k近傍探索のための量子アルゴリズム

A Quantum Algorithm for k-Nearest Neighbor Search

岩村　雅一（ Masakazu Iwamura, Ph. D.）
大阪府立大学　大学院工学研究科　知能情報工学分野　准教授
(Associate Professor, Dept. of Computer Science and Intelligent Systems, Osaka Prefecture University)
電子情報通信学会，情報処理学会，IEEE，ACM
受賞：MIRU 学生優秀賞(2018 年). MVA Best Paper Award (2017 年). IAPR/ICDAR Young Investigator Award (2011 年). IAPR Nakano Award (Best Paper Award) (2010 年). ICFHR Best Paper Award (2010 年). IAPR/ICDAR Best Paper Award (2007 年). 電子情報通信学会 論文賞 (2006 年) 他
著書：Reading-Life Log as a New Paradigm of Utilizing Character and Document Media, Springer (2017 年） Camera-Based Document Analysis and Recognition - 5th International Workshop, CBDAR 2013, Revised Selected Papers, Springer (2014 年) (編集者) Camera-Based Document Analysis and Recognition - 4th International Workshop, CBDAR 2011, Revised Selected Papers, Springer (2012 年) (編集者) 他
研究専門分野：物体認識、物体検出、視覚障碍者支援

あらまし

k近傍探索とは、検索質問データが与えられたとき、それに最も近いk個のデータをデータベースから探し出す問題である。この問題は、時間を掛けさえすれば必ず解ける。しかし、データベースの大規模化や高次元化に伴い、計算時間が大きくなってしまう。この問題を高速に解くために、探索過程に近似を導入し、探索誤りを許容するのが近似k近傍探索である。本研究では、従来用いられている計算機(古典的コンピュータ)ではなく、量子コンピュータで動作するk近傍探索手法の量子アルゴリズムを提案する。提案手法は、データ数がNのとき、$O(\sqrt{kN})$の計算量で解を求めることができる。同じオーダーの計算量を持つ既存の量子アルゴリズムもあるが、提案手法は近似計算に適用できる可能性を持つ。

１．研究の目的

本研究では、データ数がNのデータベースDと検索質問データ（クエリ）xが与えられたとき、クエリに最も近いk個のデータを$O(\sqrt{kN})$の計算量で求めることができる量子アルゴリズム（量子コンピュータで動作するアルゴリズム）を提案する[1, 2]。この問題はk近傍探索と呼ばれ、時間を掛ければ必ず解けるタイプの問題である。とはいえ、データ数Nの増加や次元数の増加に伴い、大きな計算量が必要になるため、より小さな計算量で解くことができるアルゴリズムの開発が求められる。量子コンピュータは、従来のコンピュータ（古典的コンピュータと呼ばれる）よりも高速に実行できることが期待されており、k近傍探索においても同様の期待が持てる。

k近傍探索の既存の量子アルゴリズム[3]がある。このアルゴリズムの計算量は提案手法と同じ$O(\sqrt{kN})$であるが、実現方法が大きく異なる。詳細は後述するが、既存手法は同じ方法を繰り返すことによってクエリのk近傍を見つけるが、提案手法は 3 つの段階に分けてクエリのk近傍を見つける。特に、第 2 段階の終わりでは、k近傍よりも少し多いk'近傍が$O(\sqrt{N}\log k)$の計算量で求まっている。そのため、この段階で出力すれば、近似k近傍探索手法が実現できる可能性がある。

２．量子計算機と量子アルゴリズム

量子計算機は、量子イジングマシンモデルと量子ゲートモデルの 2 種類に大別できる。前者は量子トンネリング効果を利用して局所解を回避しながら特定の最適化関数を最適化するものであり、量子アニーリング方式がよく知られている。後者は、量子ビットの状態を別の状態に遷移できる量子ゲートを組み合わせることにより問題を解くことを目指すものである。使用する量子ゲートが増えると、ノイズの蓄積や、重ね合わせやもつれ状態が解消されやすくなり、精度が下がるという技術的課題があるが、前者と比較すると、より広い範囲の応用が期待できる。代表例として IBM Q が知られている。本研究では後者を使用する。

量子コンピュータが従来用いられている古典的コンピュータと大きく異なるのは、複数の状態を同時に保持して計算できることである。古典的コンピュータで

k近傍探索のための量子アルゴリズム

A Quantum Algorithm for k-Nearest Neighbor Search

は、データをその最小単位であるビットの組み合わせで表現する。1ビットは0または1の状態を保持できる。それに対して、量子コンピュータで用いる「量子ビット」は0と1が確率的に混じった状態を表現できるため、複数の状態を重ね合わせて、それらに対して一度の処理で計算できる。そのため、このような処理を上手く使えれば、高速な処理が期待できる。ただし、量子コンピュータ特有の状態の重ね合わせが可能なのは、状態の観測前だけであることに注意が必要である。すなわち、量子ビットの状態を観測する前は、0 と 1 が確率的に重なった状態であるが、観測した瞬間に、確率1で特定の状態に決まってしまう。この制約から、多くの量子アルゴリズムの出力は1つである。もし複数の出力が必要であれば、何度も量子ビットの状態を観測する必要があるため、その分計算量が増加してしまい、量子コンピュータを使う利点が失われる。本研究で実現するk近傍探索はk個の出力が必要なため、それをいかに実現するかが 1 つの工夫ポイントである。

量子ゲートモデルの量子アルゴリズムでは、その実行に要する量子ゲートの数を基準に計算量を求める方法と、オラクルと呼ばれる関数が何回呼ばれるかを基準に計算量を算出する方法がある。ここでオラクルは、中見や実装は未知であっても、望んだ答えを出してくれる関数のことを表す。本稿では、基本的にオラクルの呼び出し回数を基準に計算量を算出する。

2. 関連アルゴリズム

2. 1 既存のk近傍探索の量子アルゴリズム

既存のk近傍探索の量子アルゴリズムである Finding k-Minimum アルゴリズム [3]を紹介する。その準備として、Finding Minimum アルゴリズム [4]を紹介する。

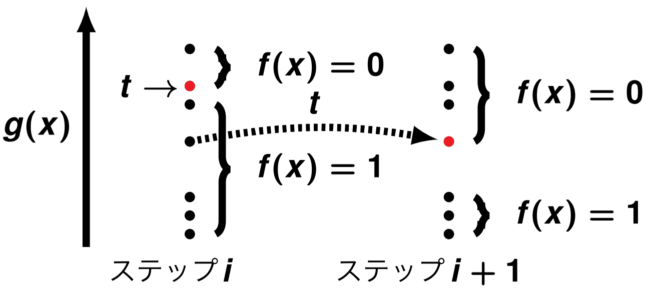

図1：Finding Minimum アルゴリズムの概要[4]

・Finding Minimum (FM) [4]

これは与えられたデータの中から最小値を持つものを発見するアルゴリズムである。図1に概念図を示す。この手法では、閾値tを設定して、その閾値よりも小さい値を持つデータにマークをする関数(オラクル)$f(x)$を用いる。すなわち、データの値が閾値tよりも小さいときに$f(x) = 1$となり、それ以外は$f(x) = 0$となる。このとき、$f(x) = 1$となったデータから 1 つ選んで新しい閾値にすれば、前の閾値よりも値が小さくなる。そこで、閾値tの初期値をランダムに設定してから、この処理を繰り返すことで最小値を発見できる。このアルゴリズムはオラクルを$O(\sqrt{N})$回呼び出すため、計算量は$O(\sqrt{N})$である。なお、$f(x)$の適用を何度も繰り返す処理は、Amplitude Amplification (AA) [5]と呼ばれるよく用いられる量子アルゴリズムである。これはよく知られた Grover の探索アルゴリズム[6]の一般化である。

この Finding Minimum アルゴリズムはデータの最小値を発見するものであり、クエリに近いデータを探すものでは無い。しかし、データとしてクエリとデータとの距離を与えることで、最近傍探索手法として用いることができる。

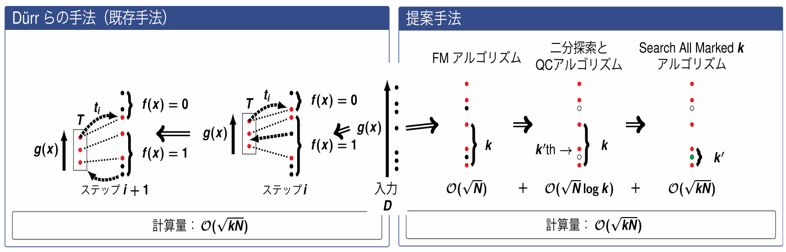

図２：Finding k-Minimum アルゴリズムの既存手法[3]と提案手法

・Finding k-Minimum (FkM) [3]

Finding Minimum アルゴリズムの出力をk個に拡張したものであり、既存のk近傍探索の量子アルゴリ

ズム[1]である。このアルゴリズムの処理の概要を図２の左側に示す。この図は、後で説明する提案手法と同じ入力に対して処理がどう違うかを表した図であり、既存手法は図の中央にあるデータが入力されて、処理が左側に進む。このアルゴリズムを一言で言えば、前述の FM アルゴリズムにおいて閾値をk個同時に用いるものである。すなわち、k個の閾値（図中では、その集合をTと表す）から１つを選び、それに対して FM アルゴリズムを実行する。この処理を繰り返し、前述のオラクル$f(x)$を$O(\sqrt{kN})$回適用することにより、k個の閾値はk個の最小値にたどり着くことが期待できる。なお､このアルゴリズムを効率よく実行するためには、毎回閾値集合Tの最大の閾値を選んで、より小さい閾値に更新するのが良い。しかし、これを実現しようとすれば、大きな計算量を要する閾値のソートが必要になるため、更新する閾値はランダムで選ぶと考えられる。また、閾値同士が同じ値にならないような工夫も必要である。

前述のように、量子アルゴリズムは１度の観測で１つの値しか出力できないため、通常多数の値を出力できない。実は、このアルゴリズムはk個の「閾値」を出力するものであるというのがミソである。つまり、閾値は別途外部メモリにでも保持しておけば、量子状態の観測は１回であっても、所望の解が得られるという考え方である。

２．２ 本研究で利用するアルゴリズム

本研究で利用する量子アルゴリズムを紹介する。

・Quantum Counting (QC) [7,8]

これは AA アルゴリズムにおいて$f(x)=1$であるデータの個数cを数える量子アルゴリズムである。Amplitude Estimation (AE) [5]と呼ばれる、データの$f(x)=1$である確率$\frac{c}{N}$を求めるアルゴリズムがある。これを利用して得られた確率にNを乗じることにより、cが得られる。AE の計算量が$O(\sqrt{N})$であるため、QC の計算量も$O(\sqrt{N})$である。

・Search All Marked k-indices (SAMkI) [9–13]

これは AA アルゴリズムにおいて$f(x)=1$となった要素を全て取り出すアルゴリズムである。いま、N個のデータのうちk個について$f(x)=1$であるとする。そして､k個のデータ全てを取り出したいとする。このとき、k個のデータのどれか １ つを取り出す処理に要する計算量は$O(\sqrt{N/k})$である。この処理の後、$N-1$個のデータが残り、そのうち$k-1$個が$f(x)=1$である。簡単のため、N個のデータが残り、そのうち$k-1$個が$f(x)=1$であるとしよう。このとき、$k-1$個のどれか １ つを取り出す処理に要する計算量は$O(\sqrt{N/(k-1)})$である｡これを繰り返すと､計算の過程は省略するが、

$$\sqrt{\frac{N}{k}}+\sqrt{\frac{N}{k-1}}+\cdots+\sqrt{\frac{N}{1}}=O(\sqrt{kN})$$

が得られる。したがって、SAMkI アルゴリズムの計算量は$O(\sqrt{kN})$である。

３．提案手法

提案するk近傍探索手法の処理の概要を図２の右側に示す。この図は処理が右側に進む。提案手法は、既存のk近傍探索手法が単一の処理（FM アルゴリズムのオラクル$f(x)$の呼び出しとそれに付随する処理）を繰り返すのとは異なり、大きく３つの処理を順に実行することでk近傍探索を実現する。

提案方法は、FM アルゴリズムを一度実行すると、その副産物として､「閾値の軌跡」が得られるというアイディアに基づく。ここで FM アルゴリズムを用いてデータの最小値を求める過程を見てみよう。最初の閾値はランダムに決まるため、何番目に小さいデータが閾値になっているのかがわからない。しかし、2 回目に実行すれば、少なくとも最初の閾値よりも小さい値になっていることが保証できる。FM アルゴリズムはこれを何度も繰り返すことで最小値を求める。このように、求まる閾値がどんどん小さくなり、降順に並んでいることがわかる。そして、閾値より値が小さいデ

[1] このアルゴリズムは、個々のデータが属性を持つという、より複雑な状況のために提案された手法であるが、特殊な場合として k 近傍探索にも適用可能である。

ータ数が単調に減少する。このことを利用すれば、閾値より値が小さいデータ数が大凡k個のところをうまく取れないか？というのが提案手法のアイディアである。この考え方に沿って、kよりも少し多いk'個のデータ（$k' \geq k$）を得ることを考える。FM アルゴリズムを実行すれば、i番目の閾値より値が小さいデータがk個以上であり、$i+1$番目の閾値より値が小さいデータがk個より少ないという状況が必ず存在する。ここでi番目の閾値より値が小さいデータの個数をk'とおく。一端k'個のデータが求まれば、k'個のデータに対して部分ソートを実行することで、値が小さいk個のデータが得られる。これが提案手法の概要である。

では、どのようにちょうど良い閾値（前述のi番目の閾値）を見つければ良いか。FM アルゴリズムを実行すれば、FkM のように、閾値の系列を記録しておくことができる。しかし、ある閾値より小さい値がいくつあるかは、自明では無い。これを実現するのが、前述の QC アルゴリズムである。ただ、QC アルゴリズムは一度の実行に$O(\sqrt{N})$の計算量を要するため、できるだけ実行回数を減らしたい。そこで二分探索を用いる。すなわち、二分探索で良い閾値を探しつつ、それが良い閾値かどうかは QC アルゴリズムで確認する。閾値の数は、オラクル$f(x)$の実行回数と等しいため、$O(\sqrt{N})$個である。閾値は整列済みで、降順に並んでいる。したがって、通常であれば、データ数$O(\sqrt{N})$のデータに対する二分探索の計算量は$O\left(\log\sqrt{N}\right) = O(\log N)$である。しかし、我々が欲しい$k$個のデータを求めるには、値が小さい方から最悪でも$k$個の閾値を探索すれば事足りる。したがって、二分探索に要する計算量は$O(\log N)$ではなく、$O(\log k)$である。QC アルゴリズムの計算量$O(\sqrt{N})$と掛け合わせることで、閾値を求めるのに$O(\sqrt{N}\log k)$の計算量が必要とわかる。

最後に、k'個のデータが得られた後の処理を考える。最初に FM アルゴリズムを実行しているので、値が一番小さいデータは既に判明している。そのため、残る$k'-1$個のデータの中から、値が小さい$k-1$個のデータを見つける必要がある。これには SAMkI アルゴリズムを用いる。簡単のために、$k'-1$ではなく、それより大きいN個のデータから$k-1$個のデータを見つける場合を考えると、この処理に要する計算量は$O\left(\sqrt{(k-1)N}\right) = O(\sqrt{kN})$であり、$O(\sqrt{kN})$の計算量で実現できることが分かる。

したがって、提案手法を構成する FM アルゴリズム、QC アルゴリズムを用いる二分探索、SAMkI アルゴリズムの計算量を合計すれば、

$$O(\sqrt{N}) + O(\sqrt{N}\log k) + O(\sqrt{kN}) = O(\sqrt{kN})$$

となり、提案手法の計算量は$O(\sqrt{kN})$となる。

４．まとめ

本研究では、$O(\sqrt{kN})$の計算量で実現できるk近傍探索の量子アルゴリズムを提案した。提案手法は既存手法と異なり、3 つの処理に分けてk近傍探索を実現する。そのため、途中で処理を止めれば、少ない計算量で近似k近傍探索が実現できる可能性がある。この検討が今後の課題である。

なお、本稿の図は共同研究者である宮本康平氏が[1] の発表に用いたポスター（http://www.m.cs.osakafu-u.ac.jp/publication_data/1681/AQIS2019_poster.pdf）から抜粋して修正したものである。

参考文献

[1] Kohei Miyamoto, Masakazu Iwamura and Koichi Kise, "A Quantum Algorithm for Finding k-Minima", Proc. 19th Asian Quantum Information Science Conference (AQIS2019), 2019.

[2] Kohei Miyamoto, Masakazu Iwamura and Koichi Kise, "A Quantum Algorithm for Finding k-Minima", arXiv preprint arXiv:1907.03315 [quant-ph], 2019.

[3] Christoph Dürr, Mark Heiligman, Peter Høyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. SIAM Journal on Computing, 35(6):1310–1328, 2006.

[4] Christoph Dürr and Peter Hyer. A quantum algorithm for finding the minimum. arXiv preprint quant-ph/9607014, 1996.

[5] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification

and estimation. In Jr. Samuel J. Lomonaco, editor, Quantum Computation and Quantum Information, volume 305, pages 53–74. American Mathematical Society, 2002.

[6] Lov K. Grover. A fast quantum mechanical algorithm for database search. In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pages 212–219, 1996.

[7] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum counting. In International Colloquium on Automata, Languages, and Programming, pages 820–831. Springer, 1998.

[8] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In Jr. Samuel J. Lomonaco, editor, Quantum Computation and Quantum Information, volume 305, pages 53–74. American Mathematical Society, 2002.

[9] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Quantum speed-up for unsupervised learning. Machine Learning, 90(2):261–287, 2013.

[10] Andris Ambainis. Quantum search algorithms. ACM SIGACT News, 35(2):22–35, 2004.

[11] Andris Ambainis. A new quantum lower bound method, with an application to strong direct product theorem for quantum search. arXiv preprint quantph/0508200, 2005.

[12] Hartmut Klauck, Robert Spalek, and Ronald De Wolf. Quantum and classical strong direct product theorems and optimal time-space tradeoffs. SIAM Journal on Computing, 36(5):1472–1493, 2007.

[13] Sebastian Dörn and Thomas Thierauf. A note on the search for k elements via quantum walk. Information Processing Letters, 110(22):975–978, 2010.

この研究は、平成２７年度ＳＣＡＴ研究助成の対象として採用され、平成２８～３０年度に実施されたものです。