ELSEVIER
爱思唯尔

# Quantum Machine Learning What Quantum Computing Means to Data Mining

HIT

国外优秀物理著作原版系列

# 量子机器学习中数据挖掘的量子计算方法

[匈] Wittek, P.（维特克）著

哈爾濱工業大學出版社
HARBIN INSTITUTE OF TECHNOLOGY PRESS

HIT

国外优秀物理著作
原版系列

# 量子机器学习中数据挖掘的量子计算方法

Quantum Machine Learning What Quantum Computing Means to Data Mining

[阌] Wittek, P.(维特克) 著

HITP
哈爾濱工業大學出版社
HARBIN INSTITUTE OF TECHNOLOGY PRESS

黑版贸审字 08-2015-062 号

Quantum Machine Learning What Quantum Computing Means to Data Mining
Peter Wittek
ISBN:9780128009536


Authorized English language reprint edition published by Elsevier (Singapore) Pte Ltd. and Harbin Institute of Technology Press


Elsevier (Singapore) Pte Ltd.
3 Killiney Road, #08-01 Winsland House I, Singapore 239519
Tel: (65)6349-0200 Fax: (65) 6733-1817
First Published 2016
2016 年初版


**图书在版编目(CIP)数据**

量子机器学习中数据挖掘的量子计算方法=Quantum Machine Learning What Quantum Computing Means to Data Mining:英文/(匈)维特克(Wittek, P.)著.—哈尔滨:哈尔滨工业大学出版社,2016.1
ISBN 978-7-5603-5759-1

Ⅰ.①量… Ⅱ.①维… Ⅲ.①数据采集-应用-量子力学-计算方法-英文
Ⅳ.①O413.1-39

中国版本图书馆 CIP 数据核字(2015)第 291576 号

策划编辑 刘培杰
责任编辑 张永芹 聂兆慈
封面设计 孙茵艾
出版发行 哈尔滨工业大学出版社
社　　址 哈尔滨市南岗区复华四道街 10 号 邮编 150006
传　　真 0451-86414749
网　　址 http://hitpress.hit.edu.cn
印　　刷 哈尔滨市工大节能印刷厂
开　　本 787mm×1092mm 1/16 印张 12 字数 260 千字
版　　次 2016 年 1 月第 1 版 2016 年 1 月第 1 次印刷
书　　号 ISBN 978-7-5603-5759-1
定　　价 98.00 元

(如因印装质量问题影响阅读,我社负责调换)

# Preface

Machine learning is a fascinating area to work in: from detecting anomalous events in live streams of sensor data to identifying emergent topics involving text collection, exciting problems are never too far away.

Quantum information theory also teems with excitement. By manipulating particles at a subatomic level, we are able to perform Fourier transformation exponentially faster, or search in a database quadratically faster than the classical limit. Superdense coding transmits two classical bits using just one qubit. Quantum encryption is unbreakable—at least in theory.

The fundamental question of this monograph is simple: What can quantum computing contribute to machine learning? We naturally expect a speedup from quantum methods, but what kind of speedup? Quadratic? Or is exponential speedup possible? It is natural to treat any form of reduced computational complexity with suspicion. Are there tradeoffs in reducing the complexity?

Execution time is just one concern of learning algorithms. Can we achieve higher generalization performance by turning to quantum computing? After all, training error is not that difficult to keep in check with classical algorithms either: the real problem is finding algorithms that also perform well on previously unseen instances. Adiabatic quantum optimization is capable of finding the global optimum of nonconvex objective functions. Grover's algorithm finds the global minimum in a discrete search space. Quantum process tomography relies on a double optimization process that resembles active learning and transduction. How do we rephrase learning problems to fit these paradigms?

Storage capacity is also of interest. Quantum associative memories, the quantum variants of Hopfield networks, store exponentially more patterns than their classical counterparts. How do we exploit such capacity efficiently?

These and similar questions motivated the writing of this book. The literature on the subject is expanding, but the target audience of the articles is seldom the academics working on machine learning, not to mention practitioners. Coming from the other direction, quantum information scientists who work in this area do not necessarily aim at a deep understanding of learning theory when devising new algorithms.

This book addresses both of these communities: theorists of quantum computing and quantum information processing who wish to keep up to date with the wider context of their work, and researchers in machine learning who wish to benefit from cutting-edge insights into quantum computing.

I am indebted to Stephanie Wehner for hosting me at the Centre for Quantum Technologies for most of the time while I was writing this book. I also thank Antonio Acín for inviting me to the Institute for Photonic Sciences while I was finalizing the manuscript. I am grateful to Sándor Darányi for proofreading several chapters.

Peter Wittek
Castelldefels, May 30, 2014

# Contents

# Notations

| | |
|---|---|
| $\mathbb{1}$ | indicator function |
| $\mathbb{C}$ | set of complex numbers |
| $d$ | number of dimensions in the feature space |
| $E$ | error |
| $\mathbb{E}$ | expectation value |
| $\mathbf{G}$ | group |
| $H$ | Hamiltonian |
| $\mathcal{H}$ | Hilbert space |
| $I$ | identity matrix or identity operator |
| $K$ | number of weak classifiers or clusters, nodes in a neural net |
| $N$ | number of training instances |
| $P_i$ | measurement: projective or POVM |
| $\mathbf{P}$ | probability measure |
| $\mathbb{R}$ | set of real numbers |
| $\rho$ | density matrix |
| $\sigma_x, \sigma_y, \sigma_z$ | Pauli matrices |
| tr | trace of a matrix |
| $U$ | unitary time evolution operator |
| $\mathbf{w}$ | weight vector |
| $\mathbf{x}, \mathbf{x}_i$ | data instance |
| $X$ | matrix of data instances |
| $y, y_i$ | label |
| $\top$ | transpose |
| $\dagger$ | Hermitian conjugate |
| $\lVert . \rVert$ | norm of a vector |
| $[., .]$ | commutator of two operators |
| $\otimes$ | tensor product |
| $\oplus$ | XOR operation or direct sum of subspaces |

# Part One

# Fundamental Concepts

# Introduction

# 1

The quest of machine learning is ambitious: the discipline seeks to understand what learning is, and studies how algorithms approximate learning. Quantum machine learning takes these ambitions a step further: quantum computing enrolls the help of nature at a subatomic level to aid the learning process.

Machine learning is based on minimizing a constrained multivariate function, and these algorithms are at the core of data mining and data visualization techniques. The result of the optimization is a decision function that maps input points to output points. While this view on machine learning is simplistic, and exceptions are countless, some form of optimization is always central to learning theory.

The idea of using quantum mechanics for computations stems from simulating such systems. Feynman (1982) noted that simulating quantum systems on classical computers becomes unfeasible as soon as the system size increases, whereas quantum particles would not suffer from similar constraints. Deutsch (1985) generalized the idea. He noted that quantum computers are universal Turing machines, and that quantum parallelism implies that certain probabilistic tasks can be performed faster than by any classical means.

Today, quantum information has three main specializations: quantum computing, quantum information theory, and quantum cryptography (Fuchs, 2002, p. 49). We are not concerned with quantum cryptography, which primarily deals with secure exchange of information. Quantum information theory studies the storage and transmission of information encoded in quantum states; we rely on some concepts such as quantum channels and quantum process tomography. Our primary focus, however, is quantum computing, the field of inquiry that uses quantum phenomena such as superposition, entanglement, and interference to operate on data represented by quantum states.

Algorithms of importance emerged a decade after the first proposals of quantum computing appeared. Shor (1997) introduced a method to factorize integers exponentially faster, and Grover (1996) presented an algorithm to find an element in an unordered data set quadratically faster than the classical limit. One would have expected a slew of new quantum algorithms after these pioneering articles, but the task proved hard (Bacon and van Dam, 2010). Part of the reason is that now we expect that a quantum algorithm should be faster—we see no value in a quantum algorithm with the same computational complexity as a known classical one. Furthermore, even

**Quantum Machine Learning. http://dx.doi.org/10.1016/B978-0-12-800953-6.00001-3**

with the spectacular speedups, the class NP cannot be solved on a quantum computer in subexponential time (Bennett et al., 1997).

While universal quantum computers remain out of reach, small-scale experiments implementing a few qubits are operational. In addition, quantum computers restricted to domain problems are becoming feasible. For instance, experimental validation of combinatorial optimization on over 500 binary variables on an adiabatic quantum computer showed considerable speedup over optimized classical implementations (McGeoch and Wang, 2013). The result is controversial, however (Rønnow et al., 2014).

Recent advances in quantum information theory indicate that machine learning may benefit from various paradigms of the field. For instance, adiabatic quantum computing finds the minimum of a multivariate function by a controlled physical process using the adiabatic theorem (Farhi et al., 2000). The function is translated to a physical description, the Hamiltonian operator of a quantum system. Then, a system with a simple Hamiltonian is prepared and initialized to the ground state, the lowest energy state a quantum system can occupy. Finally, the simple Hamiltonian is evolved to the target Hamiltonian, and, by the adiabatic theorem, the system remains in the ground state. At the end of the process, the solution is read out from the system, and we obtain the global optimum for the function in question.

While more and more articles that explore the intersection of quantum computing and machine learning are being published, the field is fragmented, as was already noted over a decade ago (Bonner and Freivalds, 2002). This should not come as a surprise: machine learning itself is a diverse and fragmented field of inquiry. We attempt to identify common algorithms and trends, and observe the subtle interplay between faster execution and improved performance in machine learning by quantum computing.

As an example of this interplay, consider convexity: it is often considered a virtue in machine learning. Convex optimization problems do not get stuck in local extrema, they reach a global optimum, and they are not sensitive to initial conditions. Furthermore, convex methods have easy-to-understand analytical characteristics, and theoretical bounds on convergence and other properties are easier to derive. Nonconvex optimization, on the other hand, is a forte of quantum methods. Algorithms on classical hardware use gradient descent or similar iterative methods to arrive at the global optimum. Quantum algorithms approach the optimum through an entirely different, more physical process, and they are not bound by convexity restrictions. Nonconvexity, in turn, has great advantages for learning: sparser models ensure better generalization performance, and nonconvex objective functions are less sensitive to noise and outliers. For this reason, numerous approaches and heuristics exist for nonconvex optimization on classical hardware, which might prove easier and faster to solve by quantum computing.

As in the case of computational complexity, we can establish limits on the performance of quantum learning compared with the classical flavor. Quantum learning is not more powerful than classical learning—at least from an information-theoretic perspective, up to polynomial factors (Servedio and Gortler, 2004). On the other hand, there are apparent computational advantages: certain concept classes are polynomial-time exact-learnable from quantum membership queries, but they are not polynomial-time learnable from classical membership queries (Servedio and Gortler, 2004). Thus quantum machine learning can take logarithmic time in both the number of vectors and their dimension. This is an exponential speedup over classical algorithms, but at the price of having both quantum input and quantum output (Lloyd et al., 2013a).

## 1.1 Learning Theory and Data Mining

Machine learning revolves around algorithms, model complexity, and computational complexity. Data mining is a field related to machine learning, but its focus is different. The goal is similar: identify patterns in large data sets, but aside from the raw analysis, it encompasses a broader spectrum of data processing steps. Thus, data mining borrows methods from statistics, and algorithms from machine learning, information retrieval, visualization, and distributed computing, but it also relies on concepts familiar from databases and data management. In some contexts, data mining includes any form of large-scale information processing.

In this way, data mining is more applied than machine learning. It is closer to what practitioners would find useful. Data may come from any number of sources: business, science, engineering, sensor networks, medical applications, spatial information, and surveillance, to mention just a few. Making sense of the data deluge is the primary target of data mining.

Data mining is a natural step in the evolution of information systems. Early database systems allowed the storing and querying of data, but analytic functionality was limited. As databases grew, a need for automatic analysis emerged. At the same time, the amount of unstructured information—text, images, video, music—exploded. Data mining is meant to fill the role of analyzing and understanding both structured and unstructured data collections, whether they are in databases or stored in some other form.

Machine learning often takes a restricted view on data: algorithms assume either a geometric perspective, treating data instances as vectors, or a probabilistic one, where data instances are multivariate random variables. Data mining involves preprocessing steps that extract these views from data.

For instance, in text mining—data mining aimed at unstructured text documents—the initial step builds a vector space from documents. This step starts with identification of a set of keywords—that is, words that carry meaning: mainly nouns, verbs, and adjectives. Pronouns, articles, and other connectives are disregarded. Words that occur too frequently are also discarded: these differentiate only a little between two text documents. Then, assigning an arbitrary vector from the canonical basis to each keyword, an indexer constructs document vectors by summing these basis vectors. The summation includes a weighting, where the weighting reflects the relative importance of the keyword in that particular document. Weighting often incorporates the global importance of the keyword across all documents.

The resulting vector space—the term-document space—is readily analyzed by a whole range of machine learning algorithms. For instance, $K$-means clustering identifies groups of similar documents, support vector machines learn to classify documents to predefined categories, and dimensionality reduction techniques, such as singular value decomposition, improve retrieval performance.

The data mining process often includes how the extracted information is presented to the user. Visualization and human-computer interfaces become important at this stage. Continuing the text mining example, we can map groups of similar documents on a two-dimensional plane with self-organizing maps, giving a visual overview of the clustering structure to the user.

Machine learning is crucial to data mining. Learning algorithms are at the heart of advanced data analytics, but there is much more to successful data mining. While quantum methods might be relevant at other stages of the data mining process, we restrict our attention to core machine learning techniques and their relation to quantum computing.

## 1.2 Why Quantum Computers?

We all know about the spectacular theoretical results in quantum computing: factoring of integers is exponentially faster and unordered search is quadratically faster than with any known classical algorithm. Yet, apart from the known examples, finding an application for quantum computing is not easy.

Designing a good quantum algorithm is a challenging task. This does not necessarily derive from the difficulty of quantum mechanics. Rather, the problem lies in our expectations: a quantum algorithm must be faster and computationally less complex than any known classical algorithm for the same purpose.

The most recent advances in quantum computing show that machine learning might just be the right field of application. As machine learning usually boils down to a form of multivariate optimization, it translates directly to quantum annealing and adiabatic quantum computing. This form of learning has already demonstrated results on actual quantum hardware, albeit countless obstacles remain to make the method scale further.

We should, however, not confine ourselves to adiabatic quantum computers. In fact, we hardly need general-purpose quantum computers: the task of learning is far more restricted. Hence, other paradigms in quantum information theory and quantum mechanics are promising for learning. Quantum process tomography is able to learn an unknown function within well-defined symmetry and physical constraints—this is useful for regression analysis. Quantum neural networks based on arbitrary implementation of qubits offer a useful level of abstraction. Furthermore, there is great freedom in implementing such networks: optical systems, nuclear magnetic resonance, and quantum dots have been suggested. Quantum hardware dedicated to machine learning may become reality much faster than a general-purpose quantum computer.

## 1.3 A Heterogeneous Model

It is unlikely that quantum computers will replace classical computers. Why would they? Classical computers work flawlessly at countless tasks, from word processing to controlling complex systems. Quantum computers, on the other hand, are good at certain computational workloads where their classical counterparts are less efficient.

Let us consider the state of the art in high-performance computing. Accelerators have become commonplace, complementing traditional central processing units. These accelerators are good at single-instruction, multiple-data-type parallelism, which is typical in computational linear algebra. Most of these accelerators derive from graphics processing units, which were originally designed to generate three-dimensional images at a high frame rate on a screen; hence, accuracy was not a consideration. With recognition of their potential in scientific computing, the platform evolved to produce high-accuracy double-precision floating point operations. Yet, owing to their design philosophy, they cannot accelerate just any workload. Random data access patterns, for instance, destroy the performance. Inherently single threaded applications will not show competitive speed on such hardware either. In contemporary high-performance computing, we must design algorithms using heterogeneous hardware: some parts execute faster on central processing units, others on accelerators. This model has been so successful that almost all supercomputers being built today include some kind of accelerator.

If quantum computers become feasible, a similar model is likely to follow for at least two reasons:

1. The control systems of the quantum hardware will be classical computers.
2. Data ingestion and measurement readout will rely on classical hardware.

More extensive collaboration between the quantum and classical realms is also expected. Quantum neural networks already hint at a recursive embedding of classical and quantum computing (Section 11.3). This model is the closest to the prevailing standards of high-performance computing: we already design algorithms with accelerators in mind.

## 1.4 An Overview of Quantum Machine Learning Algorithms

Dozens of articles have been published on quantum machine learning, and we observe some general characteristics that describe the various approaches. We summarize our observations in Table 1.1, and detail the main traits below.

Many quantum learning algorithms rely on the application of Grover's search or one of its variants (Section 4.5). This includes mostly unsupervised methods: $K$-medians, hierarchical clustering, or quantum manifold embedding (Chapter 10). In addition, quantum associative memory and quantum neural networks often rely on this search (Chapter 11). An early version of quantum support vector machines also

**Table 1.1 The Characteristics of the Main Approaches to Quantum Machine Learning**

| Algorithm | Reference | Grover | Speedup | Quantum Data | Generalization Performance | Implementation |
|---|---|---|---|---|---|---|
| *K*-medians | Aïmeur et al. (2013) | Yes | Quadratic | No | No | No |
| Hierarchical clustering | Aïmeur et al. (2013) | Yes | Quadratic | No | No | No |
| *K*-means | Lloyd et al. (2013a) | Optional | Exponential | Yes | No | No |
| Principal components | Lloyd et al. (2013b) | No | Exponential | Yes | No | No |
| Associative memory | Ventura and Martinez (2000) | Yes | | No | No | No |
| | Trugenberger (2001) | No | | No | No | No |
| Neural networks | Narayanan and Menneer (2000) | Yes | | No | Numerical | Yes |
| Support vector machines | Anguita et al. (2003) | Yes | Quadratic | No | Analytical | No |
| | Rebentrost et al. (2013) | No | Exponential | Yes | No | No |
| Nearest neighbors | Wiebe et al. (2014) | Yes | Quadratic | No | Numerical | No |
| Regression | Bisio et al. (2010) | No | | Yes | No | No |
| Boosting | Neven et al. (2009) | No | Quadratic | No | Analytical | Yes |

The column headed "Algorithm" lists the classical learning method. The column headed "Reference" lists the most important articles related to the quantum variant. The column headed "Grover" indicates whether the algorithm uses Grover's search or an extension thereof. The column headed "Speedup" indicates how much faster the quantum variant is compared with the best known classical version. "Quantum data" refers to whether the input, output, or both are quantum states, as opposed to states prepared from classical vectors. The column headed "Generalization performance" states whether this quality of the learning algorithm was studied in the relevant articles. "Implementation" refers to attempts to develop a physical realization.

uses Grover's search (Section 12.2). In total, about half of all the methods proposed for learning in a quantum setting use this algorithm.

Grover's search has a quadratic speedup over the best possible classical algorithm on unordered data sets. This sets the limit to how much faster those learning methods that rely on it get. Exponential speedup is possible in scenarios where both the input and the output are also quantum: listing class membership or reading the classical data once would imply at least linear time complexity, which could only be a polynomial speedup. Examples include quantum principal component analysis (Section 10.3), quantum $K$-means (Section 10.5), and a different flavor of quantum support vector machines (Section 12.3). Regression based on quantum process tomography requires an optimal input state, and, in this regard, it needs a quantum input (Chapter 13). At a high level, it is possible to define an abstract class of problems that can only be learned in polynomial time by quantum algorithms using quantum input (Section 2.5).

A strange phenomenon is that few authors have been interested in the generalization performance of quantum learning algorithms. Analytical investigations are especially sparse, with quantum boosting by adiabatic quantum computing being a notable exception (Chapter 14), along with a form of quantum support vector machines (Section 12.2). Numerical comparisons favor quantum methods in the case of quantum neural networks (Chapter 11) and quantum nearest neighbors (Section 12.1).

While we are far from developing scalable universal quantum computers, learning methods require far more specialized hardware, which is more attainable with current technology. A controversial example is adiabatic quantum optimization in learning problems (Section 14.7), whereas more gradual and well founded are small-scale implementations of quantum perceptrons and neural networks (Section 11.4).

## 1.5 Quantum-Like Learning on Classical Computers

Machine learning has a lot to adopt from quantum mechanics, and this statement is not restricted to actual quantum computing implementations of learning algorithms. Applying principles from quantum mechanics to design algorithms for classical computers is also a successful field of inquiry. We refer to these methods as quantum-like learning. Superposition, sensitivity to contexts, entanglement, and the linearity of evolution prove to be useful metaphors in many scenarios. These methods are outside our scope, but we highlight some developments in this section. For a more detailed overview, we refer the reader to Manju and Nigam (2012).

Computational intelligence is a field related to machine learning that solves optimization problems by nature-inspired computational methods. These include swarm intelligence (Kennedy and Eberhart, 1995), force-driven methods (Chatterjee et al., 2008), evolutionary computing (Goldberg, 1989), and neural networks (Rumelhart et al., 1994). A new research direction which borrows metaphors from quantum physics emerged over the past decade. These quantum-like methods in machine learning are in a way inspired by nature; hence, they are related to computational intelligence.