完全準同型暗号を用いた Apriori アルゴリズムの並列分散計算による高速化手法の検討

宇佐美 文梨　(指導教員：小口 正人)

1　はじめに

近年，ビッグデータの利活用が進み，またストレージや計算機の性能が著しく向上しているために，知識発見に用いられるデータ量およびアルゴリズム自体の計算量は爆発的に増大しつつある．その結果，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，個人情報や機密性の高い情報を扱う場合，第三者のクラウド事業者を完全に信頼することができなければ，情報漏洩が伝送路上だけでなくクラウド上で起こる可能性があり，対策を講じる必要がある．

高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保護技術
   e.g. k-匿名化（k-Anonymization）
2. 出力データプライバシ保護技術
   e.g. 摂動化（Perturbation）
3. 秘密計算（Multi Party Computation）

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ため，データそのものを秘匿することはできず，得られるマイニング結果も正確ではない．一方，秘密計算はデータを暗号化することで入出力を秘匿でき，暗号化したまま計算を行うことで正確なマイニング結果を得ることができる．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，和と積において準同型性を持つ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 [1] 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．

1.1　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．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)」という手法を紹介したことによって，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 [3]．

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]．高橋ら (2016) では HElib を用いて SV パッキングによる 1 対 1 のクライアント・サーバ通信の委託計算の実装を行い，高速化に成功している [6]．

1.2　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]．例えば顧客ごとの商品購買履歴リストがあるとすると，同時に購入される頻度の高い集合（頻出アイテム集合）を知ることができる．

アルゴリズムの概要を以下に示す．商品の種類をアイテム，ある顧客が購入したアイテムのリストをトランザクションといい，トランザクション中に同時に出現するアイテムのリストをパタンという．

(1) 頻出アイテム集合の候補リスト中のパタン長（パタンに含まれるアイテム数）：pattern-length=1 とする．
(2) アイテム集合または判明している頻出アイテム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を数える．これを支持度（support）という．
(4) 各候補の支持度を最低サポート数（minimum support）と比較し，頻出パタンを決定する．頻出パタンがない場合，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．

2　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支持度の数え上げを行うその他 N-1 個のワーカに分かれる．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，全トランザクション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 context と公開鍵をワーカらと共有する．マスタは受信した暗号文を N 分割し，各ワーカにそれらを送信する．各ワーカが持つ暗号文のサイズは，クライアントが送信した暗号文の 1/N となっている．

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

(4) マスタは頻出アイテム集合の候補リストをワーカらと共有する．ワーカは候補リスト中の各パタンの支持度を数え上げ，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支持度のリストを足し上げ，結果をクライアントへ送信する．

(6) クライアントは支持度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閉じる．マスタとクライアントはソケット通信を終了する．

並列分散処理においては，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，Boost.MPI Library[1] を採用した．

3 実験

3.1 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 は同一の種類である．例として hpcs01 の性能を表 1 に示した．提案手法での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 hpcs01，ワーカは hpcs02-04 が担当する．

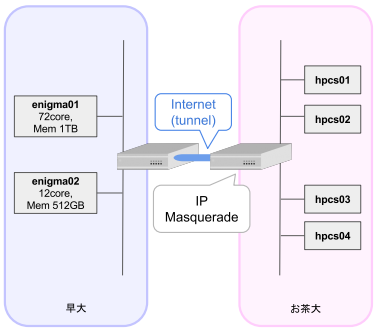

図 1: 実験環境（物理サーバ）

表 1: 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)`

3.2 関連研究の再現実験

提案手法は実装中のため，高橋ら（2016）[6] 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 T10I6N50D5kL1k（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 2 のようになった．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．

表 2: SV パッキングの有無毎の 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 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 [6] 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 CREST 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皆様に深く感謝いたします．

参考文献

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

[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] 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.

[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] 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.

[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)