2005－EVA－13（4）
2005／6／24

プライバシーを保護した分散データマイニングアルゴリズムの提案

浦邊 信太郎　王 家宏　児玉 英一郎　高田 豊雄
岩手県立大学 ソフトウェア情報学研究科

ネットワーク上に分散されたデータベースから共通の相関ルールを発見する分散データマイニングにおいて，重要な問題の 1 つにプライバシーの保護がある．例えば，異なるクレジットカード会社同士で共通のカード詐欺のパターンを抽出する際，データマイニングを行うためには自社のカード詐欺事例を他社へ伝える必要があるが，自社の契約内容などを知られることは不利益につながる．このような問題を解決するため，本研究では，他サイトに自分のデータを知られることなくデータマイニングを行い，結果のみを共有できる分散データマイニングアルゴリズムを提案する．

A Collusion-Resistant Approach to Privacy-Preserving Distributed Data Mining

Shintaro Urabe　Jiahong Wang　Eichiro Kodama　Toyoo Takata
Faculty of Software and Information Science, Iwate Prefectural University

When data mining is performed across more than one site, i.e., when distributed data mining is conducted, the privacy of any participating site must be well preserved. For example, several credit card agencies such as Banks would cooperate to mine the union of their databases to detect patterns matching credit card frauds. It is obvious that the precondition is that no privacy of the individual agency should be leaked out to others. In this paper we address the subject of privacy-preserving data mining, and propose a collusion-resistant approach to distributed privacy-preserving data mining.

1. はじめに

近年，企業で取り扱う様々なデータが電子化され，データベースの利用がより一般的になっている．それに伴い，ネットワーク上に分散されたデータベースからビジネスに活用できる有用な情報を取り出す技術として分散データマイニングが注目されている．既存の分散データマイニングアルゴリズムに FDM[1] がある．しかし，FDM は自サイトのデータをそのまま他のサイトへ送信する必要があり，プライバシーが漏洩する可能性がある．そのため，分散データマイニングにおけるプライバシー保護のための手法が提案されている．ここでプライバシー保護とは，あるサイトの持つ「トランザクション」及び「興味深い相関ルール」が他のサイトに漏洩するのを防ぐことと位置付ける．SFDM[2] は暗号化を用いてデータを秘匿し，プライバシーを保護する手法である．しかし，暗号化を用いるため計算機に大きな処理負荷がかかってしまうという問題がある．SDDM[3] は乱数を用いることによってこの問題を改善している．しかし，SDDM は高々4 つのサイトが結託することによって情報が漏洩してしまうという問題がある．そこで本研究ではサイト間の結託に対する耐性を向上させる分散データマイニングアルゴリズムを提案する．本提案アルゴリズムでは乱数を複数用いることによって情報を集計するルートを複数生成し，サイト間の結託に対する耐性を向上させる．本稿ではプライバシーを保護した分散データマイニングアルゴリズムを示すとともに，本提案アルゴリズムが既存のアルゴリズムと比較してサイト間の結託に対する耐性が高いことを示した．

2. システムモデル

本研究で対象とするのは，ネットワークに接続された複数のコンピュータが同じ項目のデータベースを保持し，それぞれでデータマイニングを行い，結果を統合するシステムである．このデータマイニングから導き出される「X を買う人は Y も一緒に買う」のような項目間の関係をアソシエーションルールと呼び，$X \Rightarrow Y$ と表す．アソシエーションルールはサポート (support) と確信度 (confidence) という二つの評価尺度によってその有効性が計られる．サポートはデータベースの中でそのアソシエーションルール $X \Rightarrow Y$ の出現するトランザクションの存在する割合を表し，確信度は X を含むトランザクションが Y も含む条件付き確率を表す．なお，ここで言うトランザクションとは顧客が購入した商品の履歴等であり，データベースとはトランザクションの集合である．データマイニングの利用者は，サポートと確信度の「下限値」（ユーザが設定した値）を超える「興味深いアソシエーションルール」だけを必要とする場合を考える．データマイニングを行う場合，確信度は全項目のサポートを求めることで計算が行える．そこ

で, サポートの下限値を min_sup, データマイニングを行うサイトの数を m とし, i 番目のサイトでアソシエーションルール x の出現するトランザクションの数を X_i, データベースのサイズを d_i, とすると以下の式が成り立ち, かつ確信度の条件を満たす x は興味深いアソシエーションルールとなる.

$$\sum_{i=1}^{m}(X_i - \texttt{min\_sup} * d_i) \geqq 0$$

3. 提案するアルゴリズム

本研究で提案するアルゴリズムの目的は, 全サイトでサポートを集計する際に各サイトがそれぞれ発信するサポートを秘匿することである. そこで, 各サイトではサポートを発信する際に乱数を付与することで秘匿を行う. 続いて, 各サイトで付与した乱数を集計し, 乱数が付与されたサポートの合計から乱数の合計を減算することによってサポートの合計を得る. 以上の手順を用いることによって, 各サイトで発信したサポートを秘匿しつつサポートの合計のみを得ることができる.

具体的なデータマイニングアルゴリズムを以下に示す.

Algorithm: データ提供者を秘匿することによりプライバシー漏洩を防ぐ分散データマイニング

Input: サポートと確信度の下限値, 各サイトにあるデータベースからなる集合

Output: 全ての興味深いアソシエーションルール

Step1: 起点となるサイトが候補アイテムセットを全サイトに送信する.

Step2: 各サイトで候補アイテムセットのサポートを計算する.

Step3: 全体で候補アイテムセットのサポートを秘匿したまま集計する.

- Step3.1: 各サイトで n 個の乱数 $Y_k(k = 1, 2, ..., n)$ を生成した後, 求めたサポートにそれらの乱数の合計を加える.
- Step3.2: Step3.1 で求めた値及び n 個の乱数を集計するルートをそれぞれ決定する. (3.1 節参照)
- Step3.3: Step3.1 で求めた値及び n 個の乱数を Step3.3 で決定したルートでそれぞれ集計する. (3.2 節参照)
- Step3.4: 乱数とサポートの集計結果から乱数の集計結果を減算し, サポートの合計を求める.

Step4: サポートの下限値を上回る候補から確信度の下限値を上回る候補を選択し, 興味深いアソシエーションルールとして出力する.

上記のアルゴリズムを使用することにより, サイトのサポートを知られること無くデータマイニングを行うことができる. 実際にサポートを秘匿しながら集計を行っているのは Step3 である. Step3 において, 乱数が付与されたサポートや乱数は各サイトを辿って集計される. そこで集計作業に先立ち, まず各値を集計するルートを決定する必要がある. このルートを決定するアルゴリズムを 3.1 節で説明する. 続いて, 決定されたルート上で各サイトがデータをやり取りし, 集計を行う. しかし, ただ決定されたルート上で, 各サイトを 1 つ 1 つ巡回して集計を行うのは非効率である. そこで, 各サイトでデータを同時に発生させ, 効率的に集計を行うアルゴリズムを 3.2 節で説明する.

3.1 ルート決定アルゴリズム

データを集計する際に使用される複数のルートはいくつかの条件の下に決定される. まず, 全てのルートは互いに同じ経路を通ってはならない. 例えば, ある乱数を集計するルート上に $Site1 \rightarrow Site3 \rightarrow Site5$ という経路があったとすると, 別の乱数を集計するルートには同じ経路が存在してはならない. これは, 決定されたルート数で最大の結託耐性を確保するためである. また, 各ルートは全てのサイトを通過して起点となるサイトへ戻らなければならない. これは, 全てのサイトで同じ結託耐性を確保するためである. しかし, この条件はある 1 つのデータの集計を対象としたもので, 1 つのデータの集計に複数のルートが選択されることもありうる. 例えば, サイト数が 8 の時, $Site1 \rightarrow Site3 \rightarrow Site5 \rightarrow Site7 \rightarrow Site1$ というルートと, $Site2 \rightarrow Site4 \rightarrow Site6 \rightarrow Site8 \rightarrow Site2$ というルートを組み合わせて 1 つのデータを集計することも可能である.

前述した条件を満足するルート決定アルゴリズムを示す. サイト数を m, サイトの集合を S とし, 各サイトを s_i $(i \leq m,\ s_i \in S)$ とする. 決定されるルート数を n とし, サポートや乱数等, ある 1 つのデータを集計する作業をセッション (Ses) と定義する. 乱数を加算したサポートを集計する作業を Ses_1, t 個目 $(1 \leq t \leq n)$ の乱数を集計する作業を Ses_{t+1} と表す. Ses_k において決定されるルートを $R_{k,j}$ $(k \leq n,\ j \leq gcm(m,k)$ $(a$ と b の最大公約数を $gcm(a,b)$ と表す)) とする. $R_{k,j}$ はサイト s_i の順序集合で構成される. また, ここでは起点サイトを s_1 とする. 全てのセッション Ses_1〜Ses_{n+1} について, 以下のアルゴリズムを適用する.

Algorithm: Ses_k においてデータを集計するルートを決定するアルゴリズム

begin: Ses_k
  $g = gcm(m, k)$
  if $g = 1$ **then**
    $x \leftarrow 1$
    for $i \leftarrow 1$ **to** m **do**
      $R_{k,1}$ に s_x $(s_x \in S\ |_{x=x+k})$ を追加する.
  if $g > 1$ **then**
    for $j \leftarrow 1$ **to** g **do**
      $x \leftarrow j$
      $d \leftarrow m \div g$
      for $i \leftarrow 1$ **to** d **do**
        $R_{k,j}$ に $s_x(s_x \in S\ |_{x=x+k})$ を追加する.
  return $R_{k,j}$ $(j = 1, 2, ..., g)$

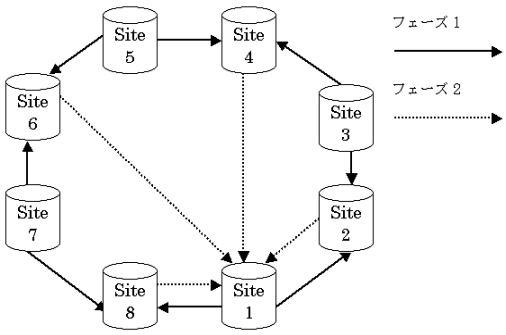

図 1: 提案するアルゴリズムの通信動作を示す例

end: Ses_k

このアルゴリズムを適用することにより, 前述した条件を満たす複数のルートを決定することができる. 基本的な考え方は, Ses_1 では $s_1 \rightarrow s_2 \rightarrow s_3$ と 1 個ごとに, Ses_k では $s_1 \rightarrow s_{k+1} \rightarrow s_{2k+1}$ と k 個のサイトごとにルートに追加していくというものである. また, k が m の約数の時, k 個ごとにサイトを追加すると全てのサイトを通らないまま起点サイトへ戻るルートができてしまうため, 全てのサイトを m/k 個ずつのグループに分け, それぞれのグループで新たにルートを決定する.

3.2 データの通信方法を決定するアルゴリズム

複数のメッセージを同時に送受信させるための各サイトの動作を決定するアルゴリズムについて述べる. 各サイトの動作を決定するにあたり, 動作の種類として前後のサイトからデータを受信する動作を「データ収集」, 前後のサイトへデータを送信する動作を「データ発生」と呼ぶこととする. データ発生で送信する 2 つのデータは, 自分の送信すべきデータをランダムに 2 つに分けたものとする. まず, あるルート上で通過するサイトについて起点サイトから順に, データ発生, データ収集と交互に行う (フェーズ 1). データを送信するサイト, 受信するサイトは互いに隣同士になる必要があるためである. 次に, これらのデータを合計するため, データ収集を行ったサイトは前後から受信した 2 つのデータに自分のデータを加え, 起点サイトへ送信する (フェーズ 2). 図 1 はサイト数が 8 の時, 乱数が加算されたサポートを集計するルート上で, 上記の動作を行った際のデータの流れを示した図である.

図 1 から, フェーズ 1 とフェーズ 2 の各通信はそれぞれ同時に発生可能であることが分かる. 本提案アルゴリズムを適用することにより, サイト数にかかわらず 2 つのフェーズで集計することが可能となる.

上記の動作を決定するアルゴリズムを以下に示す.

あるデータを集計するルートを R とし, R 上で通過するサイト数を o とする. R 上で通過する各サイトを r_i $(i \leq o)$ とし, r_i が Ses_k において渡すデータを $V_{i,k}$ とする. 全てのセッション Ses_1〜Ses_{n+1} について, 以下のアルゴリズムを適用する.

Algorithm: Ses_k における各サイトの動作を決定するアルゴリズム
begin: Ses_k
 for each r_i **do**
  if $i \ mod \ 2 = 1$ **then**
   渡すデータ $V_{k,i}$ を 2 つに分ける.
   ここで分けられた 2 つのデータをそれぞれ
   $V_{i,k,1}$, $V_{i,k,2}$ とすると,
   $V_{i,k} = V_{i,k,1} + V_{i,k,2}$
   が成り立つものとする.
   $V_{i,k,1}$, $V_{i,k,2}$ を r_{i+1}, r_{i-1} にそれぞれ送信する.
  if $i \ mod \ 2 = 0$ **then**
   r_{i+1}, r_{i-1} から渡されたデータ $(V_{i+1,k,2}, V_{i-1,k,1})$
   と自分が渡すデータ $(V_{i,k})$ を合計し,
   起点サイト r_1 に送信する.
end: Ses_k

4. 性能評価

本提案の有効性を確認するため, 提案アルゴリズムを分析, 実装し, 性能評価を行った. 4.1 節では, 提案したアルゴリズムの結託耐性, メッセージ数を導出し, 先行研究との比較を行った結果を示す. 4.2 節では, 提案したアルゴリズムを実装し, 実行時間を計測した結果を示す.

4.1 分析による評価

4.1.1 指標の定義

分析による比較ではその指標として結託耐性とメッセージ数を用いる. 結託耐性とは, あるサイトのプライバシーに関わりのあるデータを得るために結託する必要があるサイトの数から 1 を引いたものである. つまり, 何個のサイトの結託まで耐えられるかを表した値である. メッセージ数とは, データマイニングを行う際に, サイトから別のサイトへ渡されるメッセージの数である. これはさらに以下の 2 つのケースに分ける.

ケース 1: 全ての通信で発生するメッセージの総数

ケース 2: 同時発生するメッセージを 1 つとしてカウントする場合のメッセージ数

4.1.2 アルゴリズムの分析

続いて, 本提案アルゴリズムがどのようにして結託耐性を確保しているかについて述べる.

図 2 はサイト数が 8 の時, 本提案アルゴリズムを用いてサポート集計を行った際のデータの流れを示した図である. 各矢印は乱数が加算されたサポートや乱数等の通信を表す.

例として, Site3 の結託耐性を検証する. 図 2 より, 以下のことが分かる.

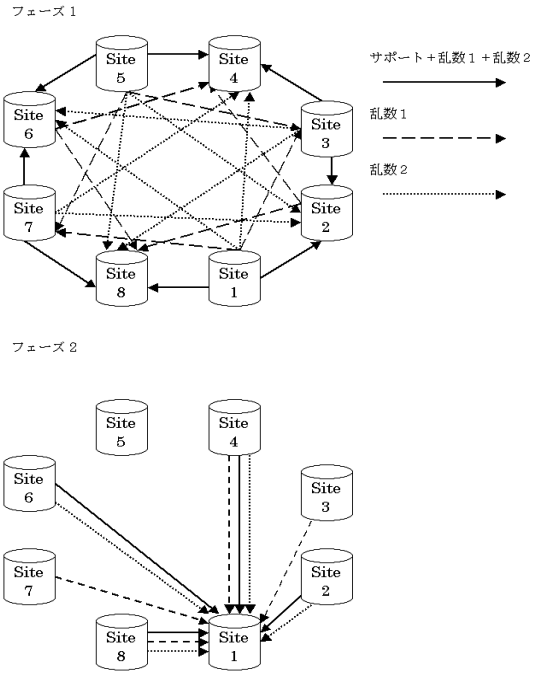

図 2: 提案するアルゴリズムのデータの流れ

- Site3 の乱数が加算されたサポートを得るためには Site2 と Site4 の結託が必要
- Site3 の乱数 1 を得るためには Site1 と Site5 の結託が必要
- Site3 の乱数 2 を得るためには Site6 と Site8 の結託が必要

よって, Site3 のサポートを得るためには Site1, 2, 4, 5, 6, 8 の 6 つのサイトの結託が必要であり, Site3 の結託耐性は 5 となる. また, フェーズ 1 の各通信は同時に発生可能で, それらの通信が完了した後, フェーズ 2 の各通信を同時に発生させることができる.

4.1.3 結託耐性による比較

図 3 は, 本研究で提案したアルゴリズム, SDDM[3], SFDM[2], Crowds[4] の 4 つの方法について結託耐性の比較を行った結果を示したグラフである. SFDM を用いた場合は結託耐性は 1 以下, SDDM はサイト数が 5 以上の時に常に 3 となる. これらに対し, 本提案アルゴリズムを用いた場合の結託耐性はサイト数の増加につれて高くなることが示された. これは, サイト数の増加に伴い集計ルート数も増やすことによって, 結託耐性を向上させることが可能となったためである.

4.1.4 メッセージ数 (ケース 1) による比較

図 4 は, データマイニングを行う際に発生する総メッセージ数 (ケース 1) を比較した結果を示したグラフである. 本提案アルゴリズムを用いた場合, SDDM, SFDM に比べて発生するメッセージ数が多

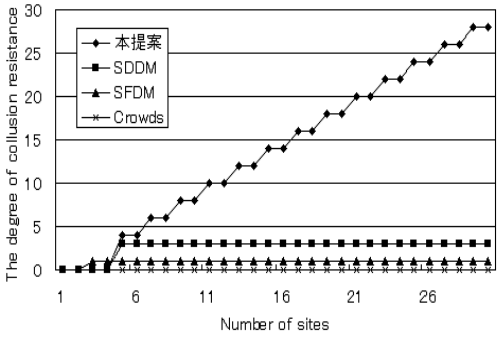

図 3: 本提案手法と既存のアルゴリズムの結託耐性による比較

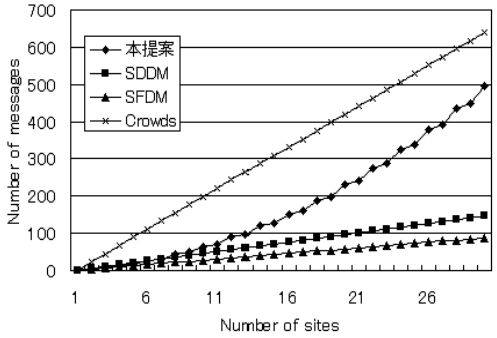

図 4: 本提案手法と既存のアルゴリズムのメッセージ数による比較 (ケース 1)

くなる. これは, SDDM に比べ複数の乱数とそれを集めるルートを用いたことにより, データの集計作業が増えたためである. また, 結託耐性の確保のため, サイト数の増加に伴い集計ルート数も増やしているため, サイト数に応じて高くなっている. しかし, この問題については, 3.1 節で提案したメッセージを同時に送受信させる手法を用いているため, 実際に処理時間に与える影響は軽減されている.

4.1.5 メッセージ数 (ケース 2) による比較

図 5 は, データマイニングを行う際, 同時に発生するメッセージを 1 つとして見た場合の総メッセージ数 (ケース 2) を比較した結果を示したグラフである. 本提案アルゴリズムは 4 つのアルゴリズムの中で最も少なく, サイト数が増えても常に 3 と変わらない値を示した. これは, SDDM や SFDM と異なり, サポートを集める際にサイトを巡回しないためである. よって, 多くのメッセージが同時に発生することを許容できる環境であれば本提案アルゴリズムが最も有用であると考えられる.

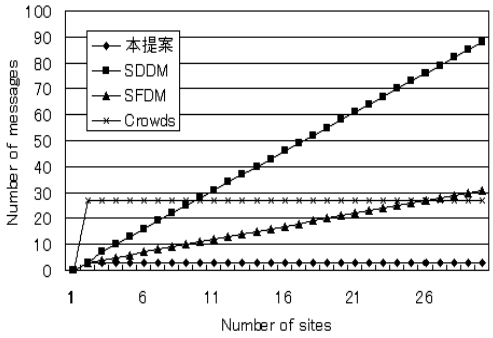

図 5: 本提案手法と既存のアルゴリズムのメッセージ数による比較 (ケース 2)

表 1: 実験環境

開発言語	java1.4.2_03
OS	Solaris 9
メモリ	512MByte

4.2 実装による評価

本提案アルゴリズムを実装し, 動作時間の比較を行った. 実装はデータマイニング実行部と通信部を作成した. 通信部は Socket を用いて他の全てのサイトと直接コネクションを張り, データマイニング実行部の要求に応じてデータの送受信を行う. データマイニング実行部は, Apriori[5] のサポートをカウントする部分で通信部を呼び出し, 各サイトのサポートを集計する操作を付け加えた. サポートを集計する操作の中に本提案アルゴリズムを実装することで, プライバシーの保護を行う. また, 各サイトのデータベースとして最大項目数 100 のトランザクション 10000 件を記述したテキストファイルを用いた. 実験は表 1 に示す条件のもと, サイト数が 5 及び 6 の場合について実行時間を計測した. また, 比較として SDDM 及び FDM についても同様の条件で実験を行った. 尚, 全ての実装は最適化を行っていない. 実験結果を表 2 に示す.

表 2: 実行時間の比較

アルゴリズム	サイト数 5	サイト数 6
FDM	111sec	111sec
SDDM	165sec	170sec
本提案	133sec	135sec

SDDM と比較し本提案アルゴリズムの実行時間は平均で約 20%短縮した. これは, SDDM ではサポートや乱数の集計の際, 各サイトを巡回してデータを集計しているのに対し, 本提案手法では 3.2 節で述べたアルゴリズムを用いることによって, 各サイトでデータを同時に発生させて集計しているため, そのぶん通信時間が短縮されたと考えられる. また, FDM と比較し平均で約 20.7%増加した. FDM ではローカルで発見された頻出アイテムセットを秘匿せず全サイトへブロードキャストするため, 本提案や SDDM が行う乱数を集計する作業について通信時間の差が表れたと考えられる.

5. まとめ

本研究では, 既存のアルゴリズムよりも結託耐性の高い分散データマイニングアルゴリズムを提案した. さらに, 本提案アルゴリズムに対し分析及び実験による評価を行い, その有用性を検証した. その結果, 本提案アルゴリズムはサイト数の増加に応じて結託耐性を向上させることが可能になり, 先行研究との比較において, 既存のアルゴリズムに比べ極めて高い結託耐性を持つことが示された. 一方, 本提案アルゴリズムは SDDM, SFDM より多くのメッセージが発生する. しかし, メッセージの多くは同時に通信可能で, 実験により, SDDM よりも短い時間で実行できること, サイト数の増加による実行時間への影響を軽減できることが示された. 以上のことから, 本提案アルゴリズムを用いることで, より安全で効率的な分散データマイニングを行うことが可能となった. 今後は, より詳細な性能評価を行い, 本提案の有用性を実証していく予定である.

参考文献

[1] D. W. -L. Cheung, J. Han, V. Ng, A. W. -C. Fu, and Y. Fu, A Fast Destributed Algorithm for Mining Association Rules, PDIS '96, pp.31-42, 1996.

[2] M. Kantarcioglu and C. Clifton, Privacy-preserving Distributed Mining of Association Rules on Horizontally Partitioned Data, IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.9, pp.1026-1037, 2004.

[3] T. Fukazawa, J. Wang, T. Takata, and M. Miyazaki, An Effective Distributed Privacy-Preserving Data Mining Algorithm, Fifth International Conference on Intelligent Data Engineering and Automated Learning '04. pp.320-325, 2004.

[4] M. K. Reiter and A. D. Rubin, Crowds: Anonymity for Web Transactions, ACM Transactions on Information and System Security, Vol.1, No.1, pp.66-92, 1998.

[5] R. Agrawal and R. Srikant, Fast algorithms for mining association rules, in Proc. the 20th International Conference on Very Large Data Bases, pp.487-499, 1994.