Apriori アルゴリズムにおける完全準同型暗号を用いた秘密計算の並列分散処理の実装と評価

宇佐美文梨† 小口 正人†

† お茶の水女子大学 〒 112–8610 東京都文京区大塚 2–1–1
E-mail: †{g1320505,oguchi}@is.ocha.ac.jp

あらまし 近年，ビッグデータの利活用が推進され，知識獲得のために用いられるデータ量やアルゴリズム自体の計算量が爆発的に大きくなっており，個人や組織の所有する計算機上でマイニングを行うことが難しくなっている．そこで計算資源として注目されているクラウドは伝送路やサーバ上においてデータの盗聴や改竄のリスクがあり，個人情報など機密性の高いデータを扱う際にはプライバシを保護する技術が必要となる．本研究では，相関ルール抽出において広く知られている Apriori アルゴリズムに完全準同型暗号を用いた秘密計算へ並列分散計算を適用することで安全かつ高速なデータマイニング手法を提案する．

キーワード プライバシ保護データマイニング, 並列分散計算, アプリオリ・アルゴリズム, 完全準同型暗号, 秘密計算

Implementation and Evaluation of Parallel and Distributed Computing for Secure Multiparty Computation using Fully Homomorphic Encryption in Apriori Algorithm

Fumiri USAMI† and Masato OGUCHI†

† Ochanomizu University 2–1–1 Otsuka, Bunkyouku, Tokyo 112–8610 JAPAN
E-mail: †{g1320505,oguchi}@is.ocha.ac.jp

Key words Privacy-Preserving Data Mining, Parallel and Distributed Computing, Apriori Algorithm, Fully Homomorphic Encryption, Secure Computation

1. はじめに

大量のデータから新たな知識を得ることで今までになかったアプローチを得るビッグデータ・アナリティクスの需要が高まり，ビッグデータの利活用は世界的に進んでいる．具体的には，企業や病院などで蓄積されていた顧客情報やカルテなどを分析することで，新たな視点による社会問題解決や新規マーケット獲得の糸口となることが期待されている．加えて，近年のストレージや計算機の著しい性能向上により，ビッグデータからの知識発見に扱われるデータ量および計算量は爆発的に増大しつつある．その結果，個人の所有する計算機上で扱えなくなった膨大なデータやアルゴリズムを扱うために，クラウドを計算資源として用いる場面が多くなっている．

しかし，データや解析を外部に委託する際，委託先の事業者を完全に信頼することができなければ，情報漏洩の可能性が生じる．個人情報や機密性の高い情報を扱う場合，重大な情報セキュリティ上の問題となり得る．

高い機密性を有するデータから，適切なセキュリティ管理のもと知識を獲得するための技術の総称をプライバシ保護データマイニング（Privacy-Preserving Data Mining）という．これに用いられる匿名化技術には大きく分けて以下の 3 種類がある．

（1） 入力データプライバシ（Input Privacy）保護技術
e.g. k-匿名化（k-Anonymization）
（2） 出力データプライバシ（Output Privacy）保護技術
e.g. 摂動化（Perturbation）
（3） 秘密計算
e.g. 秘密分散, 秘匿計算

前の 2 つは，値を丸めたりダミーを加えたりすることでプライバシを保護する．計算量は比較的軽量でデータマイニングに向いているが，実データそのものを秘匿することはできず，正確なマイニング結果を得ることはできない．また，統計的手法を用いた解析に特化している．

一方，秘密計算はクライアントが委託先にデータを開示しないまま解析を委託するものである．大きく分けて，秘密分散と，

準同型暗号を用いる秘匿計算と呼ばれる方式の 2 つがある．

秘密分散 (Secure Multparty Computation) とは，解析対象となるデータを異なる事業者に分散させて委託し，ある一定数出力が得られれば復元可能となる方式である．

準同型暗号 (Homomorphic Encryption) を用いた方式では，RSA 暗号や ElGamal 暗号などの準同型性を持つ公開鍵暗号を用いている．準同型暗号では，加法や乗法において，暗号文を暗号化したまま計算できる性質がある．クライアントは準同型暗号によって暗号化したデータをクラウドに送り，クラウドで暗号化したまま処理を行い，クライアントに暗号化された出力を送る．クライアントは出力を復号することで求める結果を得る．この方法では，入出力データを秘匿でき，正確なマイニング結果を得られるため，一般的な関数計算もデータを秘匿したまま行うことができる．しかしながら計算量が膨大であり，実用的な時間内に処理を終えることは困難であった．

本研究では，完全準同型暗号によって秘匿された Apriori アルゴリズムによる相関ルール抽出に，Boost.MPI [6] を用いた並列分散処理を適用する．クラウド上への委託計算を想定した通信環境を構築することで，かねてよりの課題であった秘密計算の実行時間を短縮する手法を提案する．

2. 従来研究

2.1 完全準同型暗号とは

準同型暗号とは，暗号化したまま計算ができる暗号の総称である．完全とは，加法と乗法を備えており，任意の関数を構成できることを示す．すなわち，数学用語の環である．

まだ計算機が高価であり，大規模な計算施設を時間借りするタイムシェアリングシステムが商用サービスとなっていた 1970 年代後半に，プライバシを保護しながら委託計算を実現したいという需要に応えるため，Rivest，Adleman，Dertouzos が秘匿準同形性という概念を提唱したことに端を発し，研究がなされてきた [1]．RSA 暗号や Paillier 暗号など，加法や乗法のみの準同型暗号は 2000 年までに構成されていたが，完全準同型暗号が構成可能かどうかについては長らく未解決であった [2]．2009 年，Gentry が演算回数に制限の付いた Somewhat 準同型暗号 (Somewhat Homomorphic Encryption) に対し暗号文中に含まれるノイズを縮小する「ブートストラップ (bootstrapping)」という手法を導入したことによって，完全準同型暗号は具体的な構成法が与えられ，研究が一気に進展した．

準同型性をもつ鍵暗号は以下の 4 つのアルゴリズムから構成される [1] [3].

- $KeyGen(\lambda) \rightarrow (sk,pk)$: セキュリティパラメータ λ を入力として秘密鍵 sk と公開鍵 pk のペア (sk,pk) を返す．このとき秘密鍵が破られる確率は $1/2^{\lambda}$ となる．
- $Encrypt(pk,m) \rightarrow C$: 公開鍵 pk と平文 m を入力として暗号文 C を返す．
- $Decrypt(sk,C) \rightarrow m/\perp$: 秘密鍵 pk と暗号文 C を入力として平文 m またはシンボル $\perp$ を返す．
- $Evaluate(pk,P,(C_1,C_2,\cdots,C_t)) \rightarrow C$: 公開鍵 pk, t 個の入力を取る AND ゲートと XOR ゲートで構成された回路 P, 暗号文の組 $(C_1,C_2,\cdots,C_t)$ を入力として，新たな暗号文 C を返す．完全準同型暗号では，このとき，

$$\forall i \in \{1,\cdots,t\},\ c_i = Encrypt(pk,m_i)$$

ならば

$$Decrypt(sk,Evaluate(pk,P,(C_1,\cdots,C_t)))=P((m_1,\cdots,m_t))$$

となり，任意の P を正確に評価できる．

完全準同型暗号を用いれば，正確な計算結果が得られ，秘匿性の高い委託計算が実現できる一方，平文に対して暗号文が大きくなりすぎることや，暗号文のノイズを取り除くためのブートストラップのオーバーヘッドなどの問題があり，完全準同型暗号の実用化に向けた研究は盛んになっている．

2011 年，ブートストラップを用いずに完全準同型暗号を実現することで計算量を削減する BGV スキームが提案された [4]．BGV スキームの C++による実装である HElib は，2012 年の Smart と Vecrauteren によるベクトルの要素ごとの和積の準同型評価が可能なパッキング手法（SV パッキング）をはじめ，同年の Genty，Halevi，Smart による GHS 最適化など，さまざまな最適化手法が施されたオープンソースライブラリである [5]．本研究では，先行研究 [3] の実装で用いられた HElib によって実装を行っている．

2.2 Apriori アルゴリズムとは

Apriori はデータマイニングの中でも相関ルール抽出やバスケット解析に用いられる非常にポピュラーなアルゴリズムである [7]．例えば，ある商店において顧客ごとの商品購買履歴のリストがあるとすると，同時に購入される頻度の高い商品の組を知ることができる．商品の種類をアイテム（item），ある顧客が購入した商品の集合を表すアイテム集合（itemset）をトランザクション（transaction）または出現（occurrence）といい，あるアイテム集合がトランザクション中に現れている数を頻出度（frequency）またはサポート（support）という．多くのトランザクション中に含まれるアイテム集合のことを頻出アイテム集合（frequent itemset）あるいは単に頻出集合という．頻出アイテム集合は，閾値として最低サポート数（minimum support）以上の頻出度を持つアイテム集合のことを指す．すなわち，ある購買履歴のリストにおいて，同時に買われている回数が最低サポート数以上であるとき，同時に購入される頻度が高いとみなすことになる．ここではトランザクション中に出現する探索対象となるアイテム集合をパタンと呼ぶ．

ある頻出集合 S の部分集合 S' は必ず頻出集合 S の頻出度以上の頻出度を持っている．この単調性を利用すると，空集合から探索を開始し，探索する頻出集合の長さを 1 つずつ増やしていくことで任意の頻出集合を得られる．Apriori はこの探索を幅優先的に行うものを指す．深さ優先的に行うものはバックトラック法と呼ばれる．

深さ優先探索と比較したときの幅優先探索の特徴として，アルゴリズム中のループを一巡するときの探索順序は変わらず，探索リストの一番後ろに新しい探索対象が付け加わる．幅優先探索はメモリ消費が激しい一方スレッドセーフであり，並列化に向いていることが知られているため，本研究では Apriori を

用いた．

アルゴリズムの概要を以下に示す．

(1) 頻出アイテム集合の候補のパタン長（パタンに含まれるアイテム数）：pattern-length=1 とする．

(2) 判明している頻出アイテム集合のうち長さが pattern-length - 1 であるものの和集合から，パタン長が pattern-length となるパタンをすべて生成し，頻出アイテム集合の候補とする．

(3) 各候補ごとに，そのパタンが全トランザクション中に現れている回数（サポート）を数える．

(4) 各候補のサポートを最低サポート数と比較し，頻出パタンを決定する．頻出パタンが空集合だった場合，アルゴリズムは終了する．

(5) pattern-length をインクリメントして (2) に戻る．

例えば，ある商店で顧客が購入した商品のリストを集計した表 1 のようなトランザクションデータベースがあったとする．このとき訪れた顧客のうち半分以上が購入した商品 A,B,C,D の組み合わせを知りたい．全トランザクション数は 7 なので，最低サポート数を 4 とすると，Apriori を実行した結果は表 2 のようになる．

表 1 トランザクションデータベースの例

番号	アイテム	バイナリ表記
1	{A,B,C,D}	1,1,1,1
2	{A,B,C}	1,1,1,0
3	{A}	1,0,0,0
4	{A,B,D}	1,1,0,1
5	{B,C}	0,1,1,0
6	{B,C,D}	0,1,1,1
7	{A,B}	1,1,0,0

表 2 最低サポート数=4 のときの Apriori 実行例

Apriori 実行回数	頻出アイテム集合の候補	候補毎のサポート	頻出アイテム集合
1	{A},{B},{C},{D}	5,6,4,3	{A},{B},{C}
2	{A,B},{B,C},{C,A}	4,4,2	{A,B},{B,C}
3	{A,B,C}	2	{}

完全準同型暗号は，1 ビットの平文を対象として和積 (XOR,AND) の準同型評価を行う方式であるため，プライバシ保護データマイニングを想定して Apriori を実行する際は，表 1 のバイナリ表記のようなデータセットを通信に用いる．あるアイテムがトランザクション中に存在するときは 1, 存在しないときは 0 で表される．

2.3 関 連 研 究

Liu ら（2015）[8] は，完全準同型暗号を用いることで，強いプライバシ保護性を持ち，様々なアウトソーシングされたデータを異なるマイニングタスクに用いることができるような，安全な委託データマイニングを実現する手法を提案している．相関ルール抽出のアルゴリズムとして広く知られる Apriori を用いて膨大なデータから知識を得たいクライアントが，サーバに暗号化したトランザクションを預け、サポートの数え上げを委託するというものである．

これを応用した手法として，高橋ら（2016）は Liu らの用いた整数ベースのスキームの代わりにブートストラップなしで完全準同型暗号を構成する BGV スキームを採用した HElib を用いて Liu らの手法を実装し，複数の平文を一つの暗号文に暗号化する SV パッキングを用いることで，暗号文の数と計算量を削減して 430 倍の高速化と 94.7 %のメモリ使用量削減に貢献している [3]．

これらの研究は 1 対 1 のクライアント・サーバ通信であり，多数のノードを持つクラウド上への委託計算を実現するためには 1 対 N のクライアント・サーバ通信を想定する必要がある．

3. 提 案 手 法

高橋ら (2016) の提案手法を，広域分散計算を想定した 1 対 N のクライアント・サーバ通信に拡張する．クライアントは 1 個のサーバとのみ通信し，高橋ら（2016）の提案手法と同一の挙動をとる．サーバは，クライアントと通信を行う 1 個のマスタと，マスタからクエリを受信しサポートの数え上げを行うその他 N-1 個のワーカの 2 種類に分かれる．

プログラムの概要を以下に示す．

(1) クライアントはマスタとソケット通信を確立し，完全準同型暗号の計算に必要な context，公開鍵，トランザクションデータベースを暗号化した暗号文を送信する．

(2) マスタはクライアントから受信したデータをワーカらと共有し，各データを読み込む．

(3) クライアントは Apriori を開始する．頻出アイテム集合の候補リストを生成し，マスタへ送信する．

(4) マスタはクライアントから受信した頻出アイテム集合の候補リストを N 分割し，分割した候補リストをワーカに送信する．ワーカは候補リスト中の各パタンのサポートを数え上げ，結果をマスタへ送信する．

(5) マスタは各ワーカから受信したサポートのリストを集計し，結果をクライアントへ送信する．

(6) クライアントはサポートが最低サポート数以上を満たすパタンを頻出アイテム集合とし，それが空でない場合は pattern-length をインクリメントして (3) に戻る．空ならば今までの頻出アイテム集合を結果として得て，Apriori を終了する．

(7) クライアントはマスタに空の候補を送信する．マスタはワーカらとそれを共有し，ワーカはプログラムを終了する．マスタとクライアントはソケット通信を閉じ，プログラムを終了する．

並列分散処理においては，Boost.MPI Library [6] を採用した．分散メモリの並列計算におけるメッセージ通信のためのライブラリ規格である MPI （Massage Passing Interface）を HElib が実装されている C++を用いて平易に扱うためである．

4. 実　　験

4.1 実験環境

本研究で想定するクラウド環境を再現するため，早稲田大学山名研究室とお茶の水女子大学小口研究室を接続して VPN 環境を構築した．簡単な構成図を図 1 に示す．サーバ hpcs01-04 及び enigma02 は同一種類である．例として hpcs01 の性能を表 3 に示した．表には示されていないが，CPU は 2 台搭載されている．提案手法におけるクライアントは早稲田大学側のサーバが担当し，マスタはお茶の水女子大学のサーバ hpcs01，ワーカは hpcs02–04 が担当する．

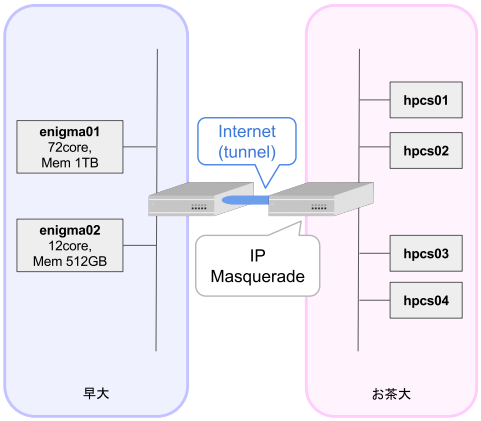

図 1　実験環境（物理サーバ）

表 3　hpcs01 の性能

OS	Linux 2.6.32-431.23.3.el6.x86_64
CPU	Intel(R) Xeon(R) CPU E5-2643 v3
	@3.40GHz 6 Core / 12 Thread
Memory	512GB
Disk	8TB SATA HDD (4 × 2TB)
	1.9TB SATA SSD (4 × 480GB)

表 4　SV パッキング適用有無別の Apriori 計測時間（秒）

isPacking	計測地点	総実行時間	Apriori 実行時間
True	client	49.2597	22.6931
True	server	49.2594	22.4361
False	client	111.476	75.0264
False	server	111.475	64.4767

4.2 関連研究の再現実験

提案手法は実装中のため，高橋ら（2016）[3] の再現実験を行った．高橋ら（2016）が用いたデータセットと同一の IBM Almaden Quest research group の生成器を用いて得られた人工データセット（アイテム数 50，トランザクション数 100）を用い，最低サポート数=10 の場合で実験を行った．

サーバに hpcs01，クライアントに hpcs02 を用いてパッキングの有無での実行時間の変化を調べたところ，表 4 のようになった．SV パッキングを行うことで通信する暗号文のデータおよび積の実行回数が約 1/全アイテム数 となり，結果として総時間は約 1/2，Apriori の実行時間は 1/3 以上に短縮されている．

5. まとめと今後の課題

本研究では，並列分散処理を用いて既存の提案手法を拡張し，完全準同型暗号を用いた秘密計算の実行速度を高速化することを目指した．今後の課題として，MPI を用いた並列分散処理による委託データマイニングを実装することで提案システムを完成させ，Liu ら [8] や高橋ら [3] との性能比較によって本研究の有効性を評価する必要がある．その後は，並列分散処理における従来の高速化手法として知られるキャッシュを利用したシステムを検討し，各種最適化を施した上でさらなる高速化につなげていきたい．

謝　　辞

本研究の一部は，JST（科学技術振興機構）戦略的創造研究推進事業 CREST「ビッグデータ統合利活用のための次世代基盤技術の創出・体系化」の支援を受けたものである．本研究を進めるにあたり，多くの助言を賜りました早稲田大学山名研究室の馬屋原さん，今林さんに深く感謝いたします．

文　　献

[1] 草川, ”完全準同型暗号の概要,” 電子情報通信学会誌 Vol. 99, No. 12, pp. 1151–1158, 2016.

[2] 縫田, ”完全準同形暗号の最近の研究動向,” 電子情報通信学会誌 Vol. 99, No. 12, pp. 1176–1183, 2016.

[3] 高橋, 石巻, 山名, ”SV パッキングによる完全準同型暗号を用いた安全な委託 Apriori 高速化,” DEIM Forum 2016 F8–6, 2016.

[4] Z.Brakerski, C.Gentry, V.Vaikuntanathan, ”Fully Homomorphic Encryption without Bootstrapping,” 3rd ITCS, Cryptology ePrint Archive, Report 2011/277, 2011, http://eprint.iacr.org/2011/277

[5] HElib, https://github.com/shaih/HElib

[6] Boost.MPI Library, 1.62.0, http://www.boost.org/ doc/libs/1_62_0/doc/html/mpi.html

[7] 宇野, 有村, ”頻出パターン発見アルゴリズム入門 – アイテム集合からグラフまで – ,” 人工知能学会第 22 回全国大会 (JSAI 2008), 3M1–1, 2008.

[8] Liu J., Li J., Xu S., Fung B.C., ”Secure outsourced frequent pattern mining by fully homomorphic encryption,” Big Data Analytics and Knowledge Discovery, pp. 70–81, Springer (2015)