre états quantiques pour la catégorisation, comme une propriété qu'on souhaite préserver en réduction de dimensionnalité ou comme critère de succès pour l'estimation de densité.

6.3.1 Catégorisation

La *catégorisation* vise à regrouper les états quantiques qui sont similaires dans la même catégorie et à placer les états dissemblables dans des catégories différentes.

[19] De la même manière que nous avons défini auparavant l'oracle de Helstrom, nous pourrions aussi définir deux versions de l'*oracle du Pretty Good Measurement*. La première version prend en entrée une description classique des matrices densité des classes, soit $\rho_1, \ldots, \rho_k$, alors que la deuxième apprend à partir d'un nombre fini s de copies de chaque état, pour $s \geq t_{PGM}$. L'oracle produit en sortie une implémentation efficace du Pretty Good Measurement (par exemple sous forme de circuit).

Tâche d'apprentissage quantique 6.5 (Catégorisation).
Entrée *: un ensemble de données* $D_n = \{|\psi_1\rangle, \ldots, |\psi_n\rangle\}$, *où* $|\psi_i\rangle \in \mathbb{C}^{2^d}$, *et* k *le nombre de catégories.*
Sortie *: un ensemble de* k *catégories tel que chaque état quantique* $|\psi_i\rangle$ *soit assigné à au moins une catégorie* $y \in \{1, \ldots, k\}$.
But *: (1) que les états quantiques dans une même catégorie partagent une fidélité élevée (*intra-similarité*) et que (2) les états quantiques se trouvant dans différentes catégories soient dissemblables (*inter-dissimilarité*).*

Une approche possible au problème de catégorisation consiste à estimer la similarité entre deux états quantiques en terme de fidélité directement en faisant une mesure conjointe pour chaque paire d'états. Supposons, par exemple que nous recevions un nombre de copies de chaque état qui est de $\Theta(en)$, pour n le nombre d'états dans l'ensemble de données et e une constante qui contrôle la précision de l'estimation de la fidélité. Dans ce cas, il suffirait d'appeler l'algorithme calcul_matrice_similarite 6.1.4 pour calculer la matrice de similarité S_n. À partir de cette information, on peut utiliser un algorithme classique, tel que k-médianes (voir section 5.6), pour résoudre le problème de catégorisation. L'algorithme suivant formalise cette méthode.

Algorithme 22 categorisation_quantique($D_n \in \mathsf{L}_{qu}^{\otimes\Theta(n)}, k$)

Appeler calcul_matrice_similarite(D_n) afin d'obtenir la matrice de similarité S_n
Appeler l'algorithme des k-médianes (classique ou quantisé) en utilisant S_n comme entrée
Retourner les catégories générées par l'algorithme des k-médianes ainsi que leurs centres

Théorème 6.8 (Borne supérieure de la catégorisation quantique). *Il est possible de catégoriser un ensemble* D_n *composé de* n *états quantiques si on dispose de* $\Theta(n)$ *copies de chacun de ces états.*

Démonstration. L'algorithme calcul_matrice_similarite requiert un nombre de copies de chaque état de $\Theta(en)$, où e est une constante qui contrôle la précision de l'estimé de la fidélité. Cette constante est fonction de la fidélité moyenne entre chaque paire d'états. Le reste de l'algorithme categorisation_quantique étant classique, le coût global de la

catégorisation quantique sera de $\Theta(n)$ copies par état, si on ne prend pas en compte la constante e lors de l'analyse. □

Corollaire 6.9. *La* tâche de catégorisation quantique *appartient à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(n)}$.

Dans [8], nous avons testé l'algorithme categorisation_quantique sur un ensemble de données simulées. Chaque catégorie était centrée autour d'un état pur sur 13 qubits (ce qui correspond à un espace de Hilbert de dimension $2^{13} = 8192$) généré aléatoirement selon la mesure de Haar. Les autres états de la catégorie sont générés aléatoirement eux aussi et sont filtrés de manière à ne pas être trop éloignés au sens de la fidélité du centre de la catégorie. En faisant varier le *seuil de fidélité* qui représente la distance maximum entre un membre de la catégorie et son centre, et en faisant la moyenne sur plusieurs ensemble de données générés aléatoirement, nous avons observé comment la performance de l'algorithme de catégorisation se dégrade lorsque le seuil baisse. Nous avons pu voir expérimentalement, que pour cette situation de catégorisation particulière, l'algorithme arrive à retrouver les catégories initiales avec une précision presque parfaite même lorsque le seuil est aussi bas que 0.6 (c'est-à-dire qu'un état peut dans le pire cas avoir une fidélité avec son centre de catégorie qui est aussi bas que 0.6). Bien sûr, ceci ne constitue qu'une expérimentation sur un ensemble de données "jouet" et il est important de développer de nouveaux ensembles de données quantique sur lesquels tester cet algorithme de catégorisation (cf. section 6.4.3).

D'autres stratégies de catégorisation peuvent aussi être développées qui sont encore plus quantiques par nature. Par exemple, nous pourrions adapter un algorithme classique de catégorisation, tel qu'un *algorithme agglomératif* qui construirait les catégories autour de *germes quantiques* de manière adaptive. Cet algorithme commencerait par sacrifier une partie de l'ensemble de données afin de déterminer les états quantiques qui sont les plus dissemblables en terme de fidélité afin de les utiliser ensuite comme germes. Durant la seconde phase, chaque état serait ensuite comparé aux germes grâce au C-Swap test et aggloméré autour du germe le plus similaire. Ce type d'algorithme pourrait être moins demandant en terme de copies que l'algorithme categorisation_quantique ce qui conduit à formuler la question ouverte suivante.

Question ouverte 6.6 (Borne inférieure de la catégorisation quantique). *Est-il possible*

de catégoriser un ensemble de données composé de n états quantiques en utilisant moins de $\Theta(n)$ copies de chaque état ?

6.3.2 Réduction de dimensionnalité quantique

La réduction de dimensionnalité est une tâche d'apprentissage qui est très proche en esprit de la *compression quantique.*

Tâche d'apprentissage quantique 6.6 (Réduction de dimensionnalité). ***Entrée*** *: $D_n = \{|\psi_1\rangle, \ldots, |\psi_n\rangle\}$, un ensemble de données quantique, où $|\psi_i\rangle \in \mathbb{C}^{2^d}$.*
Sortie *: un superopérateur agissant comme une fonction de compression f qui peut transformer n'importe quel état de l'ensemble de données $|\psi\rangle \rightarrow |\psi'\rangle$.*
But *: pour être intéressant le superopérateur f doit respecter deux propriétés : (1) $|\psi'\rangle \in \mathbb{C}^{2^{d'}}$ pour $d' \ll d$ (*réduction de dimensionnalité*) et (2) $Fid(|\psi'_i\rangle, |\psi'_j\rangle) \approx Fid(|\psi_i\rangle, |\psi_j\rangle)$ pour n'importe quelle paire $(|\psi_i\rangle, |\psi_j\rangle) \in D_n$ (*préservation de l'information*).*

Un *compresseur quantique* est un superopérateur qui transforme un état quantique qui vit en haute dimension (par exemple sur d qubits) en un état quantique défini sur une plus faible dimension (par exemple d' qubits pour $d' \ll d$). Une *fonction de décompression quantique* est un superopérateur qui réalise la transformation inverse et renvoie l'état vers sa configuration initiale. Le succès de ce processus de compression et décompression peut se mesurer en fonction de la fidélité entre l'état original ρ_{ori} et sa reconstruction ρ_{rec} après l'étape de décompression, ce qui correspond à $Fid(\rho_{ori}, \rho_{rec})$. Si cette procédure s'effectue parfaitement et sans perte alors $Fid(\rho_{ori}, \rho_{rec}) = 1$.

Le taux optimal auquel un état quantique $\rho = \sum_{i=1}^{n} p_i |\psi_i\rangle\langle\psi_i|$ peut être compressé sans perte de manière asymptotique est donné par l'*entropie de von Neumann* de l'état, qui peut être vu comme l'analogue quantique de l'entropie de Shannon.

Définition 6.19 (Entropie de von Neumann). *L'*entropie de von Neumann *d'un état ρ est égale à $H(\rho) = -Tr(\rho \log \rho)$. Elle est équivalente à l'entropie de Shannon mesurée sur les valeurs propres de ρ, soit $H(\rho) = -\sum_{i=1}^{d} \lambda_i \log \lambda_i$, où λ_i est la $i^{ème}$ valeur propre de ρ.*

Ce taux peut être atteint asymptotiquement par une méthode de compression appelée *compression de Schumacher* [167]. Bennett a proposé une approche constructive [21] pour

implémenter la compression de Schumacher qui consiste à appliquer une opération unitaire qui emmène ρ vers une base où l'état est diagonal, suivi d'une autre opération unitaire qui permute les vecteurs propres afin qu'ils soient triés par ordre décroissant par rapport à leurs valeurs propres. Comme ces deux opérations sont unitaires, la décompression peut se faire très naturellement en appliquant les opérations inverses qui sont elles aussi unitaires. L'aspect "unitaire" de cette compression offre aussi le bénéfice additionnel de garantir que si cette transformation est appliquée à deux états quantiques différents, elle va aussi préserver leur fidélité, satisfaisant ainsi la condition de préservation d'information de la définition 6.6.

Le seul inconvénient de la méthode de Bennett est qu'elle requiert que nous connaissions à l'avance les vecteurs propres et valeurs propres de la matrice densité qu'on souhaite compresser. Cependant, dans le cas où la description de la source est inconnue, une technique de compression existe [123] permettant de compresser ρ asymptotiquement vers son entropie de von Neumann. Comme ce résultat concerne le cas asymptotique, il énonce seulement ce qui est possible lorsque le nombre de copies de ρ tend vers l'infini mais ne dit rien quant au cas fini (qui est le cas qui nous intéresse pour la réduction de dimensionnalité). Cette procédure apprend en fait les vecteurs propres et les valeurs propres de l'état ρ en même temps qu'elle fait de la compression, ce qui correspond à la tâche d'estimation de densité (voir section 6.3.3). Une question ouverte importante est de déterminer si la réduction de dimensionnalité peut se faire avec un nombre polynomial de copies de chaque état tout comme la catégorisation.

Question ouverte 6.7 (Quantité d'information nécessaire pour la réduction de dimensionnalité). *Est-il possible d'apprendre un superopérateur qui permettrait de réaliser la réduction de dimensionnalité sur ensemble de données quantique D_n étant donné un nombre polynomial de copies de chaque état ?*

Il semble plausible que la réponse à cette question soit négative et que la tâche de réduction de dimensionnalité requiert autant d'états quantiques que la tâche d'estimation de densité (soit un nombre exponentiel par rapport au nombre de qubits sur lesquels les états quantiques sont définis). En effet, Blume-Kohout[20] a mis en avant la tâche

[20]Voir `http://www.crm.umontreal.ca/TC4/pdf/blumekohout.pdf` par exemple

de compression d'états quantiques comme étant *la tâche opérationnelle qui de part son essence requiert d'avoir une connaissance complète des états quantiques* à compresser.

6.3.3 Estimation de densité

L'*estimation de densité quantique* est directement reliée au concept de *tomographie quantique* [154], qui vise à reconstruire une description classique d'un état quantique aussi précise que possible à partir d'un nombre fini de copies de cet état (voir aussi le chapitre 4.6).

Tâche d'apprentissage quantique 6.7 (Estimation de densité). ***Entrée*** *: $D_n = \{(|\psi_1\rangle, p_1), \ldots, (|\psi_n\rangle, p_n)\}$, un ensemble de données quantique, où $|\psi_i\rangle \in \mathbb{C}^{2^d}$, $0 \leq p_i \leq 1$ et $\sum_{i=1}^{n} p_i = 1$. (Si jamais les états quantiques sont équiprobables alors on a pour tout état $|\psi_i\rangle$ que sa probabilité* a priori *$p_i = \frac{1}{n}$.)*
Sortie *: une description classique de la matrice de densité $\rho = \sum_{i=1}^{n} p_i |\psi_i\rangle\langle\psi_i|$.*
But *: maximiser la fidélité entre la reconstruction ρ_{guess} et la vraie matrice de densité ρ_{true}, avec dans le cas idéal $Fid(\rho_{guess}, \rho_{true}) = 1$.*

Supposons par exemple que l'ensemble de données D_n est composé d'un seul état pur $|\psi_{true}\rangle$. La fidélité moyenne espérée qui peut être atteinte entre la reconstruction de l'état, que nous nommons $|\psi_{guess}\rangle$, et le vrai état $|\psi_{true}\rangle$ est fonction de la dimension de l'état et du nombre de copies dont nous disposons. Dans le cas où nous n'avons aucune connaissance *a priori* sur l'état, la formule exacte [108] est

$$Fid_{opt}(|\psi_{guess}\rangle, |\psi_{true}\rangle) = \frac{s+1}{s+2^d} \tag{6.18}$$

où s est le nombre de copies de l'état dont on dispose et 2^d la dimension de l'espace de Hilbert dans lequel lequel l'état vit.

Lemme 6.3 (Nombre de copies requises pour une tomographie précise). *Afin d'atteindre une fidélité raisonnable de la reconstruction, un nombre exponentiel de copies par rapport à d, le nombre de qubits où l'état est défini, est nécessaire.*

Démonstration. Supposons en effet que l'ensemble de données D_n contienne un seul état pur $|\psi_{true}\rangle$ dont on a aucune connaissance préalable. De part la formule 6.18, il faut un

nombre de copies exponentielle par rapport d le nombre de qubits, car la dimension de l'espace de Hilbert est elle-même exponentielle en d. Ainsi si nous disposons d'un nombre de copies qui est moins que 2^d, on observera $Fid(|\psi_{guess}\rangle, |\psi_{true}\rangle) \ll 1$. □

Corollaire 6.10. *La* tâche d'estimation de densité *appartient à la classe d'apprentissage* $\mathsf{L}_{qu}^{\otimes\Theta(2^d)}$.

En pratique, si jamais la dimension d est petite et que s le nombre de copies est large, l'approche tomographique permettra de générer une description classique des états de qualité raisonnable. Cependant dès que d devient de taille moyenne ou large, le coût de cette méthode devient en général trop prohibitif[21]. La situation empire encore lorsque l'état à tomographier est un mélange statistique (au lieu d'être un état pur) car le nombre d'entrées de la matrice densité à estimer est de l'ordre de $O(4^d)$ (du fait que la matrice densité soit de taille 2^d par 2^d).

Réduction 6.6 (Réduction de la classification binaire à l'estimation de densité). *Étant donné un oracle qui permet de résoudre la tâche d'estimation de densité, il est possible de l'utiliser pour* réduire la tâche de classification binaire à l'estimation de densité.

Démonstration. L'idée principale de la réduction est de tout simplement séparer l'ensemble de données quantique D_n en deux sous-ensembles disjoints (un pour la classe positive et un pour la classe négative). On appelle ensuite l'oracle sur chacun des sous-ensembles de manière à obtenir la description des matrices densité de ρ_- et ρ_+. À partir de ces descriptions, nous pouvons calculer analytiquement la mesure de Helstrom qui permet de distinguer de manière optimale les deux classes ρ_- et ρ_+. Le coût de la réduction est de faire deux appels à l'oracle permettant de faire l'estimation de densité, soit globalement un coût de $\Theta(2^d)$ copies pour chaque état quantique de D_n. □

Réduction 6.7 (Réduction de la catégorisation à l'estimation de densité). *Étant donné un oracle qui permet de résoudre la tâche d'estimation de densité, il est possible de l'utiliser pour* réduire la tâche de catégorisation à l'estimation de densité.

[21] Une exception pourrait être de disposer *a priori* d'une connaissance précise sur la forme que peut prendre cet état. Dans ce cas-là, il est possible qu'il puisse être ciblé et le reconstruit précisément avec beaucoup moins de copies que nécessaire dans le cas général.

Démonstration. L'oracle d'estimation de densité peut être appelé n fois pour produire la description classique de chacun des états $|\psi_i\rangle \in D_n$. À partir de ces descriptions, nous pouvons calculer analytiquement et de manière exacte la fidélité entre chaque paire d'états de l'ensemble de données D_n. Ensuite, nous pouvons utiliser un algorithme de catégorisation tel que l'algorithme des k-médianes (classique) en qui prend en entrée cette information (comme dans la section 6.3.1). Le coût de cette réduction sera de $O(2^d)$ copies pour chaque état de D_n. □

Réduction 6.8 (Réduction de la réduction de dimensionnalité à l'estimation de densité). *Étant donné un oracle qui permet de résoudre la tâche d'estimation de densité, il est possible pour* réduire la tâche de réduction de dimensionnalité à l'estimation de densité.

Démonstration. La preuve de cette réduction est relativement directe. Il suffit de commencer par appeler l'oracle permettant de résoudre l'estimation de densité sur D_n pour obtenir une description classique de $\rho = \sum_{i=1}^{n} p_i |\psi_i\rangle\langle\psi_i|$. À partir de cette description de ρ, nous pouvons ensuite appliquer directement dessus la compression de Schumacher. Cette réduction fait un seul appel à l'oracle permettant de résoudre l'estimation de densité et demandera donc un nombre total de copies de l'ordre de $\Theta(2^d)$ pour chaque état quantique. □

Pour résumer, l'estimation de densité est une tâche très coûteuse en terme de ressources car elle demande un nombre de copies de chaque état quantique de $\Theta(2^d)$ dans le cas général. Il s'agit pourtant d'une primitive d'apprentissage très générique car elle peut servir à la réduction à d'autres tâches telles que la classification binaire (réduction 6.6), la catégorisation (réduction 6.7) ou encore la réduction de dimensionnalité (réduction 6.8). Autrement dit, comme elle permet d'obtenir une *connaissance complète des états quantiques*, elle peut être vu comme la tâche d'apprentissage la plus générale. Ceci reste vrai dans l'apprentissage machine classique, où étant donné un bon estimateur de densité, il est généralement possible de l'utiliser pour résoudre à peu près toutes les autres tâches d'apprentissage, quelles soit supervisées ou non-supervisées.

6.4 Discussion et perspectives futures

Le tableau 6.1 récapitule les coûts d'entraînement/apprentissage et de classification des différentes tâches et réductions d'apprentissage vues dans ce chapitre. Dans le cas de l'apprentissage supervisé, la classification binaire est la principale primitive d'apprentissage car la version pondérée et multiclasse de la classification peuvent s'y réduire via l'oracle de Helstrom. En apprentissage non-supervisé, l'estimation de densité semble être la tâche d'apprentissage la plus générique mais en contrepartie elle requiert un nombre de copies de chaque état de l'ensemble de données qui est exponentiel dans le nombre de qubits d sur lesquels ils sont définis (classe d'apprentissage $\mathsf{L}_{qu}^{\otimes\Theta(2^d)}$). La catégorisation peut par contre être réalisée avec un nombre de copies de chaque état de l'ensemble de données quantique D_n qui est linéaire en n (classe d'apprentissage $\mathsf{L}_{qu}^{\otimes\Theta(n)}$). Il reste maintenant à trouver des applications en théorie de la détection et de l'estimation quantique où la catégorisation pourra être utile.

Tâche d'apprentissage	Coût d'entraînement	Coût de classification
Classification binaire	$\Theta(t_{bin})$	$\Theta(1)$
Estimation de la probabilité d'appartenance à une classe (réduction par oracle de Helstrom)	$\Theta(t_{bin})$	$\Theta(1)$
Classification binaire pondérée		
(réduction par oracle de Helstrom)	$\Theta(t_{bin})$	$\Theta(1)$
(costing réduction)	$\Theta(Tt_{bin})$	$\Theta(T)$
Classification multiclasse		
(identification par C-SWAP test)	$\Theta(c)$	$\Theta(cn)$
(réduction du type une-contre-tous)	$\Theta(kt_{bin})$.	$\Theta(k)$
(réduction du type arbre binaire)	$\Theta(t_{bin}\log k)$	$\Theta(\log k)$
(Pretty Good Measurement)	inconnu	$\Theta(1)$
(Borne sur l'erreur du PGM)	$\Theta(n)$	non applicable
Catégorisation	$\Theta(n)$	non applicable
Réduction de dimensionalité	$O(2^d)$	non applicable
Estimation de densité	$\Theta(2^d)$	non applicable

TAB. 6.1 – Tableau résumant les coûts d'entraînement et de classification des différentes tâches quantiques d'apprentissage présentées dans ce chapitre.

6.4.1 Algorithmes quantiques de classification

En pratique, l'oracle de Helstrom va être implémenté par un algorithme d'apprentissage quantique, qui à partir d'un nombre fini de copies de chaque état de l'ensemble d'entraînement, va produire en sortie un POVM f jouant le rôle d'un classifieur binaire. Contrairement à l'oracle de Helstrom, cet algorithme n'a pas besoin d'être optimal au niveau de l'erreur de classification tant qu'il a une précision non-triviale, qui est meilleure de simplement faire une prédiction aléatoire sur la classe d'un état quantique inconnu. Même dans ce cas-là, la plupart des réductions présentées dans ce chapitre vont fonctionner bien que l'erreur d'entraînement globale du classifieur généré sera sûrement amoindrie à cause de la non-optimalité du POVM construit. Concevoir un algorithme d'apprentissage jouant le rôle de l'oracle de Helstrom permettra aussi d'estimer explicitement le nombre minimum de copies t_{bin} de chaque état de l'ensemble d'entraînement qui est nécessaire pour mener à bien la classification binaire.

Conjecture 6.1 (Possibilité de réaliser la classification binaire avec un nombre polynomial de copies). *Il est possible d'apprendre un POVM f qui jouera le rôle d'un classifieur binaire précis à partir d'un ensemble de données quantique D_n qui contient un nombre polynomial en n de copies de chaque état (soit $D_n \in \mathsf{L}_{qu}^{\otimes\Theta(poly(n))}$).*

Parmi les pistes possibles pour construire explicitement l'algorithme d'apprentissage se trouve le développement de variantes quantiques de ID3 [160] et de AdaBoost [89]. Ces variantes quantiques exploiteraient respectivement, la relation entre l'entropie de Shannon et l'entropie de von Neumann et la similarité entre les notions de classifieur faible et de mesure faible.

6.4.2 Généralisation

Tout l'essence de l'apprentissage machine est d'apprendre à partir d'observations sur des expériences passées afin de pouvoir généraliser à des situations non rencontrées auparavant. Pour l'instant dans le monde quantique, nous nous sommes surtout concentrés sur la tâche de classifier correctement des états de l'ensemble d'entraînement D_n mais nous n'avons pas discuté comment cette approche peut généraliser sur des états quantiques non observés auparavant. Une manière naturelle de dire qu'un classifieur f, représenté sous forme de POVM, généralise est s'il est capable de reconnaître la classe d'un état

quantique qui est proche d'un état de l'ensemble d'entraînement mais sans être identique exactement. La distance euclidienne entre deux états purs est une mesure de distance entre états quantiques qui peut être définie de la manière suivante.

Définition 6.20 (Distance euclidienne entre états purs [30]). *La* distance euclidienne entre deux états purs $|\psi\rangle = \sum_{i=1}^{d} \alpha_i |i\rangle$ *et* $|\phi\rangle = \sum_{i=1}^{d} \beta_i |i\rangle$ *est définie comme* $Dist_{L2}(|\psi\rangle, |\phi\rangle) = \sqrt{\sum_{i=1}^{d} |\alpha_i - \beta_i|^2}$.

Bernstein et Vazirani ont prouvé que si deux états purs $|\psi\rangle$ et $|\phi\rangle$ de même dimension sont à une distance euclidienne de ϵ, la même mesure fait sur les deux états génère des échantillons provenant de deux distributions ayant pour distance totale de variation entre les deux distributions au plus 4ϵ. Autrement dit, si deux états sont proches en terme de leur distance euclidienne cela donne une indication qu'un POVM f jouant le rôle de classifieur fera avec une bonne probabilité la même prédiction concernant la classe des deux états.

Parmi les travaux futurs dans ce modèle de l'apprentissage machine sur de l'information quantique se trouvent la formalisation de la notion d'erreur de test et d'erreur de généralisation, ainsi que l'étude des différents modèles de bruit classique et quantique (voir par exemple la section 8.3 de [152] pour différentes formes de bruit quantique) et comment ils affectent la robustesse des algorithmes quantiques d'apprentissage.

6.4.3 Mise au point d'ensembles de données quantiques

L'apprentissage machine est un domaine où il est important de valider expérimentalement la performance d'un algorithme et de la comparer à celle d'autres algorithmes existants. Classiquement, de nombreux répertoires d'ensembles de données sont publiquement disponibles dont le répertoire de l'université de Californie à Irvine[22] (abrégé en anglais *UCI repository*) et la base de données de reconnaissance de caractères MNIST[23]. Quantiquement, une fois que plusieurs algorithmes d'apprentissage auront été mis au point, il est important de tester expérimentalement l'erreur obtenue par ces algorithmes d'apprentissage sur des ensembles de données quantique représentant des situations réalistes qu'un expérimentateur pourrait rencontrer dans son laboratoire. Je

[22] `http://archive.ics.uci.edu/ml/`
[23] `http://yann.lecun.com/exdb/mnist/`

ne suggère pas de créer physiquement ces ensembles de données mais plutôt de mettre en libre accès auprès de la communauté leurs descriptions classiques afin que n'importe qui le souhaitant puisse les utiliser directement à partir de son langage de programmation ou son simulateur classique préféré. Un exemple de deux classes possibles pourrait être par exemple états intriqués (classe positive) et états séparables (classe négative). De plus, beaucoup de scénarios rencontrés en cryptographie quantique peuvent se reformuler comme un problème de classification où l'espion essaye de maximiser sa probabilité de deviner correctement la classe de l'état qu'il a intercepté.

CHAPITRE 7

CONCLUSION

Quelles sont les connections possibles entre la physique, l'information et l'apprentissage ? Cette question résume bien la problématique générale de l'*apprentissage quantique*. Tout comme l'informatique quantique, le domaine est encore jeune mais il grandit et mûrit au fur et à mesure que de nouvelles interactions entre l'informatique quantique et l'apprentissage machine sont explorées. D'un côté, l'informatique quantique offre un nouveau paradigme de calcul qui permet d'accélérer certains algorithmes d'apprentissage. De l'autre côté, l'apprentissage machine permet de raisonner sur certains problèmes quantiques comme étant des tâches d'apprentissage et donc d'amener des idées issues de l'apprentissage pour aider à les résoudre. Cette thèse a fait progresser l'apprentissage quantique à travers trois contributions.

7.1 Première contribution : tour d'horizon de l'apprentissage quantique

La première contribution est d'avoir *définie l'apprentissage quantique* et *donnée un tour d'horizon du domaine* (chapitre 4). Bien que l'informatique quantique et l'apprentissage machine semblent être *a priori* deux domaines très différents, nous avons vu dans cette thèse qu'ils se sont rencontrés et ont interagi de multiples fois par le passé. L'apprentissage quantique est le domaine qui est né des différentes rencontres entre l'informatique quantique et l'apprentissage machine. Les travaux en apprentissage quantique présentés dans l'état de l'art sont :

- les travaux en *théorie calculatoire de l'apprentissage* (section 4.2) comparant l'apprentissage quantique et classique dans le modèle PAC et le modèle d'apprentissage exact à partir de requêtes.
- les *variantes quantiques d'algorithmes d'apprentissage* (section 4.3) dont les réseaux de neurones quantiques, la quantisation de l'entraînement des machines à vecteurs de support et la version quantisée d'un algorithme d'apprentissage par renforcement.
- un *algorithme classique de catégorisation puisant son inspiration dans la mécanique*

quantique (section 4.4).

- des *bornes en complexité de la communication quantique s'appuyant sur des notions d'apprentissage machine* (section 4.5).
- l'*estimation de systèmes et de processus quantiques* (section 4.6) utilisant des techniques standards d'estimation de densité.
- un *calcul bayésien pour mettre à jour notre connaissances des matrices densité* (section 4.7).
- des *cryptosystèmes basés sur des problèmes d'apprentissage considérés difficiles même pour un ordinateur quantique* (section 4.8).
- *apprendre à généraliser sur des POVMs* (section 4.9).

7.2 Deuxième contribution : quantisation d'algorithmes d'apprentissage non-supervisé

La deuxième contribution de cette thèse concerne l'exploration d'une voie de recherche très peu considérée auparavant, celle de la *quantisation d'algorithmes d'apprentissage non-supervisé* (chapitre 5). La quantisation d'un algorithme d'apprentissage classique consiste à remplacer certaines parties de l'algorithme par des sous-routines quantiques afin d'obtenir une accélération ou de sauver sur le coût de communication dans le cas d'une situation d'apprentissage distribuée. Les principales avancées apportées concernant cette direction de recherche sont :

1. **Bris de la boîte noire** (section 5.3.3) : un apport important de ma recherche est de s'être détaché du paradigme de la boîte noire, qui est usuellement considéré en informatique quantique. À la place, j'ai décrit une *construction explicite permettant à partir de la description classique d'un ensemble de données de produire un circuit quantique qui peut être interrogé en superposition* à l'intérieur d'algorithmes d'apprentissage. Le jour où un ordinateur quantique de taille raisonnable sera physiquement disponible, cette "recette" pourra être utilisée pour produire un circuit quantique encodant un ensemble de données classique.
2. **Algorithmes quantiques d'apprentissage non-supervisé** (sections 5.5, 5.6 et 5.7) : en utilisant des variantes de l'algorithme de Grover, nous avons pu *quantisé les algorithmes de catégorisation divisive, des k-médianes (version standard et*

distribuée), de construction d'un graphe de voisinage, de détection d'anomalies et d'initialisation "intelligente" des centres des catégories. Pour toutes ces versions quantisées, nous avons obtenu des gains significatifs par rapport aux contreparties classiques. De plus, les sous-routines et outils quantiques développés sont suffisamment génériques pour être utilisés afin de quantiser d'autres algorithmes d'apprentissage (comme les méthodes de voisinage en apprentissage supervisé).

Limites et perspectives futures (section 5.8) : comme toutes les sous-routines quantiques développées dans cette thèse sont basées sur des variantes de l'algorithme de Grover, le gain obtenu par rapport à la version classique est au mieux quadratique. Afin de dépasser la "barrière de Grover", il est nécessaire de développer des algorithmes d'apprentissage se basant sur d'autres techniques algorithmiques telles que les marches aléatoires et les chaînes de Markov quantiques, ou encore d'adopter un paradigme de calcul différent comme celui du calcul adiabatique et du modèle basé sur la mesure. De plus, comme la plupart des algorithmes quantisés présentés dans cette thèse sont des algorithmes de catégorisation, il est naturel de vouloir étendre la même approche de quantisation à d'autres tâches d'apprentissage non-supervisé comme la réduction de dimensionnalité ou l'estimation de densité. Finalement, la mise au point de bornes inférieures permettrait de caractériser la difficulté intrinsèque des différentes situations d'apprentissage.

7.3 Troisième contribution : apprentissage machine dans un monde quantique

La troisième contribution de cette thèse (chapitre 6) est le développement de l'*analogue de l'apprentissage machine dans un monde où l'ensemble de données est composé d'états quantiques*, et non plus d'observations classiques sur des objets classiques. Ce changement de situation d'apprentissage a un impact important sur le processus d'apprentissage et ses limitations. Les principaux avancements concernant cette voie de recherche sont :

1. **Apprentissage machine sur un ensemble de données quantique** (section 6.1) : en reformulant certains problèmes de la théorie quantique de la détection et de l'estimation en tâches d'apprentissage, nous avons vu qu'il devient *possible d'utiliser des notions d'apprentissage machine, telles que les réductions d'appren-*

tissage, pour aider à résoudre ces problèmes. Les concepts d'*ensemble de données quantique*, de *classe d'apprentissage*, de *réduction d'apprentissage* et de *matrice de similarité* ont été définis car ils constituent les éléments de base de l'apprentissage sur de l'information quantique.

2. **Classification quantique** (section 6.2) : en apprentissage supervisé quantique, la *classification binaire est la principale primitive d'apprentissage.* En effet, étant donné l'accès à un oracle de Helstrom qui permet de résoudre cette tâche, nous pouvons aussi résoudre efficacement les *versions pondérée et multiclasse de la classification.* Différentes réductions, ayant chacunes des coûts d'entraînement et de classification différents, ont été définis.
3. **Apprentissage non-supervisé quantique** (section 6.3) : en apprentissage non-supervisé, la *fidélité semble être la mesure quantique clé* pour évaluer le succès des *versions quantiques de la catégorisation, la réduction de dimensionnalité et l'estimation de densité.* La tâche de catégorisation peut être réalisée efficacement avec un nombre linéaire de copies de chaque état de l'ensemble d'entraînement alors que l'estimation de densité, bien qu'étant une tâche d'apprentissage plus générique (car elle peut se réduire à la plupart des autres tâches), requiert un nombre exponentiel de copies des états quantiques.

Limites et perspectives futures (section 6.4) : l'oracle de Helstrom est un construction abstraite qui en pratique sera implémentée par un algorithme d'apprentissage. Une question fondamentale est donc de construire des d'algorithmes d'apprentissage performants (en terme de la quantité d'information consommée et du temps de calcul) pour la tâche de classification binaire qui peuvent jouer concrètement le rôle de l'oracle de Helstrom. Ensuite, ces algorithmes vont être évalués sur des ensembles de données quantique standard qui eux aussi restent à créer. D'autres travaux futurs concernent l'étude et la formalisation de l'erreur de généralisation et du bruit dans ce nouveau modèle, ainsi que le développement de nouvelles réductions entre des tâches d'apprentissage quantiques et certaines primitives de la théorie de l'information quantique.

BIBLIOGRAPHIE

[1] Aaronson, S.: *The learnability of quantum states.* Proceedings of the Royal Society A, 463(2088):3089–3114, 2007.

[2] Aharonov, D., W. van Dam, J. Kempe, J. Landau, S. Lloyd et O. Regev: *Adiabatic quantum computation is equivalent to standard quantum computation.* Dans *Proceedings of Foundations Of Computer Science (FOCS'04)*, pages 42–51, 2004.

[3] Aizerman, A., E. Braverman et L. Rozonoer: *Theoretical foundations of the potential function method in pattern recognition learning.* Automation and Remote Control, 25:821–837, 1964.

[4] Ambainis, A.: *Quantum walks and their algorithmic applications.* International Journal of Quantum Information, 1(0403120):507–518, 2003.

[5] Aïmeur, E., G. Brassard, H. Dufort et S. Gambs: *CLARISSE : A machine learning tool to initialize student models.* Dans *Proceedings of the 6th International Conference on Intelligent Tutoring Systems (ITS'02)*, pages 718–728, 2002.

[6] Aïmeur, E., G. Brassard et S. Gambs: *Quantum algorithms for unsupervised learning.* En préparation.

[7] Aïmeur, E., G. Brassard et S. Gambs: *Quantum learning tasks.* En préparation.

[8] Aïmeur, E., G. Brassard et S. Gambs: *Machine learning in a quantum world.* Dans *Proceedings of the 19th Canadian Conference on Artificial Intelligence (Canadian AI'06)*, pages 433–444, 2006.

[9] Aïmeur, E., G. Brassard et S. Gambs: *Quantum clustering algorithms.* Dans *Proceedings of the 24th Annual International Conference of Machine Learning (ICML'07)*, pages 1–8, 2007.

[10] Aïmeur, E., G. Brassard, S. Gambs et B. Kégl: *Privacy-preserving boosting.* Dans *Proceedings of Workshop on Privacy and Security Issues in Data Mining, in conjunction with the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, pages 51–69, 2004.

[11] Aïmeur, E. et S. Gambs: *Data mining and privacy.* Encyclopedia of Data Warehousing and Mining (2nd edition), 2008.

[12] Angiulli, F. et C. Pizzuti: *Fast outlier detection in high dimensional spaces.* Dans *Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'02)*, pages 15–26, 2002.

[13] Angluin, D.: *Queries and concept learning.* Machine Learning, 2:319–342, 1988.

[14] Anguita, D., S. Ridella, F. Riviecco et R. Zunino: *Quantum optimization for training support vector machines.* Neurals Networks, 16(1):763–770, 2003.

[15] Atıcı, A.: *Advances in Quantum Computational Learning Theory.* Thèse de doctorat, Columbia University, 2006. Disponible à `http://www.alpatici.net/dissertation.pdf`.

[16] Bang, J., J. Lim, M.S. Kim et J. Lee: *Quantum learning machine.* Disponible sur `arXiv:0803.2976v2[quant-ph]`.

[17] Barenco, A., A. Berthiaume A., D. Deutsch D., A. Ekert, R. Jozsa R. et C. Macchiavello: *Stabilisation of quantum computations by symmetrisation.* SIAM Journal of Computing, 26(5):1541–1557, 1997.

[18] Barnum, H. et E. Knill: *Reversing quantum dynamics with near-optimal quantum and classical fidelity.* Journal of Mathematical Physics, 43(5):2097–2106, 2002.

[19] Bell, J.S.: *On the Einstein-Podolsky-Rosen paradox.* Physics, 1(3):195–200, 1964.

[20] Bennett, C.H.: *Logical reversibility of computation.* IBM Journal of Research Development, 17:525–532, 1973.

[21] Bennett, C.H.: *Quantum information and computation.* Physics Today, pages 24–30, 1995.

[22] Bennett, C.H., E. Bernstein, G. Brassard et U. Vazirani: *Strengths and weaknesses of quantum computing.* SIAM Journal of Computing, 26(5):1510–1523, 1997.

[23] Bennett, C.H. et G. Brassard: *Quantum cryptography: Public key distribution and coin tossing.* Dans *Proceedings of the IEEE Conference on Computers, Systems and Signal Processing, Bangalore*, pages 175–179, 1984.

[24] Bennett, C.H., G. Brassard, C. Crépeau, R. Jozsa, A. Peres et W. Wootters: *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels.* Physical Review Letters, 70:1895–1899, 1993.

[25] Bennett, C.H., G. Brassard et N.D. Mermin: *Quantum cryptography without Bell's theorem.* Physical Review Letters, 68(5):557–559, 1992.

[26] Bennett, C.H. et S. Wiesner: *Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states.* Physical Review Letters, 69:2881–2884, 1992.

[27] Bentley, J.L.: *Multidimensional binary search tree used for associative searching.* Communications of the ACM, 18(9):509–517, 1975.

[28] Bergou, J., U. Herzog et M. Hillery: *Discrimination of quantum states.* Dans Paris, M. et J. Rehacek (rédacteurs): *Ch. 11: invited review article in Lecture Notes in Physics vol. 649: Quantum state estimation*, pages 417–465. Springer-Berlin, 2004.

[29] Berkhin, P.: *Survey of clustering data mining techniques.* Rapport technique, Accrue Software, 2002.

[30] Bernstein, E. et U. Vazirani: *Quantum complexity theory.* Dans *Proceedings of the 25th Annual ACM Symposium on Theory of Computing (STOC'93)*, pages 11–20, 1993.

[31] Bespamyatnikh, S.N.: *An efficient algorithm for the three-dimensional diameter problem.* Dans *Proceedings of the 9th Symposium on Discrete Mathematics (SODA'98)*, pages 137–146, 1998.

[32] Beygelzimer, A., V. Dani, T. Hayes, J. Langford et B. Zadrozny: *Error limiting reductions between classification tasks.* Dans *Proceedings of the 22th Annual International Conference of Machine Learning (ICML'05)*, pages 49–56, 2005.

[33] Beygelzimer, A., J. Langford et P. Ravikumar: *Multiclass classification with filter trees.* 2008. Version préliminaire disponible à `http://hunch.net/~jl/projects/reductions/mc_to_b/invertedTree.pdf`.

[34] Beygelzimer, A., J. Langford et B. Zadrozny: *Weighted one-against-all.* Dans *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI'05)*, pages 720–725, 2005.

[35] Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algoritms.* Plenum Press, 1981.

[36] Bishop, C.: *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[37] Blumer, A., A. Ehrenfeucht, D. Haussler et M.K. Warmuth: *Learnability and the Vapnik-Chernovenkis dimension.* Journal of the ACM, 36(4):929–965, 1989.

[38] Bonner, R. et R. Freivalds: *Quantum learning by finite automata.* Dans *Proceedings of the 3rd International Workshop on Quantum Computation and Learning (QCL'02)*, pages 85–96, 2002. Disponible à l'adresse `http://www.mdh.se/ima/forskning/prints/02.proceedings/proc-200205xx-yy_riga.pdf`.

[39] Bonner, R. et R. Freivalds: *A survey of quantum learning.* Dans *Proceedings of the 3rd International Workshop on Quantum Computation and Learning (QCL'02)*, pages 106–119, 2002. Disponible à l'adresse `http://www.mdh.se/ima/forskning/prints/02.proceedings/proc-200205xx-yy_riga.pdf`.

[40] Borůvka, O.: *O jistém problému minimálním im.* Práce Moravské Přirodovědecké Společnosti, 3:37–58, 1926.

[41] Boyer, M., G. Brassard, P. Høyer et A. Tapp: *Tight bounds on quantum searching.* Fortschritte Der Physik, 46:493–505, 1998.

[42] Brassard, G.: *Quantum Information Processing for Computer Scientists.* MIT Press. En préparation.

[43] Brassard, G.: *Quantum communication complexity.* Foundations of Physics, 33(11):1593–1616, 2003.

[44] Brassard, G., A. Broadbent, J. Fitzsimons, S. Gambs et A. Tapp: *Anonymous quantum communication.* Dans *Proceedings of 13th Annual International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT'07)*, pages 460–473, 2007. Accepté pour présentation at the 11th Workshop on Quantum Information Processing (QIP), New Delhi, décembre 2007.

[45] Brassard, G., A. Broadbent et A. Tapp: *Quantum pseudo-telepathy.* Foundations of Physics, 35:1877–1907, 2005.

[46] Brassard, G., P. Høyer, M. Mosca et A. Tapp: *Quantum amplitude amplification and estimation.* Contemporary Mathematics, 305:53–74, 2002.

[47] Brassard, G., P. Høyer et A. Tapp: *Quantum counting.* Dans *Proceedings of the International Conference on Automata, Languages and Programming (ICALP'98)*, pages 820–831, 1998.

[48] Breiman, L.: *Bagging predictors.* Machine Learning, 24(2):123–140, 1996.

[49] Breiman, L.: *Random forests.* Machine Learning, 45(1):5–32, 2001.

[50] Breunig, M.M, H. P. Kriegel, R.T. Ng et J. Sander: *LOF: Identifying density-based local outliers.* Dans *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 93–104, 2000.

[51] Broadbent, A. et E. Kashefi: *Parallelizing quantum circuits.* 2007. Disponible sur `http://arxiv.org/abs/0704.1736`.

[52] Bshouty, N. et J.C. Jackson: *Learning DNF over the uniform distribution using a quantum example oracle.* Dans *Proceedings of the 8th Annual Conference on Computational Learning Theory (COLT'95)*, pages 118–127, 1995.

[53] Buhrman, H., R. Cleve, J. Watrous et R. de Wolf: *Quantum fingerprinting.* Physical Review Letters, 87(16):167902, 2001.

[54] Bužek, V.: *Quantum tomography from incomplete data via MaxEnt principle.* Dans *Lectures Notes in Physics*, pages 189–234. Springer-Verlag Berlin, 2004.

[55] Childs, A.M., R. Cleve, S.P. Jordan et D.L. Yeung: *Discrete-query quantum algorithm for NAND trees.* 2007. Disponible sur `quant-ph/0702160`.

[56] Chiribella, G., G. Mauro D'Ariano, P. Perinotti et M. Sacchi: *Covariant quantum measurements which maximize the likelihood.* Physical Review A, 70(062105), 2004.

[57] Chrisley, R.: *Quantum learning.* Dans *New Directions in Cognitive Science: Proceedings of the International Symposium of Finnish Association of Artificial Intelligence*, pages 77–89, 1995.

[58] Chrisley, R.: *Learning in non-superpositional quantum neurocomputers.* Dans Pylkkänen, P. et P. Pylkkö (rédacteurs): *Brain, Mind and Physics*, pages 126–139. Amsterdam:IOS Press, 1997.

[59] Cleve, R., W. van Dam, M. Nielsen et A. Tapp: *Quantum entanglement and the communication complexity of the inner product function.* Dans *Proceedings of the First NASA International Conference on Quantum Computing and Quantum Communications*, pages 61–74, 1999.

[60] Cleve, R., A. Ekert, C. Macchiavello et M. Mosca: *Quantum algorithmes revisited.* Proceedings of the Royal Society of London, Series A, 454(1969):339–354, 1998.

[61] Cleve, R., D. Gavinsky et D.L. Yeung: *Quantum algorithms for evaluating MIN-MAX trees.* 2007. Disponible sur `http://arxiv.org/abs/0710.5794`.

[62] Cox, T. et M. Cox: *Multidimensional scaling.* Chapman and Hall, 1994.

[63] Croke, S., E. Andersson, S.M. Barnett, C.R. Gilson et J. Jeffers: *Maximum confidence quantum measurements.* Physical Review Letters, 96, 2006.

[64] Dasgupta, S.: *Learning mixtures of Gaussians.* Dans *Proceedings of the 40th Annual Symposium on the Foundations of Computer Science (FOCS'99)*, pages 634–644, 1999.

[65] Dasgupta, S. et P.M. Long: *Performance guarantee for hierarchical clustering.* Journal of Computer and System Sciences, 70(4), 2005.

[66] Day, W. et H. Edelsbrunner: *Efficient algorithms for agglomerative hierarchical clustering methods.* Journal of Classification, 1(1):7–24, 1984.

[67] Dempster, A.P., N.M. Laird et D.B. Rubin: *Maximum-likelihood from imcomplete data via the EM algorithm.* Journal of Royal Statistical Society B, 39:1–38, 1977.

[68] Deutsch, D.: *Quantum theory, the Church-Turing principle and the universal quantum computer.* Proceedings of the Royal Society of London, 400:96–117, 1985.

[69] Deutsch, D.: *Quantum computational networks.* Proceedings of the Royal Society of London, 425:73–90, 1989.

[70] Deutsch, D. et R. Jozsa: *Rapid solutions of problem by quantum computation.* Proceedings of the Royal Society of London, 439:553–558, 1992.

[71] Devijver, P.A. et J. Kittler: *Pattern Recognition: a Statistical Approach.* Prentice Hall International, 1982.

[72] Dieks, D.: *Communication by EPR devices.* Physics Letter A, 92(6):271–272, 1982.

[73] Dietterich, T.G.: *Approximate statistical tests for comparing supervised classification algorithms.* Neural computation, 10(7):1895–1924, 1998.

[74] Dietterich, T.G. et G. Bakiri: *Solving multiclass learning problems via error-correcting output codes.* Journal of Artificial Intelligence Research, 2:263–284, 1995.

[75] Dijkstra, E.W.: *A note on two problems in connexion with graphs.* Numerische Mathematik, 1:269–271, 1959.

[76] Dong, D., C. Chen et Z. Chen: *Quantum reinforcement learning.* Dans *Proceedings of the First International of Advances in Natural Computation (ICNC'05)*, pages 686–689, 2005.

[77] Dürr, C., M. Heiligman, P. Høyer et M. Mhalla: *Quantum query complexity of some graph problems.* Dans *Proceedings of the International Conference on Automata, Languages and Programming (ICALP'04)*, pages 481–493, 2004.

[78] Dürr, C. et P. Høyer: *A quantum algorithm for finding the minimum.* 1996. Disponible sur `quant-ph/9607014`.

[79] Duda, R., P. Hart et D. Stork: *Pattern Classification (Chapitre 10).* Wiley-Interscience, 2001.

[80] Eggermont, P.P.B. et V.N. LaRiccia: *Maximum Penalized Likelihood Estimation.* Springer-Verlag, 2005.

[81] Einstein, A., B. Podolsky et N. Rosen: *Can quantum-mechanical description of physical reality be considered complete?* Physical Review, 47:777–780, 1935.

[82] Ester, M., H.P. Kriegel, J. Sander et X. Xu: *A density-based algorithm for discovering clusters in large spatial databases with noise.* Dans *Proceedings of the 2nd ACM SIGKDD*, pages 226–231, 1996.

[83] Ezhov, A.A. et G.P. Berman: *Introduction to quantum neural technologies.* Rinton Press, 2003.

[84] Farhi, E., J. Goldstone et S. Gutmann: *A quantum algorithm for the Hamiltonian nand tree.* 2007. Disponible sur `quant-ph/0702144`.

[85] Feynman, R.: *Simulating Physics with computer.* International Journal of Theoretical Physics, 21:467–488, 1982.

[86] Finocchiaro, D.V. et M. Pellegrini: *On computing the diameter of a point set in high dimensional euclidean space.* Theoretical Computer Science, 287(2):501–514, 2002.

[87] Fisher, D.H.: *Knowledge acquisition via incremental concept clustering.* Machine Learning, 2:609–616, 1987.

[88] Fredkin, E. et T. Toffoli: *Conservative logic.* International Journal of Theoretical Physics, 29:219–253, 1982.

[89] Freund, Y. et R. Schapire: *A decision-theoretic generalization of on-line learning and an application to boosting.* Journal of Computer and System Sciences, 55(1):119–139, 1997.

[90] Funahashi, K.: *On the approximate realization of continuous mappings by neural networks.* Neural Networks, 2:183–192, 1989.

[91] Gall, F. Le: *Exponential separation of quantum and classical online space complexity.* Dans *Proceedings of the 18th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'06)*, pages 67–73, 2006.

[92] Gambs, S.: *Quantum classification.* 2008. En préparation.

[93] Gambs, S.: *Quantum learning : a survey.* 2008. En préparation.

[94] Gambs, S., B. Kégl et E. Aïmeur: *Privacy-preserving boosting.* Data Mining and Knowledge Discovery, 14(1):131–170, 2007.

[95] Gammelmark, S. et K. Mølmer: *Quantum learning by measurement and feedback.* Disponible sur `arXiv:0803.1418v2[quant-ph]`.

[96] Garey, M. et D.S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.

[97] Gavinsky, D., J. Kempe et R. de Wolf: *Strengths and weaknesses of quantum fingerprinting.* Dans *Proceedings of the 21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 288–298, 2006.

[98] Ghahramani, Z.: *Advances Lectures on Machine Learning (Chapitre).* Springer-Verlag, 2004.

[99] Goldberger, J., S. Roweis, G. Hinton et R. Salakhutdinow: *Neighbourhood components analysis.* Dans *Proceedings of the Neural Information Processing Systems (NIPS'05)*, pages 513–520, 2005.

[100] González, T.F.: *Clustering to minimize the maximum intercluster distance.* Theoretical Computer Science, 38, 1985.

[101] Gower, J.C. et G.J.S. Ross: *Minimum spanning trees and single linkage cluster analysis.* Applied Statistics, 18(1), 1969.

[102] Grünwald, P.D.: *The Minimum Description Length Principle.* MIT Press, 2007.

[103] Grover, L.K.: *Quantum mechanics helps in searching for a needle in a haystack.* Physical Review Letters, 79(2):325–328, 1997.

[104] Grover, L.K.: *A framework for fast quantum mechanical algorithms.* Dans *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC'98)*, pages 53–62, 1998.

[105] Hamerly, G. et C. Elkan: *Learning the k in k-means.* Dans *Proceedings of the Neural Information Processing Systems (NIPS'03)*, pages 281–288, 2003.

[106] Harel, D. et Y. Koren: *On clustering using random walks.* Dans *Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'01)*, pages 18–41, 2001.

[107] Hausladen, P. et W.K. Wootters: *A "pretty good" measurement for distinguishing quantum states.* Journal of Modern Optics, 41, 1994.

[108] Hayashi, A., T. Hashimoto et M. Horibe: *Reexamination of optimal quantum state estimation of pure states.* Physical Reviews A, 72, 2005.

[109] Helstrom, C.W.: *Quantum Detection and Estimation Theory.* Academic Press, 1976.

[110] Helstrom, C.W.: *Quantum detection and estimation theory.* Academic Press, 1976.

[111] Herzog, U. et J.A. Bergou: *Optimum unambiguous discrimination of two mixed quantum states.* Physical Reviews A, 71, 2005.

[112] Hinnenburg, A. et D. Keim: *An efficient approach to clustering large multimedia databases with noise.* Dans *Proceedings of the 4th ACM SIGKDD*, pages 58–65, 1998.

[113] Hinton, G., S. Osindero et Y. Teh: *A fast learning algorithm for deep belief nets.* Neural Computation, 18:1527–1554, 2006.

[114] Hinton, G. et T.J. Sejnowski (rédacteurs): *Unsupervised Learning - Foundation of Neural Computation.* MIT Press, 1999.

[115] Holevo, A.S.: *Bounds for the quantity of information transmitted by a quantum mechanical channel.* Problems of Information Transmissions, 9:177–183, 1973.

[116] Horn, D. et A. Gottlieb: *The method of quantum clustering.* Dans *Proceedings of the Neural Information Processing Systems (NIPS'01)*, pages 769–776, 2001.

[117] Horn, D. et A. Gottlieb: *Algorithms for data clustering in pattern recognition problems based on quantum mechanics.* Physical Review Letters, 88(1), 2002.

[118] Horodecki, M., S. Horodecki, A. Sen De et U. Sen: *Common origin of no-cloning and no-deleting principles - conservation of information.* 2004. Disponible sur `quant-ph/0407038`.

[119] Jain, A.K., M.N. Murty et P.J. Flynn: *Data clustering: a review.* ACM Computing Surveys, 31(3):264–323, 1999.

[120] Jolliffe, I.T.: *Principal Component Analysis.* Springer-Verlag, 1986.

[121] Jordan, M.I.: *Learning in Graphical Models.* MIT Press, 1999.

[122] Jozsa, R.: *An introduction to measurement based quantum computation.* 2005. Disponible sur `http://arxiv.org/abs/quant-ph/0508124`.

[123] Jozsa, R. et S. Presnell: *Universal quantum information compression and degrees of prior knowledge.* Proceedings of the Royal Society of London, 459:3061–3077, 2003.

[124] Jozsa, R. et J. Schlienz: *Distinguishability of states and von Neumann entropy.* Physical Reviews A, 62(012301), 2000.

[125] Kaufman, L. et P. Rousseeuw: *Clustering by means of medoids.* Dans *Statistical Data Analysis Based on the L_1-Norm and Related Methods*, pages 405–416. Y. Dodge (éditeur), North-Holland, Amsterdam, 1987.

[126] Kaye, P., R. Laflamme et M. Mosca: *An Introduction to Quantum Computing.* Oxford University Press, 2007.

[127] Kearns, M.J. et L.G. Valiant: *Cryptographic limitations on learning Boolean formulae and finite automata.* Journal of the ACM, 41(1):67–95, 1994.

[128] Kempe, J.: *Quantum random walks — an introductory overview.* Contemporary Physics, 44(4):307–327, 2003.

[129] Kibriya, A.M. et E. Frank: *An empirical comparison of exact nearest neighbour algorithms.* Dans *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'07)*, pages 140–151, 2007.

[130] Klivans, A.R. et A.A. Sherstov: *Cryptographic hardness for learning intersections of halfspaces.* Dans *Proceedings of the 47th Foundations of Computer Science (FOCS'06)*, pages 553–562, 2006.

[131] Kohavi, R.: *A study of cross-validation and boostrap for accuracy estimation and model selection.* Dans *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 1137–1145, 1995.

[132] Kushilevitz, E. et N. Nisan: *Communication Complexity.* Cambridge University Press, 1997.

[133] Langford, J. et A. Beygelzimer: *Sensitive error correcting output codes.* Dans *Proceedings of the 18th Annual Conference on Computational Learning Theory (COLT'05)*, pages 158–172, 2005.

[134] Langford, J. et B. Zadrozny: *Estimating class membership probabilites using classifiers learners.* Dans *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS'05)*, pages 198–205, 2005.

[135] Langley, P., W. Iba et K. Thompson: *An analysis of Bayesian classifiers.* Dans *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 223–228, 1992.

[136] Lenstra, A.K., H.W. Lenstra, M.S. Manasse et J.M. Pollard: *The number field sieve.* Dans *Proceedings of the ACM Symposium on Theory of Computing (STOC'90)*, pages 564–572, 1990.

[137] Linial, N. et A. Shraibman: *Learning complexity vs. communication complexity.* Dans *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity (CCC'08)*, pages 53–63, 2008.

[138] MacQueen, J.: *Some methods for classification and analysis of multivariate observations.* Dans *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1965.

[139] Magniez, F., A. Nayak, J. Roland et M. Santha: *Search via quantum walk.* Dans *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC'07)*, pages 575–584, 2007.

[140] Mahalanobis, P.C.: *On the generalized distance in statistics.* Proceedings of the National Institute of Science of India, 12:49–55, 1936.

[141] Massart, D.L., L. Kaufman et P.J. Rousseeuw: *Least median of squares–a robust method for outlier and model error detection in regression and calibration.* Analytica Chimica Acta, (187):171–179, 1986.

[142] McLachlan, G.J. et E. Basford: *Mixture Models.* Dekker, 1988.

[143] Meila, M.: *Comparing clusterings: an axiomatic view.* Dans *Proceedings of the 22nd International Conference on Machine Learning (ICML'05)*, pages 577–584, 2005.

[144] Mishra, N., D. Oblinger et L. Pitt: *Sublinear time approximate clustering.* Dans *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms (SODA '01)*, pages 439–447, 2001.

[145] Mitchell, T.: *Machine Learning.* McGraw Hill, 1997.

[146] Montanaro, A.: *Structure, randomness and complexity in quantum computation.* Thèse de doctorat, University of Bristol, United Kingdom.

[147] Montanaro, A.: *A lower bound on the probability of error in quantum state discrimination.* Dans *Proceedings of IEEE Information Theory Workshop 2008*, 2008.

[148] Nayak, A.: *Lower Bounds for Quantum Computation and Communication.* Thèse de doctorat, University of California, Berkeley.

[149] Nayak, A. et F. Wu: *The quantum query complexity of approximating the median and related statistics.* Dans *Proceedings of the 31st ACM Symposium on Theory of Computing (STOC'99)*, pages 384–393, 1999.

[150] Newman, I. et M. Szegedy: *Public vs. private coin flips in one round communication games.* Dans *Proceedings of the 28th ACM Symposium on Theory of Computing (STOC'96)*, pages 561–570, 1996.

[151] Ng, A.Y., M.I. Jordan et Y. Weiss: *On spectral clustering: Analysis and an algorithm.* Dans *Proceedings of the Neural Information Processing Systems (NIPS'01)*, 2001.

[152] Nielsen, M.A. et I. Chuang: *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[153] Papadimitriou, C.H.: *Worst-case and probabilistic analysis of a geometric location problem.* SIAM Journal of Computing, 10(3):542–557, 1981.

[154] Paris, M. et J. Řeháček (rédacteurs): *Quantum State Estimation.* Springer, 2004.

[155] Parzen, E.: *On the estimation of a probability density function and mode.* Annals of Mathematical Statistics, 33:1064–1076, 1962.

[156] Peres, A.: *Quantum theory: Concepts and methods.* Kluwer Academic Publishers, 1993.

[157] Peres, A. et W.K. Wootters: *Optimal detection of quantum information.* Physical Review Letters, 66(9), 1991.

[158] Preparata, F.P. et M.I. Shamos: *Computational Geometry: an Introduction.* Springer-Verlag, New-York, 1985.

[159] Prim, R.: *Shortest connecting networks and some generalizations.* Bell Systems Technical Journal, 1957.

[160] Quinlan, J.R.: *Induction of decision trees.* Machine Learning, 1:81–106, 1986.

[161] Quinlan, J.R.: *C4.5: Programs for machine learning.* Morgan Kaufmann, 1993.

[162] Regev, O.: *On lattices, learning with errors, random linear codes, and cryptography.* Dans *Proceedings of the ACM Symposium on Theory Of Computing (STOC'05)*, pages 84–93, 2005.

[163] Ricks, B. et D. Ventura: *Training a quantum neural network.* Dans *Proceedings of the Neural Information Processing Systems (NIPS'03)*, 2003.

[164] Rivest, R.L., A. Shamir et L. Adleman: *A method of obtaining digital signatures and public-key cryptosystems.* Communications of ACM, 21(2):120–126, 1978.

[165] Roweis, S. et L. Saul: *Nonlinear dimensionality reduction by locally linear embedding.* Science, 290(5500):2323–2326, 2000.

[166] Sasaki, M. et A. Carlini: *Quantum learning and universal quantum matching machine.* Physical Reviews A, 66(2), 2002.

[167] Schumacher, B.: *Quantum coding.* Physical Reviews A, 51, 1995.

[168] Servedio, R.: *Separating quantum and classical learning.* Dans *Proceedings of the International Conference on Automata, Languages and Programming (ICALP'01)*, pages 1065–1080, 2001.

[169] Servedio, R. et A. Gortler: *Equivalences and separations between quantum and classical learnability.* SIAM Journal of Computing, 33(5):1067–1092, 2004.

[170] Shannon, C.E.: *A mathematical theory of communication.* Bell Systems Technical Journal, 27:379–423/623–656, 1948.

[171] Shi, Y.: *Both Toffoli and controlled-NOT need little help to do universal quantum computation.* Quantum Information and Computation, 3(1):84–92, 2003.

[172] Shor, P.W.: *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.* SIAM Journal of Computing, 26:1484–1509, 1997.

[173] Silva, V. de et J.B Tenenbaum: *Sparse multidimensional scaling using landmark points.* Dans *in preparation.* En préparation.

[174] Silva, V. de et J.B Tenenbaum: *Global versus local methods in nonlinear dimensionality reduction.* Dans *Proceedings of the Neural Information Processing Systems (NIPS'02)*, pages 721–728, 2002.

[175] Simon, D.: *On the power of quantum computation.* SIAM Journal of Computing, 26(5):1474–1483, 1997.

[176] Small, C.G.: *A survey of multidimensional medians.* International Statistical Review, 58(3):263–277, 1990.

[177] Sutton, R.S. et A.G. Barto: *Reinforcement learning: An introduction.* MIT Press, 1998.

[178] Szegedy, M.: *Quantum speed-up of Markov chain based algorithms.* Dans *Proceedings of Foundations Of Computer Science (FOCS'04)*, pages 32–41, 2004.

[179] Tenenbaum, J.B., V. de Silva et J.C. Langford: *A global geometric framework for nonlinear dimensionality reduction.* Science, 290(5500):2319–2323, 2000.

[180] Toffoli, T.: *Physics and computation.* International Journal of Theoretical Physics, 21:165–175, 1982.

[181] Valiant, L.G.: *A theory of the learnable.* Communications of the ACM, 27:1134–1142, 1984.

[182] Vapnik, V.: *The Nature of Statistical Learning Theory.* Springer, 1995.

[183] Vernam, G.S.: *Cipher printing telegraph systems for secret wire and radio telegraphic communications.* Journal of American Institute of Electrical Engineering, 45:109–115, 1926.

[184] Vollmer, H.: *Introduction to Circuit Complexity: a Uniform Approach.* Springer, 1999.

[185] Warmuth, M.K.: *A Bayes rule for density matrices.* Dans *Proceedings of the Neural Information Processing Systems (NIPS'05)*, 2005.

[186] Warmuth, M.K. et D. Kuzmin: *A Bayesian probability calculus for density matrices.* Dans *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI'06)*, pages 503–511, 2006.

[187] Warmuth, M.K. et D. Kuzmin: *Online kernel PCA with entropic matrix updates.* Dans *Proceedings of the 22th International Conference for Machine Learning (ICML'07)*, pages 465–472, 2007.

[188] Witten, I.H. et E. Frank: *Data Mining: Practical Machine Learning Tools and Techniques (2nd Edition).* Morgan Kaufmann, 2005.

[189] Wolf, R. de: *Quantum communication and complexity.* Theoretical Computer Science, 287(1):337–353, 2002.

[190] Wolpert, D.H.: *Stacked generalization.* Neural Networks, 5:241–249, 1992.

[191] Wolpert, D.H.: *The supervised learning no-free-lunch theorems.* Dans *Proceedings of the 6th Online World Conference on Soft Computing in Industrial Applications*, 2001.

[192] Wootters, W.K. et W.C. Żurek: *A single quantum cannot be cloned.* Nature, 66:802–803, 1982.

[193] Yamanishi, K., J. Takeuchi, G.J. Williams et P. Milne: *On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms.* Data Mining and Knowledge Discovery, 8(3):275–300, 2004.

[194] Yao, A.: *Quantum circuit complexity.* Dans *Proceedings of Foundations Of Computer Science (FOCS'93)*, pages 352–361, 1993.

[195] Zadrozny, B., J. Langford et A. Naoki: *Cost-sensitive learning by cost-proportionate example weighting.* Dans *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, pages 435–442, 2003.

[196] Zahn, C.T.: *Graph-theoretical methods for detecting and describing gestalt clusters.* IEEE Transactions on Computers, 20(1):68–86, 1971.

[197] Zhang, T., R. Ramakrishnan et M. Livny: *BIRCH: an efficient data clustering method for very large databases.* Dans *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pages 103–114, 1996.

[198] Zhu, Xiaojin: *Semi-Supervised Learning with Graphs.* Thèse de doctorat, Carnegie Mellon University, 2005. CMU-LTI-05-192.

[199] Ziman, M., M. Plesch et V. Bužek et P. Štelmachovič: *Process reconstruction: From unphysical to physical maps via maximum likelihood.* Physical Reviews A, 72, 2005.