ユーザ存在の特定を困難にした分散匿名化の提案
——2診療機関のレセプトデータを用いた有効性の評価——

竹之内隆夫[†,††]　川村　隆浩[††]　大須賀昭彦[††]

Distributed Anonymization Method with Hiding the Presence of Individuals

Takao TAKENOUCHI[†,††], Takahiro KAWAMURA[††], and Akihiko OHSUGA[††]

あらまし　複数機関が保持するユーザのパーソナル情報を結合・分析し，新たな知見を得ることが期待されている．例えば，医学研究のために複数医療機関が保持する医療情報を結合・分析することが期待されている．一方，パーソナル情報は個人のプライバシーに関する情報であるため，他の機関へ全開示した結合はできない．また，結合したパーソナル情報の組合せからユーザが特定されるおそれもある．そのため，パーソナル情報が含まれたテーブルを必要最小限の開示にとどめながら結合し，個人が特定されない形に加工した結合匿名テーブルを生成・開示する分散匿名化手法が注目されている．しかし既存手法では，双方の機関のユーザ集合が一致しない場合に，ユーザのパーソナル情報がその機関に保持されているか否かというユーザ存在が他方の機関に漏えいする問題があった．そこで本論文では，ユーザ存在を隠ぺいしつつ結合匿名テーブルを生成・開示する分散匿名化手法を提案する．そして，提案手法の計算量・通信量の評価と実際のレセプトデータを用いた有効性評価を行い，ユーザ存在を隠ぺいしながらも相対誤差 15%以下でデータ分析な結合匿名テーブルが生成可能であることを確かめた．

キーワード　k-匿名性，分散匿名化，プライバシー保護データパブリッシング

1. まえがき

近年，複数の機関が保持するユーザのパーソナル情報を結合・分析し，新たな知見を得ることで，新たなサービスを創出することが期待されている．例えば，日本のセンチネル・プロジェクトに関する提言 [1] では，複数の医療機関が保持するレセプトデータ（診療報酬明細書）等の医療情報を結合・分析することで，「ある医薬品の使用者における特定の副作用（有害事象）の発生頻度を，当該医薬品を使用していない場合の有害事象の発生頻度と比較することが可能」になるといわれている．しかし，現状はプライバシー保護の観点で医療情報を結合・分析することは限定的となっている．そこで，今後はプライバシーを適切に保護し，医療情報を副作用分析等の医学研究に利活用することが求められている [1]．なお，本論文におけるパーソナル情報とは，個人を特定することができる個人情報にとどまらず，広く個人に関する属性情報の集合とする．

本論文では，機関 A と機関 B が医療機関であり，診療情報を保持しているとする．そして，医学研究のために双方の機関が連携し，機関 A と機関 B が保持する診療情報を結合して公開することを想定する．そして機関 A は一般的な内科診療を行っている病院，機関 B は性病（性行為感染症）を専門的に診療している病院とし，診療した患者の診療情報として，被保険者番号，診療日，疾病情報，薬品情報を保持しているとする．すると，機関 A と機関 B がもつ診療日と疾病情報と薬品情報を，被保険者番号を用いてひもづけて結合することで，ある患者の機関 B で処方した薬品情報と機関 A での疾病情報が結合されたデータを生成することができる．この結合されたデータが開示されることにより，ある民間の研究機関 C は，新薬を注射した患者に対する疾病の割合と，従来の薬品を注射した患者に対する疾病の割合を比較することで，副作用分析が可能になると考えられる．

† 日本電気株式会社情報・ナレッジ研究所，川崎市
Knowledge Discovery Research Laboratories, NEC Corporation, 1753 Shimonumabe, Nakahara-ku, Kawasaki-shi, 211–8666 Japan
†† 電気通信大学大学院情報システム学研究科，調布市
Graduate School of Information Systems, The University of Electro-Communications, 1–5–1 Chofugaoka, Chofu-shi, 182–8585 Japan

しかし, [1] で指摘されているように，医療情報は「直接個人を特定できる情報を除去しても，個人の特定につながる可能性のある情報」が含まれているとされており，複数の情報を組み合わせることで，個人が特定できてしまう可能性がある．例えば，先ほどの結合されたデータでは，たとえ被保険者番号のような個人を識別する情報を削除したとしても，研究機関 C にいる研究員が個人を特定できてしまう可能性がある．例えば，この研究員は，患者 X さんが 1 月 1 日に機関 A に診療し，2 月 2 日に機関 B に診療したことを知っていたとする．そして，このような患者が全患者の中で X さんの 1 名だけであったとする．すると，この研究員は，結合されたデータの X さんの診療情報を特定できてしまう．そして，X さんの疾病コードや投薬コードを知り，病名や病気がどのくらい進行しているか等の知られたくない情報を知ることができてしまう．このように，複数の情報の組合せから個人を特定されるおそれがあるため，機関 A，B は情報を開示する際の責務として，個人特定を防ぐための処理を行うべきであるといわれている [1], [2].

また，医療情報などのパーソナル情報は個人のプライバシーに関する情報であるため，他の機関へ全開示して結合することはできない．例えば，米国の HIPAA (Health Insurance Portability and Accountability Act) 法における必要最小限の情報開示の要件 (minimum necessary requirements) [3] のように，パーソナル情報を結合する際の情報開示は最小限にする必要がある．つまり,「(問題 1) 機関 C において個人が特定される問題」の解決が必要である．

そこで，各機関がもつ情報を必要最小限の開示にとどめながら結合し，ユーザが特定されない形式に加工した結合匿名テーブルを生成・提供する分散匿名化が注目されている [4]〜[7]．ここで，必要最小限の開示とは，ユーザが特定されない形式として開示された情報よりも詳しい情報が開示されていないということを意味する．例えば，結合匿名テーブルとして開示される診療日が年月レベルであった場合，機関 A，B の双方に開示される診療日は年月日レベルであってはならず，年月レベルにとどめなければならない．

しかし既存の分散匿名化の手法では，双方の機関のユーザ集合が一致しない場合に，結合匿名テーブルを参照することで，ユーザのパーソナル情報がその機関に保持されているか否かというユーザ存在が他方の機関に漏えいしてしまう問題があった．これは，機関 A の医師が，結合匿名テーブルを参照することで，風邪の診療の来た Alice さんが機関 B にも通院していることを知ることになる．性病の専門病院等への通院を他の一般の内科等の病院には知られたくないと考えられるため，ユーザ存在はユーザのプライバシに関わる情報といえる．つまり,「(問題 2) 機関 A，B の双方に対してユーザ存在が漏えいしてしまう問題」の解決が必要である．なお，ユーザ存在は患者のプライバシーに関わる情報であるので，機関 A，B にユーザ存在が漏えいすることを防ぐ必要がある．

そこで本論文では，従来の分散匿名化が対象としている問題 1 だけでなく，ユーザ存在が漏えいしてしまうという問題 2 の解決も目指す．この問題は，双方の機関が異なる属性のパーソナル情報を保持している際の分散匿名化において，双方の機関のユーザ集合が一致しない場合に発生する．本論文では，この問題を解決するために，分散匿名化におけるユーザ存在が知られる可能性を示した新たな指標として *δ-site-presence* を提案する．これは，既存の集中型の匿名化におけるユーザ存在が知られる可能性を示した *δ-presence* [8] という指標を，分散匿名化のために拡張した指標である．また本論文では，*δ-site-presence* を満たし，ユーザ存在を隠ぺいした新たな分散匿名化手法を提案する．そして，実際の患者のレセプトデータを用いて，提案手法の有用性を評価する．更に提案手法の計算量・通信量の評価を行う．

本論文は，以下のような構成になっている．まず，**2.** で関連研究を示す．次に，**3.** で分散匿名化におけるユーザ存在の隠ぺいの課題について説明する．続いて，**4.** でその課題を解決するための新たな指標を提案し，**5.** で分散匿名化手法を提案する．そして **6.** で，提案手法の有効性と計算量・通信量を評価する．最後に **7.** で本論文の内容をまとめる．

2. 関 連 研 究

本論文における「匿名化」とは，識別子を削除したとしても，いくつかの情報の組合せからユーザが特定されることを防ぐために，パーソナル情報を加工することである．ここで，パーソナル情報とは「属性」と「属性値」によって表現されるユーザに関する属性情報の集合である．本論文では，パーソナル情報をテーブル形式で表現し，レコードをユーザに，カラムを「属性」に，フィールドの値をユーザの属性の「属性値」にそれぞれ対応させる．また，単一の属性では

ユーザを特定できないが，複数組み合わせるとユーザを特定できる可能性のある属性の組合せを準識別子 (quasi-identifier) と呼び，ユーザが特定された状態で開示されることが望ましくない属性をセンシティブ属性 (sensitive attribute) と呼ぶ．また，匿名化に関する指標として，k-匿名性 (*k-anonymity*) [2], [9] がよく知られている．あるテーブルにおいて，準識別子の属性値によって識別されるレコードが少なくとも k 個以上ある場合に，そのテーブルは k-匿名性を満たすという．k-匿名性 (*k-anonymity*) を満たすために，属性値を曖昧な値に汎化 (generalize) する手法がよくとられる（例：23 歳 ⇒ 20〜25 歳）．

複数の機関が保持するテーブルを結合して匿名化する処理を分散匿名化と呼ぶ [4]〜[7]．分散匿名化は，パーソナル情報の分割形態の違いにより垂直分割と水平分割に分類される．垂直分割とは，本論文と同様に，ユーザのパーソナル情報が属性ごとに異なる機関に保持されている分割形態である．水平分割とは，ユーザのパーソナル情報がユーザごとに異なる機関に保存されている分割形態である．

垂直分割での分散匿名化としては [4]〜[6] などが存在する．[4], [5] では，本論文と同じ Top Down アプローチとセキュア計算 (secure computation) [10] を組み合わせた手法で，分散匿名化を実現している．Top Down アプローチとは，準識別子の属性値を最も汎化されている状態から徐々に詳細化 (specialize) する手法である．ここで詳細化とは，準識別子の属性値で識別されるユーザ集合を，ある境目で分割することである．この分割の境目となる属性値を分割点と呼ぶ．例えば，年齢を「30」という分割点で分割すると，「30 歳以上」と「30 歳未満」に分割することになる．分割後のユーザ集合のユーザ ID は，双方の機関で共有される．そして k-匿名性が満たされている間，分割を続ける．最後に，分割した双方のテーブル（内部匿名テーブル）を結合して最終的な結合匿名テーブルを生成する．Top Down アプローチで分割点を決定するために，分割点決定関数というヒューリスティック関数が用いられる．この関数の計算にはセキュア計算 [10] が用いられる．セキュア計算とは，自機関がもつ属性値を相手の機関に秘密にしながら，大小比較などが行える暗号プロトコルである．セキュア計算を用いる事で，属性値を相手機関に隠ぺいしながら分割点を決定することができる．

[6] では，Bottom Up アプローチを用いた垂直分割での分散匿名化を提案している．これは，それぞれの機関で個別に内部匿名テーブルを生成した後，結合匿名テーブルの匿名性が保たれることを確認しながら内部匿名テーブルを結合していく手法である．[7] では，水平分割での分散匿名化で発生するパーソナル情報の保存形式の違いから，情報の保存場所を知られてしまうという問題を，Top Down アプローチで解決している．また，この問題を解決するため *ℓ-site-diversity* という指標を提案している．

一方，分散匿名化ではないが公開テーブルと匿名テーブルにおいてユーザ存在の隠ぺいを目指した匿名化の研究が行われている．[8] では，*δ-presence* というユーザの存在の可能性を示す指標と，その指標を満たすための匿名化アルゴリズムを提案している．しかし，このアルゴリズムは分散匿名化ではないため，双方の機関でユーザが異なる場合におけるユーザ存在の隠ぺい課題には適用できない．一方，提案されている指標は分散匿名化にも適用可能である．そこで本論文では，*δ-presence* を分散匿名化に適用した指標を *δ-site-presence* として新たに定義している．

3. 分散匿名化における課題

3.1 分散匿名化

本節では，本論文の分散匿名化の前提を説明する．まず，機関 A，B が保持するパーソナル情報のテーブル形式を定義する．機関 A はテーブル形式 T_A を，機関 B は T_B を保持するとする．T_A はユーザ ID と QI_A（機関 A がもつ準識別子）を保持し，同様に T_B はユーザ ID と QI_B（機関 B がもつ準識別子）と SA（センシティブ属性）を保持するテーブル形式である．本論文では，以下のように表記する．

$$T_A(ID, QI_A), T_B(ID, QI_B, SA)$$

ここで，ID は共通のユーザ ID，QI_A, QI_B は機関 A，B がもつ準識別子，SA はセンシティブ属性である．また，分散匿名化によって生成される結合匿名テーブル T^* の形式は，

$$T^*(QI_A, QI_B, SA)$$

とする．T^* は，QI_A と QI_B の各属性値の組合せからの個人特定を防ぐために，T^* が k-匿名性を満たすように QI_A と QI_B の属性値が加工されている．これにより「(問題 1) 機関 C において個人が特定される

問題」を解決することができる．

更に分散匿名化では，必要最小限の開示にとどめながらテーブルを結合し匿名化を行う．ここで，必要最小限の開示にとどまる必要があるのは，異なる機関で完全な信頼関係を築くのは困難であり，テーブルを全て開示するのは危険であると考えられているためである．ただし，本論文では機関 A，B はある程度の信頼がおける機関であることを前提とし，各機関は semi-honest [11] で振舞うとする．semi-honest とは，各機関はプロトコルを介して得られた情報を解析して相手機関の情報を知ろうとするが，プロトコルを逸脱した攻撃は行わないという振舞いモデルのことである．つまり，例えば機関 A が，機関 B に保持されているパーソナル情報を得るために，機関 B に何度も分割を行わせるようなプロトコルを逸脱した攻撃は想定しない．

3.2 ユーザ存在の漏えいの課題

既存の垂直分割の分散匿名化では，双方の機関のユーザ集合が一致している前提があった [4]～[6]．しかし，今後は様々な機関同士でのパーソナル情報の結合が期待されるため，ユーザ集合が一致しない場合への対応が必要である．つまり，一部のユーザが片方の機関にだけ存在する場合にも対応する必要がある．

ユーザ集合が完全に一致せずに一部のユーザだけが一致する場合（一部のユーザだけが共通ユーザとなる場合），結合匿名テーブル T^* は，機関 A と機関 B の共通ユーザのレコードだけとなり，片方の機関にだけ存在するユーザのレコードは含まれない．この場合，「(問題 2) 機関 A，B の双方に対してユーザ存在が漏えいしてしまう問題」が発生する．この問題 2 は，更に「(問題 2-1) 結合匿名テーブルによるユーザ存在の漏えい問題」と「(問題 2-2) ユーザ ID 通知によるユーザ存在の漏えい問題」に分割できる．

まず,「(問題 2-1) 結合匿名テーブルによるユーザ存在の漏えい問題」について説明する．この問題は，自機関がもつテーブルと結合匿名テーブルの比較によってユーザ存在が漏えいしてしまう問題である．例えば機関 A がもつテーブル T_A が 表 1 (a)，機関 B がもつテーブル T_B が表 1 (b)，結合匿名テーブル T^* が表 1 (c) であったとする．表 1 では,「疾病 A」と「疾病 B」を，機関 A と機関 B での疾病を 3 けたの数字で表現した疾病コードとしている．また,「分類」を疾病 B の進行の区分（例：ガンの進行ステージ等）としている．

表 1　結合匿名テーブルによるユーザ存在の漏えい
Table 1　Example of the joined-anonymized table.

(a) 機関 A の T_A

ID	疾病A
User 1	110
User 2	140
User 3	155
User 6	165
User 7	171
User 8	190

(b) 機関 B の T_B

ID	疾病B	分類
User 1	550	I
User 2	540	II
User 4	580	I
User 5	560	II
User 6	500	I
User 7	521	II
User 9	520	I
User 10	510	II
User 11	525	I
User 12	500	II

(c) T^*(漏洩する場合)

疾病A	疾病B	分類
000-149	530-599	I
000-149	530-599	II
150-199	500-529	I
150-199	500-529	II

(d) T^*(漏洩しない場合)

疾病A	疾病B	分類
000-159	530-599	I
000-159	530-599	II
160-199	500-529	I
160-199	500-529	II

また,「疾病 A」と「疾病 B」は **3.1** における QI_A と QI_B，分類は SA のことである．つまり，T_A（表 1 (a)）の「疾病 A」と T_B（表 1 (b)）の「疾病 B」の値は，汎化されていない元の値である．また，T^*（表 1 (c)）の「疾病 A」と「疾病 B」の値は，汎化された値である．例えば，T^*（表 1 (c)）の疾病 A の「000-149」という汎化された値は，疾病 A の値が 000 以上かつ 149 以下の範囲であることを意味する．

このとき，T^*（表 1 (c)）では疾病 A が 000-149 である患者は 2 名，T_A（表 1 (a)）でも疾病 A が 000-149 に該当する患者は 2 名である．このことから機関 A は，User1，2 は確実に T^*（表 1 (c)）に含まれていると推測できる．更に，T^*（表 1 (c)）に含まれるユーザは機関 A と機関 B の双方に存在する共通ユーザであることから，機関 A は，User1，2 の 2 名が確実に機関 B にも存在すると推測できる．それに対し，T^* が表 1 (d) のように疾病コードが「160」で分割されていた場合，機関 A は，User1，2，3 の 3 名のうちいずれか 2 名が機関 B に存在することまでしか推測できない．

次に,「(問題 2-2)「ユーザ ID 通知によるユーザ存在の漏えい問題」について説明する．これは，プロトコル中のユーザ ID の通知によって，相手機関に自機関のユーザ存在が知られてしまう問題である．もし単純に既存の分散匿名化プロトコルを適用してしまうと，分割後のユーザを相手機関に通知する際に，自機関に存在するユーザ ID だけを通知することになる．すると，通知を受け取った機関は，通知されたユーザ ID のユーザは通知をしてきた機関に存在することを容易に推測できてしまう．

4. 提案指標：*δ-site-presence*

本章では「(問題 2-1) 結合匿名テーブルによるユーザ存在の漏えい問題」を解決するために，ユーザ存在

の隠ぺいの課題を解決するための新たな指標を提案する．集中型の匿名化におけるユーザ存在の推測の可能性を示す指標として，*δ-presence* [8] があるので，この指標を拡張し分散匿名化におけるユーザ存在の推測の可能性を示す指標として *δ-site-presence* を定義する．*δ-presence* とは，テーブル T_1 と匿名化されたテーブル T_2^* における，T_1 に存在するユーザのレコード内のデータが T_2^* にも存在する可能性を示した指標である．この T_2^* とは，T_1 の一部のレコードのデータから構成されたテーブル $T_2(T_2 \in T_1)$ を匿名化したテーブルである．例えば，T_1 がある会社の社員名簿のテーブルであり，T_2^* がその会社内のガン患者名簿を匿名化したテーブルであるとすると，*δ-presence* は社員がガン患者名簿に存在する可能性を示す指標といえる．

[8] では，ある属性の値 v で識別される T_1 のレコードのレコード数を $|T_1[v]|$，v で識別される T_2^* のレコードのレコード数を $|T_2^*[v]|$ としたとき，テーブル T_1 の属性 v で識別されるレコード $(T_1[v])$ が T_2 にも存在する可能性を $|T_2^*[v]|/|T_1[v]|$ と定義している．*δ-presence* によって，ユーザ存在の可能性が示されるため，意図しないユーザ存在の漏えいを防ぐことができる．

そして，このユーザ存在の可能性の定義を分散匿名化に適用し，ある機関に存在するユーザが相手機関にも存在するという意味のユーザ存在の可能性を示した指標として，*δ-site-presence* として定義する．これは，T_A のレコードが T^* にも存在する可能性と，T_B のレコードが T^* にも存在する可能性の両方を示す．

［定義 1］ T_A，T_B を機関 A，B がもつテーブル，T^* を結合匿名テーブルとする．そして，T^* のうち機関 $n \in \{A, B\}$ がもつ属性の属性値の組合せの集合を $\{v_{n,1}, \cdots, v_{n,m_n}\}$ とし，$v_{n,i} \in \{v_{n,1}, \cdots, v_{n,m_n}\}$ とおく．また，$v_{n,i}$ で識別されるテーブル T_n のレコード数を $|T_n[v_{n,i}]|$，$v_{n,i}$ で識別されるテーブル T^* のレコード数を $|T^*[v_{n,i}]|$ と表現する．このとき，以下の式で示されるように，機関 $n \in \{A, B\}$ の各 $v_{n,i}$ によるユーザ存在の推測の可能性が $\delta_{max,n}$ 以下かつ $\delta_{min,n}$ 以上であるとき，T^* は $\{\delta_{min,A}, \delta_{max,A}, \delta_{min,B}, \delta_{max,B}\}$-*site-presence* を満たすと定義する．

$$\delta_{min,n} \leq \frac{|T^*[v_{n,i}]|}{|T_n[v_{n,i}]|} \leq \delta_{max,n} \quad \forall n \in \{A, B\} \quad (1)$$

なお，このときの $\delta_{min,n}$ と $\delta_{max,n}$ は，機関 n のユーザが相手機関にも存在するというユーザ存在の可能性の最大値と最小値といえる．

例えば表 1 (d) のうち機関 A がもつ属性（疾病 A）の属性の属性値の組合せの集合 $\{v_{A,1}, v_{A,2}\}$ は {000-159, 160-199} である．まず，「000-159」について考える．T^*（表 1 (d)）のうち疾病 A が「000-159」であるレコードは二つであるので，$|T^*[v_{A,1}]|$=2 である．そして，T_A（表 1 (a)）のうち疾病 A が「000-159」を満たすレコードは三つであるので，$|T_A[v_{A,1}]|$=3 である．よって，疾病 A の「000-159」についてはユーザ存在の推測の可能性は 2/3 である．同様に「160-199」についても，ユーザ存在の推測の可能性は 2/3 である．続いて，機関 B がもつ属性（疾病 B，分類）の属性値の組合せの集合についても同様に計算すると，表 1 (d) は $\{2/3, 2/3, 1/3, 1/2\}$-*site-presence* を満たすテーブルであることが分かる．

このように *δ-site-presence* は，*δ-presence* のようにテーブルにおけるユーザ存在を示しているのではなく，機関におけるユーザ存在を示している．これは例えば，*δ-site-presence* を三つ以上の機関へ拡張した場合（**4. 2** で説明），単に結合匿名テーブルにユーザが存在するか否かを意味するのではなく，自機関に存在するユーザが他の二つの機関の両方にも存在するか否かを意味する．このように，*δ-site-presence* は，分散匿名化において重要な，機関におけるユーザ存在を表すことができる．

4. 1 *δ-site-presence* の設定の指針

本節では，*δ-site-presence* の $\delta_{min,n}, \delta_{max,n} (n \in \{A, B\})$ をどのような指針で設定するかについて説明する．これらに設定するべき値は，扱うパーソナル情報の種類に依存する．例えば，ユーザ存在が漏えいしてもプライバシーの侵害が小さいと考えられるような場合は $\delta_{max,n}$ の値は大きく設定し，ある程度のユーザ存在の漏えいを許容するようにしてもよい．逆に，例えば犯罪者データベースに存在するかどうかのように，ユーザ存在が漏えいした際のプライバシーの侵害が大きい場合は $\delta_{max,n}$ の値は小さく設定するべきである．

また，ユーザ存在が推測された際の被害額をもとに，これらの値を設定する方法もある．例えば，既存研究の [8] では，糖尿病患者であるかどうかを他人に知られた場合における被害額から，許容するユーザ存在が推測の確率を求める方法が提案されている．

また，[8] で示されているとおり，ユーザ存在の確率は T^* の全レコード数 $(|T^*|)$ と T_n の全レコード数 $(|T_n|)$

によって，ある程度決定される．例えば，表 1 の場合，属性値を無視してレコード数を数えると $|T_A| = 6$, $|T^*| = 4$ であるので，T_A に存在するレコードは少なくとも $4/6 = 2/3$ の可能性で T^* に存在すると推測される．これは，表 1 の場合，$\delta_{max,A}$ を $2/3$ よりも小さくすることはできないことを意味する．$\delta_{min,A}$ についても同様なことがいえ，$\delta_{min,A}$ を $2/3$ よりも大きくすることはできない．このように，$\delta_{max,n}$ は $|T^*|/|T_n|$ よりも小さくできず，$\delta_{min,n}$ は $|T^*|/|T_n|$ よりも大きくできない．つまり，$\delta_{min,n}$ と $\delta_{max,n}$ は以下の範囲で設定される必要がある．

$$0 \leq \delta_{min,n} \leq \frac{|T^*|}{|T_n|} \leq \delta_{max,n} \leq 1 \qquad (2)$$

4.2 三つ以上の機関への拡張の検討

本論文で提案する *δ-site-presence* は 2 機関に限定した指標となっているが，この指標を拡張し 3 機関以上でも適用可能であることを示す．まず，3 事業者の場合の例を示す．例えば機関 A，機関 B と機関 C が存在し，それぞれがもつ T_A，T_B，T_C を結合して匿名化した T^* を生成するとする．このとき，T^* には T_A，T_B，T_C に含まれる共通ユーザのレコードのみとなる．そして，例えば機関 A から見た場合，機関 A のユーザが機関 B,C の両方にも存在する可能性は，T_A で識別されるレコードのうち，どのくらいのレコードが T^* に存在するかという可能性になる．つまり，3 機関の場合の *δ-site-presence* は，ある機関のユーザが他の全機関にも存在する可能性を指定する指標となる．

3 機関の場合の *δ-site-presence* の拡張方法を踏まえ，複数機関の場合の *δ-site-presence* について定義する．

[定義 2] $\{T_1, \cdots, T_N\}$ を機関 $n \in \{1, \cdots, N\}$ がもつテーブル，T^* を $\{T_1, \cdots, T_N\}$ の結合匿名テーブルとする．そして，T^* のうち機関 n がもつ属性の属性値の組合せの集合を $\{v_{n,1}, \cdots, v_{n,m_n}\}$ とし，$v_{n,i} \in \{v_{n,1}, \cdots, v_{n,m_n}\}$ とおく．また，$v_{n,i}$ で識別されるテーブル T_n のレコード数を $|T_n[v_{n,i}]|$，$v_{n,i}$ で識別されるテーブル T^* のレコード数を $|T^*[v_{n,i}]|$ と表現する．このとき，以下の式で示されるように，機関 n の各 $v_{n,i}$ によるユーザ存在の推測の可能性が $\delta_{max,n}$ 以下かつ $\delta_{min,n}$ 以上であるとき，T^* は $\{\delta_{min,1}, \delta_{max,1}, \cdots, \delta_{min,N}, \delta_{max,N}\}$-*site-presence* を満たすと定義する．

$$\delta_{min,n} \leq \frac{|T^*[v_{n,i}]|}{|T_n[v_{n,i}]|} \leq \delta_{max,n} \ \forall n \in \{1, \cdots, N\} \qquad (3)$$

このように，*δ-site-presence* を三つ以上の機関に拡張することは可能であるが，本論文で提案している手法をそのまま三つ以上の事業者で用いることはできない．これは，提案手法で用いているセキュア計算のいつくかは 2 機関限定となっているためである．しかし，3 機関でも動作可能なセキュア計算の研究 [10] や，三つの機関以上の機関における分散匿名化手法 [4], [7] を参考にすることで，提案手法を三つ以上の機関に対応するように拡張可能であると考える．

5. 提案手法：ダミーユーザ手法

本章では，*δ-site-presence* を満たしつつ，「(問題 2-2) ユーザ ID 通知によるユーザ存在の漏えい問題」を解決するための分散匿名化の手法を提案する．問題 2-2 は，ユーザ ID を通知する際に，通知をする機関に存在するユーザ ID だけを通知するから発生してしまう．そこで，存在しないユーザのユーザ ID も通知するために，ダミーユーザを導入する．ダミーユーザは，自機関に存在しないユーザを，あたかも存在するかのように扱うユーザのことである．なお，ダミーユーザに対して，存在するユーザを存在ユーザと呼ぶ．ダミーユーザを導入することにより，通知されるユーザ ID がダミーユーザなのか存在ユーザなのかの区別を困難にでき，問題 2-2 を解決することができる．

このようなダミーユーザを用いた提案手法は，問題 1，2 を満たすために以下の要件を満たしつつ，できるだけ詳細な結合匿名テーブル T^* を出力する必要がある．

(要件 1)　T^* は *k*-匿名性を満たすこと
(要件 2)　プロトコルの通信内容から，T^* から推測される以上の詳しい情報が極力漏れないこと
(要件 3)　T^* は *δ-site-presence* を満たすこと
(要件 4)　プロトコルの通信内容から，T^* から推測される以上の詳しいユーザ存在が極力漏れないこと

ここで，要件 1 と要件 2 は既存の分散匿名化の要件と同じであり，問題 1 の解決のための要件であたる．そして要件 3 と要件 4 は，問題 2 の解決のために追加された要件であり，それぞれ問題 2-1 と問題 2-2 の解決のための要件にあたる．

そこで，要件 1 と要件 2 だけでなく要件 3 と要件 4 も満たすために，既存の Mondrian [12] を拡張し，ダミーユーザを導入したダミーユーザ手法を提案する．なお，Mondrian とは，*k*-匿名化を行うための Top Down アプローチの匿名化アルゴリズムとして広く利

表 2　内部匿名テーブル T_A^*, T_B^* と結合匿名テーブル T^*
Table 2　Internal anonymized table T_A^*, T_B^* and joined-anonymized table T^*.

(a) 機関Aの内部匿名テーブル T_A^*

	GID	IDs	疾病A
初期	1	User 1-15	000-199
1回目の分割	2	User 1-10	000-199
	3	User 11-15	000-199
2回目の分割	4	User 1-5	000-149
	5	User 6-10	150-199
	3	User 11-15	000-199

(b) 機関Bの内部匿名テーブル T_B^*

GID	IDs	疾病B	userCounts
1	User 1-15	500-599	-
2	User 1-10	500-529	-
3	User 11-15	530-599	-
4	User 1-5	500-529	I:1, II:1
5	User 6-10	500-529	I:1, II:1
3	User 11-15	530-599	I:1, II:2

(c) 結合匿名テーブル T^*

疾病A	疾病B	分類
000-149	500-529	I
000-149	500-529	II
150-199	500-529	I
150-199	500-529	II
000-199	530-599	I
000-199	530-599	II
000-199	530-599	II

用されているアルゴリズムであり，既存の [7] の分散匿名化手法でも採用されている．そして，提案するダミーユーザ手法では，k-匿名化だけでなく δ-*site-presence* も満たす必要があるため，既存の Mondrian の分割点決定関数を拡張する．更に，分散匿名化では各機関がもつ属性値などの情報が相手機関に知られないようにするため，セキュア計算を用いる．

ダミーユーザ手法では，まず機関 A，B が内部匿名テーブル $T_n^*(n \in \{A, B\})$ を分割するためのプロトコル（分割プロトコル）を実行し，各機関内で T_n^* を生成する．その後，機関 C が機関 A，B から T_n^* を取得し，単純に結合することで T^* を得る．以降では，**5.1** で，ダミーユーザ手法の分割プロトコルの処理の詳細を説明する．なお，T_n^* の分割と T^* の例を表 2 に示す．この例では，表 1 と同様に疾病 A，疾病 B が QI_A，QI_B で，分類が SA である．続いて，**5.2** で，k-匿名性だけでなく δ-*site-presence* も満たすように拡張したダミーユーザ手法のための分割点決定関数について説明する．更に，**5.3** では，ダミーユーザ手法においてセキュア計算をどのように用いているかについて説明する．

5.1　ダミーユーザ手法の処理

本節では，ダミーユーザ手法における機関 A，B の T_n^* を分割するための分割プロトコルの処理の詳細を説明する．このプロトコルは，三つの Step で構成される．以降の節で，各 Step の動作の詳細を説明する．

5.1.1　Step1：ダミーユーザの割当と T_n^* の初期化

最初に，機関 A と機関 B は自機関のダミーユーザを割り当てる．本手法では，双方の機関のユーザを包含する母集団ユーザ集合 U を事前に知っているという前提を置く．ここで U は，機関 A に存在するユーザ集合を U_A，機関 B に存在するユーザ集合を U_B，機関 A，B のどちらにも存在しないユーザ集合を U_O としたとき $U = U_A \cup U_B \cup U_O$ $(U_O \neq \phi, U_A \cap U_B \neq \phi)$ となる．このような前提は，例えば機関 A，B が Open ID [13] のような中央集中型の認証サーバを利用している場合に成立し，認証サーバに存在する全ユーザが U となる．そして機関 A と機関 B は，それぞれのダミーユーザを $U - U_A$，$U - U_B$ と割り当てる．

function split(U_p:分割対象となるユーザ集合の IDs)
1: U_p のダミーユーザのダミー値を更新
2: $d \Leftarrow$ 分割点決定関数を用いて分割点を決定
3: **if** k-匿名性と δ-*site-presence* を満たせない **then**
4: 　U_p についての split 処理終了
5: **endif**
6: **if** d は自機関の T_n^* の分割点 **then**
7: 　T_n^* を d で分割し，分割後の IDs を相手の機関へ送信
8: **else**
9: 　相手から分割後の IDs を受信し，T_n^* を分割
10: **endif**
11: $U_{hi}, U_{low} \Leftarrow$ 分割後の IDs. split(U_{hi}),split(U_{low}) を実行

図 1　ダミーユーザ手法の Step2 のアルゴリズム
Fig. 1　Algorithm of step 2 (split function).

次に内部匿名テーブル T_n^* を初期化し，最も一般化された状態にする（表 2 (a)，(b) 上）．各機関の内部匿名テーブルは $T_A^*(GID, IDs, QI_A)$，T_B^* $(GID, IDs, QI_B, userCounts)$ である．ここで GID とは，シーケンシャルに割り当てられる T_n^* の各レコードの識別子である．IDs とは，T_n^* のレコードに該当するユーザ ID の集合である．$userCounts$ とは，IDs で示されたユーザ集合における SA の各属性値の共通ユーザ数であり，Step3 で計算される．

5.1.2　Step2：分割点の決定と分割処理

続いて，T_n^* を分割していく分割処理を行う（図 1）．まず，機関 A，B は自機関のダミーユーザの準識別子 (QI_A, QI_B) の属性値に適切な値を割り当てる．この値をダミー値と呼ぶ．ダミーユーザは，相手機関からみて存在ユーザなのかダミーユーザなのか区別がつかないようにする必要がある．また，分割を多くできるように存在ユーザとダミーユーザの割合を全体で均一にすると良い．そのため，分割を行う度に分割対象のユーザ集合における存在ユーザの準識別子の属性値の

分布に沿ってダミー値を割り当てる．

このようにダミーユーザ手法では，単にランダムにダミー値を決定するのではなく，分布に沿ってダミー値を決定している．特に，分割を行うたびにダミー値を修正している点が特徴である．これは，機関 A，B の属性間に相関がある場合にダミーユーザが偏ってしまうことを防ぐためである．機関 A，B は相手機関の属性値を知ることができないため，機関 A，B の属性間に相関があったとしても，相関に沿ってダミー値を決定することができず，ダミーユーザが偏ってしまう．そこで，分割によって判明した相手機関の属性値の分布に沿ってダミー値を修正することにより，ダミーユーザの偏りを少なくしている．その結果，ダミーユーザを相関に沿って均一に配置することができ，多くの分割ができるようになる．このように，本手法のダミー値の決定方法は，ランダムノイズを加えるような摂動法 [14] とは異なり，分散匿名化におけるユーザ存在の隠ぺいに特化した手法となっている．

次に，分割点決定関数を用いて分割点を決定する．この処理の詳細は **5. 2** で説明する．そして，決定した分割点で分割しても k-匿名性と δ-*site-presence* を満たせるかを確認し，指標を満たしている場合のみ T_A^*，T_B^* を分割する．なお，どのように指標を満たしているか計算するかについては，**5. 3. 2** で説明する．そして，分割後のユーザ ID に対して再帰的に上記の分割処理を繰り返していく．表 2 (a)，(b) 中下に機関 A と機関 B の内部匿名テーブル (T_A^*, T_B^*) の 1 回目と 2 回目の分割の例を示す．この例の 1 回目の分割（表 2 (a)，(b) 中）では，機関 B の「疾病 B」の「530」で分割され，疾病 B が「500-529」のレコード（GID が 2 のレコード）と，疾病 B が「530-599」のレコード（GID が 3 のレコード）が生成されている．そして，2 回目の分割（表 2 (a)，(b) 下）では，GID が 2 のレコードが機関 A の「疾病 A」の「150」で分割され，疾病 A が「000-149」のレコード（GID が 4 のレコード）と，疾病 A が「150-199」のレコード（GID が 5 のレコード）が生成されている．なお GID が 3 のレコードは，2 回目の分割において分割の対象とならなかったので，1 回目の分割からの変更は特になく，テーブル内に残ることになる．

5. 1. 3　Step3：ダミーユーザの削除

全ての分割処理が完了したらダミーユーザを削除し，共通ユーザ数を求める．なお，ダミーユーザの削除ではセキュア計算を用いて計算する必要がある．詳細は，**5. 3. 3** で説明する．

以上のような Step1〜3 までの処理によって，機関 A，B はお互いにユーザ存在を隠ぺいしながら機関 A，B は内部匿名テーブル T_A^*，T_B^* を分割していく．そして，機関 C が T_A^*，T_B^* を取得して GID をキーに結合することで T^* を得る（表 2 (c)）．なお，機関 C には IDs は不要なので削除する．

5. 2　ダミーユーザ手法の分割点決定関数

本節では，ダミーユーザ手法のための分割点決定関数を説明する．従来の Mondrian の分割点決定関数は，各属性の正規化済みの値域 (normalized range) が最大となる属性を選択し，その属性の中央値 (median) を分割点にしている．この従来の分割点決定関数を拡張し，新たに δ-*site-presence* も満たしやすい分割点が選ばれるようにする．そのためには，分割後のユーザ集合にダミーユーザが偏りなく入る分割点が選ばれると良いと考えられる．例えば表 1 (c) の T_A では，「000-149」は user1,2,「150-199」は user3，6，7，8 である．このうち機関 B のダミーユーザは user3，8 であるため，ダミーユーザが偏っている．それに対し表 1 (d) はダミーユーザが偏っていない．

そこで，ダミーユーザのエントロピー（シャノンの平均情報量）を導入する．エントロピーは，事象全体における各事象の発生確率の偏りが小さいほど大きな値になる．ダミーユーザのエントロピー (Dummy Entropy，DE) を，以下のように定義する．

$$DE(c,n) = -\sum_{U_i \in \{U_{hi}, U_{low}\}} \frac{|d(n,U_i)|}{|U_i|} \cdot log\left(\frac{|d(n,U_i)|}{|U_i|}\right) \tag{4}$$

ここで c は分割点候補であり，分割前のユーザ集合 U_p を上位 U_{hi} と下位 U_{low} へ分割する属性値を意味する．また，$d(n,U_i)$ はユーザ集合 $U_i \in \{U_{hi}, U_{low}\}$ から機関 n のダミーユーザを抜き出したユーザ集合である．

この DE を利用して，ダミーユーザ手法の分割点決定関数を定義する．まず，従来の Mondrian と同様に normalized range が最大となる属性を選ぶ．そして，その属性における分割点の候補となる属性値 $(x_i \in X)$ を分割点候補 c_i として，以下のように定義したスコア値 S を計算する．

$$S(c_i) = (1-\alpha)\left(\frac{-L(c_i)}{\max_{x_j \in X}(L(x_j))}\right)$$

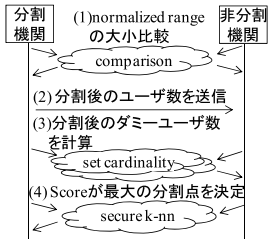

図 2 Step 2 の分割点決定関数
Fig. 2 Heuristic function.

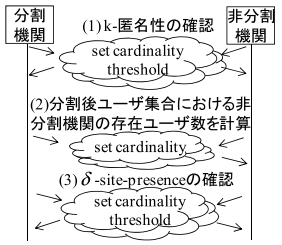

図 3 Step 2 の各指標確認
Fig. 3 Verifying indicators.

$$+\alpha\frac{1}{2}\sum_{n\in A,B}\left(\frac{DE(c_i,n)}{\max_{x_j\in X}(DE(x_j,n))}\right) \quad (5)$$

$$L(c_i)=\sum_{x_j\in X}|x_j-c_i| \quad (6)$$

ここで $\alpha(0\le\alpha\le 1)$ は，DE の影響を調整するための重みである．また，L は c_i の属性の各属性値 x_i と c_i の距離の和を意味する．median とは L が最小となる点と言い換えることができるため，α=0 としたときは c_i が median のときに S が最大となり，従来の Mondrian と同様に median が分割点に決定される．スコア値 S は，L と機関 A，B についての DE を正規化して，重み付で足した値となる．そして S を最大化させる分割点で分割を行うことで，分割後のユーザ集合における，ユーザ数に対する機関 A，B のダミーユーザ数の割合の偏りがほぼなくなるように分割が行われ，結果的に *δ-site-presence* を満たしつつ多くの分割が可能になることが期待される．なお，提案した分割点決定関数をセキュア計算を用いて計算する方法については，**5. 3. 1** で説明する．

5. 3 ダミーユーザ手法におけるセキュア計算

5. 3. 1 セキュア計算を用いた分割点の決定

本項では，ダミーユーザ手法の分割プロトコルの Step2 における分割点決定関数の計算を，セキュア計算を用いてどのように行っているかについて説明する．提案する分割点決定関数は，属性値やユーザ存在を隠ぺいしたまま計算する必要があるため，3 種類のセキュア計算を用いる（図 2）．まず，分割点の属性を選ぶ処理で *secure comparison* [15] を用いる（図 2 (1)）．これは，機関 A，B がもつ値を秘密にしながら大小関係を求めるプロトコルである．*secure comparison* を用いて，機関 A，B がローカルで計算した最大の normalized range を比較し，どちらが大きいかを求め，分割を行う機関（分割機関）と行わない機関（非分割機関）を決定する．

次に，機関 A，B で分割点候補 c_i の DE を計算する．ここで，非分割機関は分割後のユーザ集合を知らないので，分割後のユーザ数 ($|U_i|$) と非分割機関 n のダミーユーザ数 ($|d(n,U_i)|$) をローカルで計算できない．そこで $|U_i|$ は分割機関から取得する（図 2 (2)）．$|d(n,U_i)|$ については，*secure set intersection* の *cardinality* を用いて，分割後のユーザ集合 U_i と非分割機関 n のダミーユーザの積集合の要素数を得ることで計算する（図 2 (3)）．以上により，分割機関の c_i の分割点の属性値や分割後のユーザ集合を知ることなく，DE の計算に必要な情報を得ることができたため，DE を機関内でローカルに計算できる．

最後に，機関 A，B は *secure k-nearest neighbor* [16] というセキュア計算のプロトコルを用いて，分割点を決定する（図 2 (4)）．これは，機関 A，B がローカルで計算した正規化した DE と L について，それらを足した $S(c_i)$ が最大となる分割点候補を得る処理になる．以上のように，属性値やユーザ存在を相手機関に秘密にしながら分割点を決定することができる．

5. 3. 2 セキュア計算を用いた指標の確認

本項では，ダミーユーザ手法の分割プロトコルの Step2 における，*k*-匿名性と *δ-site-presence* を満たしているかの確認処理を，セキュア計算を用いてどのように行っているかについて説明する．これらの指標を満たしているかの確認には相手機関にユーザ存在を知られてはいけないので，*secure set intersection* [17] というセキュア計算 [10] のプロトコルを用いる（図 3）．このプロトコルは，機関 A，B がもつ集合をお互いに隠ぺいしながら，それらの集合の積集合や，積集合の要素数 (*cardinality*) や，積集合の要素数と指定した値との大小関係 (*cardinality threshold*) を求めることができる．分割後のグループで *k*-匿名性を満たしているかを確認するためには，機関 A，B は存在ユーザの

ユーザ ID の集合を入力として *cardinality threshold* を実行し，積集合の人数 ($|T^*[v_{n,i}]|$) が k 以上であるかを求めればよい（図 3 (1)）．*δ-site-presence* を満たしているかを確認にするには，例えば $\delta_{max,A}$ の確認の場合は，先ほどと同様に *cardinality threshold* を用いて「$|T^*[v_{A,i}]| \leq \delta_{max,A}|T_A[v_{A,i}]|$」を確認すればよい．ただし，機関 A が非分割機関であった場合は，機関 A は分割点候補の分割後のユーザ集合を知らないので $|T_A[v_{A,i}]|$ をローカルで計算できない．そこで機関 A は，機関 B の分割後グループの *ID*s と機関 A の存在ユーザの *ID*s を入力として *cardinality* を実行し，$|T_A[v_{A,i}]|$ を得る（図 3 (2)）．そして，$\delta_{max,B}$，$\delta_{min,A}$，$\delta_{min,B}$ についても同様に計算し，*cardinality threshold* を用いて，*δ-site-presence* を満たしているかを確認する（図 3 (3)）．

5.3.3 セキュア計算を用いたダミーユーザの削除

本節では，ダミーユーザ手法の分割プロトコルの Step3 におけるダミーの削除の処理を，セキュア計算を用いてどのように行っているかについて説明する．

これは，*secure set intersection* の *cardinality* を用いて，T_n^* の各レコードの SA の各属性値 s について，機関 A の存在ユーザのユーザ ID の集合，機関 B で s をもつ存在ユーザのユーザ ID の集合との積集合の個数を求めればよい．例えば表 2 (b) 下の user1-5 のレコードでは，機関 A の存在ユーザのユーザ ID の集合と，「分類」が「I」の機関 B の存在ユーザのユーザ ID の集合を入力として与えた結果，積集合の個数が 1 として出力された例である．

6. 評　　価

6.1 有効性評価

提案手法をプロトタイプ実装し，有効性を評価した．実装は Java 1.6 で行い，仮想的に双方の機関で通信を行う構成で動作させた．

評価データには，JMDC（株式会社日本医療データセンター）が提供している，いくつかの特定の健康保険組合に加入している約 10 万人の糖尿病患者の糖尿病以外の過去の疾病を含む実際のレセプトデータ（診療報酬明細書）の一部を用いた．このデータは，個人の特定はできないが複数の医療機関での個人データの結合はできるように，氏名や地域に関する情報は別コードに置き換えられている．このレセプトデータから，異なる診療科の医療機関のうち，共通の患者数が一番多い医療機関を機関 A，機関 B として抽出した．機関 A は約 3500 人の患者 (U_A) のデータをもつ内科の病院であり，機関 B は約 300 人患者 (U_B) のデータをもつ耳鼻科の病院である．そして，これら機関の共通の患者 ($U_A \cap U_B$) は約 230 人である．つまり，評価データは内科と耳鼻科のレセプトデータである．評価では，これらの患者とは別に，機関 A，B に通院していない患者 (U_O) を約 1430 人抜き出し，母集団の患者 (U) を約 5000 人とした．そして，二つの機関間で患者の疾病履歴を結合して病気の相関を調べるというユースケースを想定し，T_A(ID, 病名 A1, 病名 A2)，T_B(ID, 病名 B1, 病名 B2, 分類) というデータ形式のテーブルを生成した．ここで ID は機関 A と機関 B で共通な患者の識別子，「病名 A1」と「病名 A2」は機関 A における直近に診療した 2 件の疾病の疾病コードである．機関 B の「病名 B1」と「病名 B2」も同様である．また，「分類」は疾病の進行を想定しており，今回の評価結果に影響がないため疑似的に「I」と「II」をランダムに生成した．なお，結合テーブルの形式は T^*(病名 A1, 病名 A2, 病名 B1, 病名 B2, 分類) であり，{病名 A1, 病名 A2, 病名 B1, 病名 B2} が QI，分類が SA である．

評価は，結合匿名テーブルに対してデータマイニングを行った場合に，マイニング結果にどの程度の誤差が発生するのかという観点で行った．評価手法は，既存の匿名化の研究 [18] と同様に，ある条件に合致するユーザ数をカウントするクエリ (“select count(*) from T^* where 条件部”) の結果の相対誤差 (*relative error*) を計測するという手法である．なお，このクエリはデータマイニングにおける基本的な集約クエリ (*aggregate query*) とされている．この評価手法では，まずカウントされるユーザ数の割合の期待値 (*expected selectivity*) を $\theta(0\% < \theta < 100\%)$ とおいて，条件部に指定する検索範囲が全体の θ 倍になるようなクエリをランダムに生成する．つまり，T^* に含まれるユーザ数が 1200 であった場合，θ=10%としたクエリで検索されるユーザ数（レコード数）は約 120 となる．そして，生成したクエリを用いて，匿名化前の結合テーブル T_{AB}（T_A と T_B を単純に内部結合したテーブル）に対して得られたユーザ数を *act*，結合匿名テーブル T^* に対して得られたユーザ数を *est* とし，その相対誤差を $|act - est|/act$ で計算する．なお，*est* はクエリの条件部に記載された範囲と，汎化された値の重なり度合に応じて算出する．例えば T^* に「20〜29 歳」というレコードが 5 個であり，クエリが「20〜21 歳」

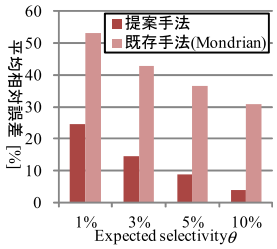

図 4　既存分散匿名化との比較
Fig. 4　v.s. Mondrian.

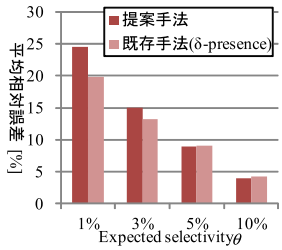

図 5　既存集中型との比較
Fig. 5　v.s. MPALM.

であった場合は，このクエリは「20〜29 歳」の 20%が重なっているので，$est = 5 \times 20\% = 1$ となる．なお，条件部に利用する属性は二つとし，ランダムに選択した．また各評価値は，ランダムなクエリを 10,000 回生成し相対誤差を計測した値の平均である．

評価は大きく三つの観点で行った．まず，既存の分散匿名化手法と提案手法を比較し，提案手法の有用性の評価を行った．続いて，既存の集中型のユーザ存在隠ぺいの匿名化手法と提案手法を比較評価した．最後に，本手法のパラメータである重み α が与える影響について評価した．

6.1.1　既存の分散匿名化との比較

最初に，提案手法となるダミーユーザ手法の有効性を評価するために，既存手法となる Mondrian を単純に分散環境に対応させた分散対応 Mondrian との比較を行う．この分散対応 Mondrian は，提案手法と比較するために k-匿名性だけでなく δ-*site-presence* も満たしている際に分割を行い，最終結果では共通ユーザだけを出力する分散匿名化手法である．

図 4 に，提案手法と既存手法のそれぞれについて，k=2，$\delta_{max,A}$=$\delta_{max,B}$=0.99，$\delta_{min,A}$=$\delta_{min,B}$=0.01，θ=$\{1\%, 3\%, 5\%, 10\%\}$ として平均相対誤差を計測した結果を示す．なお，重み α は 0.5 として DE の影響を半分にしている．

この結果が示すように，θ=3%のときの既存手法の相対誤差は約 40%と大きいが，提案手法の相対誤差は約 15%程度と小さい．これは，ダミーユーザのエントロピー (DE) の追加や分割後のダミー値の更新により，ユーザ存在が隠ぺいできるような分割点が選ばれるようになったためである．

また，相対誤差が約 15%というのは，例えば相関ルールマイニングを行った際に得られる相関ルールの支持度 (support) や確信度 (confidence) の相対誤差が約 15%程度であることを意味している．図 6 は，匿名化前の結合テーブル T_{AB} に対して相関ルールマイニングを行い，支持度が 2%以上，確信度が 50%以上となる疾病についての相関ルールを，支持度が高い順に出力した結果である(注1)．この結果に示したように，支持度 (support) が 3.1%と 2.2%の相関ルールが得られている．もし，匿名結合テーブル (T^*) に対して相関ルールマイニングを行った場合は，これらの相関ルールの支持度に 15%の誤差が入るので 3.1%と 2.2%の相関ルールの支持度は約 2.6〜3.6%と約 1.9〜2.5%になる．よって，この程度の誤差であれば，得られた相関ルールの支持度の大小関係が逆転するようなことは少なく，得られた相関ルールに大きな差はないと考える．このように提案手法がマイニング結果に与える影響は小さいと考えられるため，提案手法は十分有用であると考える．

A:急性気管支炎 ⇒B:急性副鼻腔炎 [sup=3.1%,conf=71.4%]
A:急性気管支炎 ⇒B:アレルギー性鼻炎 [sup=3.1%,conf=57.1%]
A:急性上気道炎 ⇒B:急性咽頭喉頭炎 [sup=2.2%,conf=60.0%]

図 6　機関 A（内科）と機関 B（耳鼻科）の疾病の相関ルール
Fig. 6　Assosication rules of diseases.

続いて，機関 A，B の δ_{min} か δ_{max} を設定を変化させて相対誤差を計測した．図 7 に θ=3%，$\delta_{max,A}$=$\{0.9, \cdots, 0.7\}$，　$\delta_{min,A}$=$\{0.5, \cdots, 0.8\}$，$\delta_{max,B}$=$\{0.10, \cdots, 0.04\}$，$\delta_{min,B}$=$\{0.02, \cdots, 0.08\}$ とした際の提案手法と既存手法の平均相対誤差を示す．なお，例えば $\delta_{max,A}$ を設定している際は他の δ の設定せずに $\delta_{max,A}$ の影響のみを評価している．

この結果が示すように，提案手法も既存手法も $\delta_{max,A}$ か $\delta_{min,A}$ が 0.75 に近づくと急激に誤差が大きくなるのに対し，$\delta_{max,B}$ か $\delta_{min,B}$ が 0.06 に近づくと急激に誤差が大きくなる．これは，機関 A から

(注1)：図 6 に示した疾病は，鼻や咽喉頭の炎症が気道や気管支に到達した際に起こる合併症としてよく知られている．

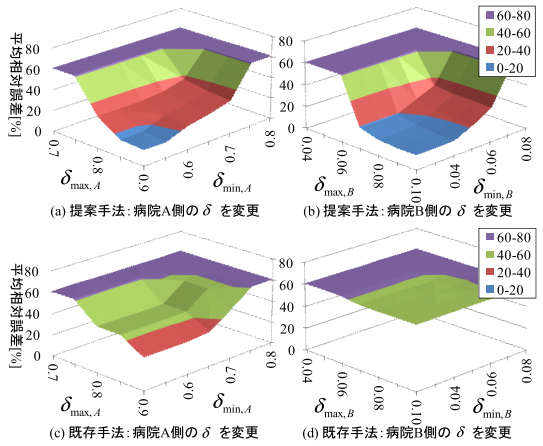

図 7　δ を変化させた際の相対誤差
Fig. 7　Relative errors of several δ.

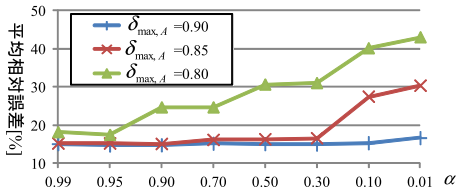

図 8　重み α の影響
Fig. 8　Effective of weight α.

見たユーザ存在の隠ぺいの限界値（$\delta_{max,A}$ として設定できる値の最小値，$\delta_{min,A}$ として設定できる値の最大値）が 0.75 ($\simeq$ 230/300) であり，機関 B から見た限界値が 0.06 ($\simeq$ 230/3500) であるからである．

既存手法（図 7 (b)，(d)）は δ_{max} や δ_{min} を限界値近くに設定していなくとも，誤差が 30〜50%もあり，マイニング結果に与える影響が大きくなってしまう．それに対し，提案手法（図 7 (a)，(c)）は限界値付近でなければ相対誤差が小さいことが分かる．つまり，δ_{max} や δ_{min} を限界値近くに設定しなければ，相対誤差が小さくなるような有効な匿名化が行えることが分かった．

6.1.2　既存の集中型のユーザ存在隠ぺいとの比較

次に，集中型（非分散環境の匿名化）でのユーザ存在の隠ぺい手法である *δ-presence* を満たすための MPALM アルゴリズム [8] と比較し，分散型（分散環境の分散匿名化）に対応した提案手法の有用性がほぼ同等であることを示す．集中型での既存手法は，あるテーブルと匿名テーブルにおけるユーザ存在を隠ぺいする手法であり，提案手法のように機関 A と機関 B の双方からみた，ユーザ存在の推測を防ぐというものではない．そこで，公平な評価を行うために機関 B 側から見た $\delta_{min,B}$ と $\delta_{max,B}$ を設定せずに評価を行った．図 5 に θ={1%, 3%, 5%, 10%} として提案手法と既存手法の平均相対誤差の値を計測した結果を示す．なお，その他のパラメータは **6.1.1** と同じにした．

この結果が示すように，θ が 1〜3%のときは提案手法は既存手法よりも数%ほど誤差が大きい．これは，集中型の既存手法のほうがより多くの分割を行えるため，より小さい範囲のユーザ数のカウントであっても相対誤差を小さくできるからである．これは，分散型の提案手法は分割点決定関数を用いて分割点を探索するアルゴリズムであるのに対し，集中型の既存手法は分割点候補に対して実際に分割を行った際にユーザ存在を隠ぺい可能であるかを何度も確認し，分割点を探索するアルゴリズムのためである．もし分散型でこのような探索を行うと，分割可能かどうかの情報からユーザ存在が知られてしまう．しかし，θ が 5〜10%のときは提案手法と既存手法の差はほぼない．これは，比較的広い範囲の条件でのユーザ数のカウントであれば，集中型によって得られるユーザ数とほぼ同じであるということである．この結果から，提案手法は集中型の既存手法と大きな差がなく，有効な匿名化が行えることが分かった．

6.1.3　適切な重み α の設定

更に分割点決定関数の重み α の最適値を調べるために，α を変化させて評価を行った．図 8 に $\delta_{max,A}$ を {0.90, 0.85, 0.80} として評価を行った結果を示す．なお，その他のパラメータは **6.1.1** と同じにした．

この結果が示すように，δ が限界値付近に設定された場合 ($\delta_{max,A}$=0.80) は α の重みが重要になり，α が小さいほど相対誤差が小さくなる傾向がある．これは，α が限界値に近い場合は，ダミーユーザのわずかな偏りで *δ-site-presence* を満たさなくなるので，*DE* の影響が大きくなるように設定したほうがよくなる．評価の結果，α は 0.90〜0.95 に設定するとよい．

6.2　計算量と通信量の評価

本節では，提案手法の計算量と通信量を評価する．我々は以前，[19] において，本論文の提案手法に似た手法を提案した．この手法は，ダミーユーザを用いることでユーザ存在の隠ぺいした分散匿名化を実現しているが，Multi Party Computation (MPC) [15], [20] という暗号プロトコルを用いている手法であった．し

かし，MPC は任意の複雑な関数を計算できるが，計算量や通信量が多くなってしまう問題があり，以前提案した手法は実際に動かすことが困難な手法となっていた．それに対し本論文の提案手法は，計算量や通信量を少なくするために，MPC の代わりに積集合や比較演算などの関数に限定されるが比較的計算量が少ない暗号プロトコルであるセキュア計算 [10] を用いている．そこで，提案したダミーユーザ手法の平均的な計算量と通信量のオーダーを算出する．そして，計算量と通信量が，既存のセキュア計算よりも大幅に増加していないことを確認する．

ダミーユーザ手法では，Step2 の処理が分割を繰り返す処理のため最も計算量と通信量が大きくなる．特に Step2 の分割点決定関数を算出する処理において，分割点候補ごとにセキュア計算を実行するため，計算量と通信量が大きくなる（図 1 の 2 の処理）．分割点決定関数では，図 2 に示したように，まず *secure comparison* を用いて分割する属性を決定し（図 2 (1)），次に *secure set intersection* を用いて分割点候補ごとにスコア値 S を計算し（図 2 (2)，(3)），最後に *secure k-nearest neighbor* を用いてスコア値が最大となる分割点候補を選ぶという処理をしている（図 2 (4)）．この処理のうち「(i) 分割点候補ごとにスコア値を計算する処理」と「(ii) スコア値が最大の分割点候補を選ぶ処理」のセキュア計算は計算量と通信量が大きくなる．これらの処理の計算量と通信量を算出し，それをもとにダミーユーザ手法の計算量と通信量を算出する．

6. 2. 1 平均計算量の算出

まず「(i) 分割点候補ごとにスコア値を計算する処理」の平均計算量を算出する．分割の 1 回目は，分割対象のグループのサイズは $|U|$ となる．なお，簡略化のため $|U|$=N とおく．本手法の分割では多少の偏りはあるが平均的に中央で分割されるので 2 回目以降のグループサイズは $\frac{N}{2}, \frac{N}{4}, \cdots$ となる．また，これらのグループの個数は $2, 4, \cdots$ となる．更に，各グループにおける分割点候補は，グループのサイズとほぼ同じなので $N, \frac{N}{2}, \frac{N}{4}, \cdots$ となる．ここで，*secure set intersection* の計算量は，二つの集合の要素数を両方とも M とおいたとき $O(MloglogM)$ となる [17]．よって，1 回目の分割での計算量は，サイズ N の一つのグループに対する計算量を N 個の分割点候補分行うので $O(NloglogN) \times 1 \times N$，2 回目の分割ではサイズ $\frac{N}{2}$ の二つのグループに対する計算量を $\frac{N}{2}$ 個の分割点候補分行うので $O(\frac{N}{2}loglog\frac{N}{2}) \times 2 \times \frac{N}{2}$，3 回目は $O(\frac{N}{4}loglog\frac{N}{4}) \times 4 \times \frac{N}{4}$ となる．ここで，log は単調増加関数であることから，これらを足した値は以下の関係を満たす．

$$N^2loglogN + \frac{N^2}{2}loglog\frac{N}{2} + \frac{N^2}{4}loglog\frac{N}{4} \cdots < \left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) N^2loglogN < 2N^2loglogN \quad (7)$$

つまり，2 回目以降の分割の計算量の合計は 1 回目の計算量よりも小さい．よって，平均計算量は $O(N^2loglogN)$ である．

次に，「(ii) スコア値が最大の分割点候補を選ぶ処理」の平均計算量を算出する．1 回目の分割ではグループのサイズが N なので，N 個の分割点候補分のスコア値から，最大のスコア値となる分割点候補を選ぶ処理になる．*secure k-nearest neighbor* [16] は，比較対象の要素数を M とおいたときに $O(M^2)$ の計算量であるため，1 回目の分割における計算量は $O(N^2) \times 1$，2 回目は $O((\mathrm{N}/2)^2) \times 2$，3 回目は $O((\mathrm{N}/4)^2) \times 4$ となる．よって先ほどと同様に，2 回目以降の分割の計算量の合計は 1 回目の計算量よりも小さい．ゆえに，最大のスコア値を選ぶ処理の計算量は $O(N^2)$ である．

以上より，ダミーユーザ手法の平均計算量は $O(N^2loglogN + N^2) = O(N^2loglogN)$ となる．なお，分割点候補ごとのスコア値の計算は並列化が可能であるため，適切に並列化を行うことで，ある程度の計算時間の低減が見込める．

6. 2. 2 平均通信量の算出

続いて平均通信量を算出する．まず「(i) 分割点候補ごとにスコア値を計算する処理」の平均通信量を算出する．*secure set intersection* [17] の通信量は，二つの集合の要素数を両方とも M とおいたとき $O(M)$ であるため，1 回目の分割での通信量は，サイズ N の一つのグループに対する通信を N 個の分割点候補分行うので $O(N) \times 1 \times N$，2 回目は $O(\mathrm{N}/2) \times 2 \times \mathrm{N}/2$，3 回目は $O(\mathrm{N}/4) \times 4 \times \mathrm{N}/4$ となる．よって先ほどと同様に，2 回目以降の分割の通信量の合計は 1 回目の通信量よりも小さいので，平均通信量は $O(N^2)$ である．同様に「(ii) スコア値が最大の分割点候補を選ぶ処理」の通信量を算出する．*secure k-nearest neighbor* は，比較対象の要素数を M とおいたときに $O(M^2)$ の通信量である [16]．これも，計算量のオーダー計算と同様に計算すると，$O(N^2)$ となる．結果，ダミーユーザ手法の平均通信量は $O(N^2 + N^2) = O(N^2)$ となる．

6.3 評価結果の考察

以上の評価結果より，提案手法を用いることでユーザ存在を隠ぺいしながらも相対誤差 15%以下でデータ分析が可能であることが確かめられた．今回は，機関 A，B における疾病の相関ルール分析を行うというユースケースを想定した評価であったが，同等のユースケースにおいても有効であると考える．

また，計算量と通信量についても既存のセキュア関数計算よりも大幅に増加することがないことが確かめられた．これにより，データ規模が大きくなければ，適切に並列化を行うことで提案手法を実際に動かすことが可能であると考える．

7. むすび

本論文では，垂直分割の分散匿名化において，双方の機関のユーザ集合が異なる場合に，ユーザ存在が知られてしまう問題を扱った．そして，ユーザ存在が推測される可能性を示した *δ-site-presence* という指標と，この指標を満たすための新たな分散匿名化手法としてダミーユーザ手法を提案した．また，提案手法を実際の患者のレセプトデータを用いて評価を行い，患者が通院しているか否かを隠ぺいしながらも相対誤差 15%以下でデータ分析が可能であることが確かめた．今後は，分割点決定関数の更なる改良を行い有効性の向上と，計算量・通信量の低減を図る予定である．

謝辞 本研究の一部は，経産省の「平成 23 年度次世代高信頼・省エネ型 IT 基盤技術開発・実証事業（レセプト情報等の利活用基盤の開発）」プロジェクトの成果である．

文　献

[1] 厚生労働省医薬品の安全対策における医療関係データベースの活用方策に関する懇談会，“電子化された医療情報データベースの活用による医薬品等の安全・安心に関する提言（日本のセンチネル・プロジェクト）について，” 2010.

[2] L. Sweeney, “k-anonymity: A model for protecting privacy,” Int. J. Uncertain. Fuzziness Knowl.-Based Syst., vol.10, pp.557–570, 2002.

[3] U.S. National Archives and Records Administration, “Standards for privacy of individually identifiable health information,” Federal Register, vol.67, no.157, pp.53182–53273, 2002.

[4] N. Mohammed, B.C.M. Fung, K. Wang, and P.C.K. Hung, “Privacy-preserving data mashup,” Proc. EDBT’09, pp.228–239, ACM, 2009.

[5] K. Wang, B.C.M. Fung, and G. Dong, “Integrating private databases for data analysis,” Proc. 2005 IEEE International Conference on Intelligence and Security Informatics (ISI), vol.3495, pp.171–182, 2005.

[6] W. Jiang and C. Clifton, “Privacy-preserving distributed k-anonymity,” Proc. DBSec’05, pp.166–177, Springer, 2005.

[7] P. Jurczyk and L. Xiong, “Distributed anonymization: Achieving privacy for both data subjects and data providers,” Proc. DBSec’09, pp.191–207, Springer, 2009.

[8] M.E. Nergiz, M. Atzori, and C. Clifton, “Hiding the presence of individuals from shared databases,” Proc. SIGMOD’07, pp.665–676, ACM, 2007.

[9] P. Samarati, “Protecting respondents’ identities in microdata release,” IEEE Trans. Knowl. Data Eng., vol.13, no.6, pp.1010–1027, 2001.

[10] Y. Lindell and B. Pinkas, “Secure multiparty computation for privacy-preserving data mining,” J. Privacy and Confidentiality, vol.1, pp.59–98, 2009.

[11] O. Goldreich, Foundations of Cryptography: Volume 2, Basic Applications, Cambridge University Press, 2004.

[12] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, “Mondrian multidimensional k-anonymity,” Proc. ICDE’06, p.25, IEEE, 2006.

[13] OpenID Foundation, “Openid authentication 2.0,” 2007.

[14] R. Agrawal and R. Srikant, “Privacy-preserving data mining,” Proc. SIGMOD’00, pp.439–450, ACM, 2000.

[15] A.C. Yao, “Protocols for secure computations,” Proc. SFCS’82, pp.160–164, IEEE Computer Society, 1982.

[16] J. Zhan, L. Chang, and S. Matwin, “Privacy preserving k-nearest neighbor classification,” International Journal of Network Security, vol.1, no.1, pp.46–51, 2005.

[17] M.J. Freedman, K. Nissim, and B. Pinkas, “Efficient private matching and set intersection,” Proc. EUROCRYPT’04, pp.1–19, Springer-Verlag, 2004.

[18] X. Xiao and Y. Tao, “m-invariance: Towards privacy preserving re-publication of dynamic datasets,” Proc. SIGMOD’07, pp.689–700, ACM, 2007.

[19] 竹之内隆夫，伊東直子，川村隆浩，大須賀昭彦，“クラウド上での事業者間データ連携のための分散型パーソナル情報保護エージェント，” 合同エージェントワークショップ &シンポジウム 2011(JAWS2011) 論文集，2011.

[20] O. Goldreich, S. Micali, and A. Wigderson, “How to play any mental game or a completeness theorem for protocols with honest majority,” Proc. STOC’87, pp.218–229, ACM, 1987.

（平成 24 年 6 月 4 日受付，10 月 3 日再受付）

竹之内隆夫　（正員）

2003 電気通信大・電気通信・情報工学卒．2005 同大大学院情報システム学研究科博士前期課程了．同年日本電気（株）入社．現在，情報・ナレッジ研究所主任．2011 年電気通信大学大学院情報システム学研究科博士後期課程入学．主としてパーソナル情報の利活用におけるプライバシー保護の研究に従事．情報処理学会会員．

川村　隆浩

1992 早大・理工・電気卒．1994 同大大学院理工学研究科電気工学専攻修士課程了．同年（株）東芝入社．現在，同社研究開発センター主任研究員．工博．2001〜2002 米国カーネギーメロン大学ロボット工学研究所客員研究員．2003 より電気通信大学大学院情報システム学研究科客員准教授．2007 より大阪大学大学院工学研究科非常勤講師．主としてマルチエージェントシステム，セマンティック Web の研究・開発に従事．情報処理学会，人工知能学会各会員．

大須賀昭彦　（正員）

1981 上智大・理工・数学卒．同年（株）東芝入社．同社研究開発センター，ソフトウェア技術センター等に所属．1985〜1989（財）新世代コンピュータ技術開発機構 (ICOT) 出向．2007 より電気通信大学大学院情報システム学研究科教授．工博（早稲田大学）．主としてソフトウェアのためのフォーマルメソッド，エージェント技術の研究に従事．1986 年度情報処理学会論文賞受賞．現在，IEEE Computer Society Japan Chapter Chair，人工知能学会理事．情報処理学会，人工知能学会，日本ソフトウェア科学会，IEEE CS 各会員．