Computer Security Symposium 2016
11 - 13 October 2016

ストリームデータに対する遅延のないk-匿名化の情報劣化を改善する方式

山岡 裕司[1,a)]　伊藤 孝一[1]

概要：近年，IoT 技術等の隆盛とプライバシー意識の高まりから，ストリームデータの匿名化への需要が高まっている．パーソナルデータの代表的な匿名化方法に k-匿名化があり，ストリームデータに対して遅延のない k-匿名化方式が提案されているが，初期データのみに基づいて匿名化する方式なため，初期データにない値は過剰に匿名化されてしまう課題がある．本稿では，その方式で使用される一般化関数を時間経過と共に円滑に自動更新し，過剰な匿名化を低減する改良方式を提案する．k-匿名化のベンチマークであるデータ Adult のレコードをストリームに見立てた実験により，提案方式は高速で情報劣化が小さいことを確認した．

Entropy Improvement on Delay-Free k-Anonymization of Streaming Data

Yamaoka Yuji[1,a)]　Itoh Kouichi[1]

Keywords: Privacy-Preserving Data Publishing, k-Anonymity, Data Stream

1. はじめに

近年，PPDP（Privacy-Preserving Data Publishing，プライバシー保護データ開示）技術への期待が高まっている [1]．PPDP とは，パーソナルデータ（個人に関する情報）について，プライバシーを保護できる範囲内でなるべく多くの情報を開示するためのデータ変換のことである．日本においては，2015 年 9 月に個人情報保護法の改正が決まり，「適切な規律の下で個人情報等の有用性を確保」するための規定が整備された．この規定では，「匿名加工情報を作成するとき」，すなわちある種の PPDP を実施するときは，「個人情報保護委員会規則で定める基準」に従わなければならないとされている．このように，日本でも PPDP は注目されている．

本稿が対象とするパーソナルデータは，他の多くの PPDP 研究 [2], [5], [6], [10], [13] と同様に，1 人 1 レコードに対応する 2 次元表であり，ミクロデータと呼ぶ．様々なミクロデータが開示されれば，それらをデータマイニングし分析することで，有用な知見が得られることが期待できる．一方で，ミクロデータからプライバシー侵害が起きないようにするには，PPDP が必要になる．たとえば，医療系のミクロデータは医療向上の研究に役立つが，個人を特定できるデータからは個人のプライバシーに関わる情報，たとえば病状などがわかってしまう．医療向上の研究では個人を特定できる必要がない場合もあり，その場合は PPDP を適用することが望ましい．

PPDP において達成すべきプライバシー保護のモデルとして研究が活発なものに，k-匿名性 [11] がある．k-匿名性とは，ミクロデータにおいて，どの個人も（準識別子と設定した属性組み合わせにおいて）同じレコードを持つ人が k 人以上いる，という性質である．ミクロデータを k-匿名化，つまり k-匿名性を満たすように変換すれば，ミクロデータ中のどの個人も一意に特定できないようになる．k-匿名性は，単独で個人識別情報となるような，氏名や個人番号と

[1] 株式会社富士通研究所
FUJITSU LABORATORIES LTD.
[a)] yamaoka.yuji@jp.fujitsu.com

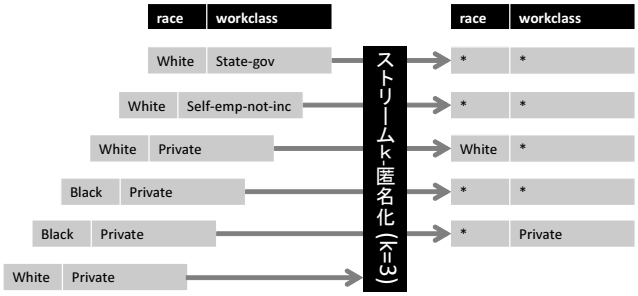


図 1: ストリームデータの k-匿名化
Fig. 1 k-Anonymization on Record Stream

いった属性を削除しただけでは保護が不十分であるとの考えの下に提案されてきた．その考えの根拠として，たとえば，性別と生年月日と 5 桁郵便番号の 3 属性の組み合わせにより，アメリカ国民の 6 割以上を一意に特定（identify）できるとする研究結果 [3], [11] がある．日本の個人情報保護法においては，特定の個人を識別できないデータは個人情報ではないとされていることから，k-匿名性のあるデータは個人情報とはみなし難く，その観点でも k-匿名性は注目されている．

また，近年は IoT 技術の発展により，ビッグデータをストリームで流通させ，順次高速に処理することを求められる場面が多い．そのため，ミクロデータをレコードのストリームとして収集し，レコード単位で順次匿名化できることへの需要が高まっている．たとえば，Zhou ら [14] は，ストリームデータに対する匿名化が有用な場面として，クレジットカードの詐欺検知の外注などを挙げている．しかし，従来の多くの k-匿名化方式 [2], [5], [6], [10], [13] は，変換の過程でミクロデータ全体を複数回走査する必要があり，そのままではストリームデータに対する適用が困難であるという課題がある．

そこで，山岡ら [16] は，ストリームデータを遅延なく高速に k-匿名化する方式を提案している．その方式は，最初に過去データを k-匿名化した後のデータを初期データとして受け付け，そのデータから，ストリームデータを一般化して k-匿名化（拡張された k-匿名性を達成）する一般化関数を生成し，その一般化関数を用いる方式である．しかし，初期データのみに基づいて匿名化するため，初期データに含まれていない値は何人分ストリーム入力しても一般化されてしまうという課題がある．なお，一般化とは情報をより広い概念に置換することであり，たとえば「21」歳を「20〜29」歳にしたり，抑制（墨塗り）したりする置換である．

図 1 は，ストリームデータの k-匿名化の様子で，ミクロデータをレコード単位で順次 k-匿名化している．セルの値「*」は，元の値を抑制したことを示している．

本稿では，山岡ら [16] の遅延のないストリーム k-匿名化方式の一般化関数を自動更新する方式を提案する．提案方式は，一般化する処理と，最新の一般化関数を生成する処理を並行して動かし，両処理を協調させることを特徴とする．前者である一般化処理は，ストリームデータを遅延なく高速に k-匿名化する他，過剰な匿名化をより少なくできる可能性のあるレコードを保存して生成処理にその旨を通知する．後者である生成処理は，通知が来た／来ていたら，保存されているデータから一般化関数を生成して一般化処理に渡し，また通知を待ち受ける．生成処理の計算量は多いが，一般化関数の生成中は通知を無視するため，いつまでも生成が終わらないようなことや，最新レコードが来るたびに必要とする計算資源が増加するようなことはない．

提案方式により，初期データの入力が不要になり，ストリーム自体から自動的に最新データを取り込んだ，過剰な一般化をすることがより少ない最新一般化関数に更新される．よって，長期運用することで情報劣化が小さい一般化を実現できる．

k-匿名化のベンチマークとして使われているデータを使って実験し，提案方式は情報劣化を抑えながら高速に匿名化できることを確認した．

2. 関連研究

ストリームデータに対する k-匿名化方式を，Zhou ら [14] や Mohammadian ら [9] が提案している．しかし，それらの研究では，後続のレコードが入力されるまでの時間が長いレコードは，出力までの遅延が大きくなるという課題がある．現実のレコードは人間の行動をきっかけに発生することが多く，そのストリームは入力間隔の長い時期が生じる場合が多い．そのため，現実的には遅延が生じやすいといえる．

ストリームデータに対して遅延なく，k-匿名性の拡張といわれる ℓ-多様性 [8] を達成する方式を，Kim ら [4] が提案している．しかし，その方式は虚偽のデータを出力するため，利用場面が限られるという課題がある．分析目的によっては虚偽のデータを出力しないことが重要であると，Fung ら [1] は主張している．

提案方式は遅延がなく，レコードの一般化しかしない方式である．後続のレコードの入力タイミングに非依存に匿名化し，虚偽のデータを出力しないため，場面を限らず利用することができる．

3. 遅延なしストリーム k-匿名化

提案方式で利用する，k-匿名化方式，および遅延なしストリーム k-匿名化方式について説明する．

3.1 k-匿名化

k-匿名性を満たすようにミクロデータを変換することを，k-匿名化 [10] という．多くの k-匿名化方式 [2], [5], [6], [10], [13] は，一般化のみをおこなう．

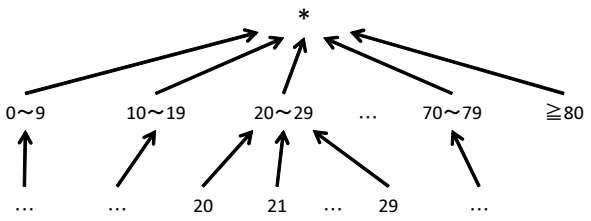


図 2: 一般化木の例

Fig. 2 Example of Generalization Tree

方式によっては，ミクロデータの各属性について数値属性かカテゴリ属性によって処理を変えるが，本稿では全属性をカテゴリ属性として扱う．数値属性は，トップ・コーディング [15] やグルーピング [15] などの一般化により容易にカテゴリ属性化できるため，カテゴリ属性に対応すれば全属性に対応できる．

カテゴリ属性に対する多くの k-匿名化方式は，各属性に対し一般化木の設定を要求する [5], [10], [13]．一般化木とは，当該属性の定義域について，各値がどのような一般化関係にあるのかを木構造で示したものである．根は抑制，つまり最も一般化された値に該当する．たとえば，年齢属性の定義域に，0〜79 歳の各年齢の他に，80 歳以上，10 歳単位の各年代，そして抑制があるとする．この場合，「21」歳の親は「20〜29」歳でさらにその親は「*」（抑制を示す値）であり，「≥ 80」歳の親は「*」である，などといった情報を**図 2** のように一般化木として設定するのが適当である．一般化は，一部（または全部）の属性の値を，各一般化木における祖先の値に置換することに相当する．

一般化は情報を劣化させるが，虚偽のデータにすることはない．

3.2 遅延なしストリーム k-匿名化

山岡ら [16] は，ストリームデータを高速に k-匿名化する方式を提案している．その方式は，最初に過去データを k-匿名化した後のデータを受け付け，そのデータから，ストリームデータを一般化して k-匿名化する一般化関数を生成し，その一般化関数を用いる方式である．一般化関数はレコードを入力とし，高速に一般化したレコードを出力できる．

ただし，適用できる k-匿名化方式に制限があり，各属性の値について無変換か抑制しかおこなわないものである必要がある．そのような k-匿名化方式は，各属性に対し一般化木の設定を要求する k-匿名化方式に，深さ 1 の一般化木を設定することで実現できる．

この方式を遅延なしストリーム k-匿名化方式と呼ぶ．

ストリーム k-匿名化が達成すべき性質は，伝統的な k-匿名性 [11] とは少し異なる．これについては，次節で説明する．

4. 用語および記法の定義

本稿で使用する用語および記法について説明する．

以下，ミクロデータの全属性集合を A とする．

定義 (属性値 $R[a]$)**.** ミクロデータにおける，レコード R の属性 $a \in A$ の値を属性値 $R[a]$ という．

定義 (レコードの同値関係 $=$)**.** 2 つのレコード R, R' が，$\forall a \in A, R[a] = R'[a]$ であるとき，$R = R'$ と書く．

また，$\exists a \in A, R[a] \neq R'[a]$ であるとき，$R \neq R'$ と書く．

定義 (一般化関係 $\rightarrow$)**.** 属性値 $R'[a]$ が，属性値 $R[a]$ を一般化した値のとき，この関係を属性値の一般化関係と呼び，$R[a] \rightarrow R'[a]$ と書く．

また，匿名化前のレコード R と，匿名化後のレコード R' が，$R \neq R'$ かつ $\forall a \in A, (R[a] = R'[a]) \vee (R[a] \rightarrow R'[a])$ の関係にあるとき，これをレコードの一般化関係と呼び，$R \rightarrow R'$ と書く．

定義 (マッチ $rightharpoonup$)**.** 匿名化前のレコード R と，匿名化後のレコード R' が，$R = R'$ または $R \rightarrow R'$ の関係にあるとき，それらのレコードはマッチしていると呼び，$R \rightharpoonup R'$ と書く．

定義 (抑制レコード $\emptyset$)**.** レコード R の全属性値が抑制値であるとき，R は抑制レコードと呼び，$R = \emptyset$ と書く．

定義 (レコード群対応 $M = (m_1, m_2, \ldots)$)**.** 匿名化前の n 個のレコード $R_1, R_2, \ldots, R_n$ と，それらの匿名化後のレコード $R'_1, R'_2, \ldots, R'_n$ について，匿名化前後のレコード同士が 1 対 1 にマッチする組の集合をレコード群対応と呼び，R' に対応する R の添え字の数列 M で表し，$M = (m_1, m_2, \ldots, m_n)$ と書く．たとえば，$n = 3$ で，$R_2 \rightharpoonup R'_1, R_3 \rightharpoonup R'_2, R_1 \rightharpoonup R'_3$ が成り立つ場合，$m_1 = 2, m_2 = 3, m_3 = 1$ であり $M = (2, 3, 1)$ となる．

定義 (照合候補 C)**.** 匿名化前の n 個のレコード $R_1, R_2, \ldots, R_n$ と，それらの匿名化後のレコード $R'_1, R'_2, \ldots, R'_n$ について，レコード群対応の i 番目の要素が取り得る値の集合をレコード R'_i の照合候補と呼び，C_i と書く．たとえば，$n = 3$ で，レコード群対応が $M_1 = (1, 2, 3)$ と $M_2 = (2, 1, 3)$ の 2 通りしかない場合，$C_1 = \{1, 2\}, C_2 = \{1, 2\}, C_3 = \{3\}$ となる．

4.1 ストリーム k-匿名性

遅延なしストリーム k-匿名化を設計するにあたり，伝統的な k-匿名性をそのまま使用するのは不適切である．なぜなら，1 レコードずつ出力する前提では，伝統的な k-匿名性を達成することは不可能なためである．たとえば最初の 1 レコード目を出力する時点でもう達成不可能である．

そこで，本稿ではストリーム k-匿名化されたデータが達成すべき性質であるストリーム k-匿名性を定める．

本稿では，k レコード以上の場合は Wong ら [12] の k-匿

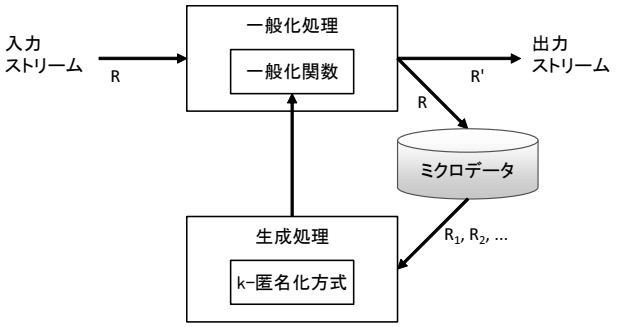


図 3: 提案方式
Fig. 3 Our Proposal

名性と同じとした．Wong らの k-匿名性の定義は，匿名化後の各レコードについて，匿名化前のレコード k 個未満に照合候補を絞り込めないこと，である．一方，k レコード未満の場合は，Wong らの k-匿名性を満たすことは不可能なため，匿名化後の各レコードが抑制レコードであること，とした．

定義 (ストリーム k-匿名性)**.** それまでに入出力した n 個のレコードに対し, 任意の $m \leq n$ について, 匿名化前の最初の m 個のレコード $R_1, R_2, \ldots, R_m$ と, それらの匿名化後のレコード $R'_1, R'_2, \ldots, R'_m$ について，$|C_i| \geq k \quad (i = 1, 2, \ldots, m)$ または $R'_i = \emptyset \quad (i = 1, 2, \ldots, m)$ を満たす場合，それらのレコードはストリーム k-匿名性を満たすと呼ぶ．

5. 提案方式

本稿では，遅延なしストリーム k-匿名化方式の一般化関数を自動更新する方式を提案する．元の遅延なしストリーム k-匿名化方式は，k-匿名化後のデータを初期データとして与えられるようになるまで運用できないことに加え，初期データに含まれていない値は何人分ストリーム入力しても一般化されてしまうという課題がある．そこで，ストリーム自体から得られる最新データから自動的に一般化関数を生成し直し，古い一般化関数を置き換える，という処理を繰り返すようにした．それにより，過剰な一般化が起きることが少なくなり，情報劣化が小さくなる．

提案方式は，一般化する処理と，一般化関数を生成する処理を並行して動かし，両処理を協調させることを特徴とする．前者である一般化処理は，遅延なしストリーム k-匿名化の一般化関数を使うため，高速に処理できる．後者である生成処理は，最新データから遅延なしストリーム k-匿名化の一般化関数を生成して一般化処理に渡す．**図 3** に両処理とデータの関係を示す．一般化関数を生成は時間がかかるが，一般化処理と並行しているため，一般化処理の高速性は保たれる．これらにより，最新データを一般化関数に取り込む，遅延なしストリーム k-匿名化を実現できる．

以降，提案方式がストリーム k-匿名性を達成すること，従来方式の課題，一般化処理の詳細，生成処理の詳細，の順に説明する．

5.1 ストリーム k-匿名性の達成

まず，次に説明する逐次方式がストリーム k-匿名性を達成することを説明し，その後一般化関数が自動更新される遅延なしストリーム k-匿名化方式もストリーム k-匿名性を達成することを説明する．

逐次方式とは，レコード R が入力されるたび，R と過去の全（匿名化前）レコードを合わせたミクロデータを k-匿名化し，R の匿名化後のレコード R' を出力する方式である．ただし，レコード数が k 未満のミクロデータに対しては，全てを抑制レコードに変換する．

逐次方式はストリーム k-匿名性を達成する．その理由は次の通りである．

逐次方式で n 個のレコードを匿名化した後の状況を考える．まず, $n < k$ の場合, 匿名化後のレコードは全て抑制レコードなため, ストリーム k-匿名性を達成している．次に, $n = k$ の場合，匿名化後のレコードは $n-1$ レコード目までは全て抑制レコードであり，レコード R'_n も k-匿名性により $R_1, R_2, \ldots, R_{n-1}$ の各レコードにマッチするため，明らかにストリーム k-匿名性を達成している．それら以外，すなわち $n > k$ の場合，まず，$n-1$ 個のレコードまでがストリーム k-匿名性を達成していると仮定する．その下で，$|C_n| \geq k$ を示せれば，数学的帰納法によりストリーム k-匿名性を達成しているといえる．その下で，匿名化前のレコード $R_1, R_2, \ldots, R_{n-1}$ と，それらの匿名化後のレコード $R'_1, R'_2, \ldots, R'_{n-1}$ のレコード $2(n-1)$ 個をノードとし，ノード R'_i から R_i と，C_i の各要素から $R'_i (i = 1, 2, \ldots, n-1)$ にエッジを引いた，有向グラフを考える．詳細は省略するが，R_1 から各 $R'_i \quad (i = 1, 2, \ldots, n-1)$ に至る経路が存在すれば $|C_n| \geq k$ が成立し，またそのような経路は必ず存在する．なお，n レコード目を k-匿名化したときのミクロデータ上での照合候補を C'_n（k-匿名性から $|C'_n| \geq k$ が成立）とすると，$C_n = C'_n$ となる．

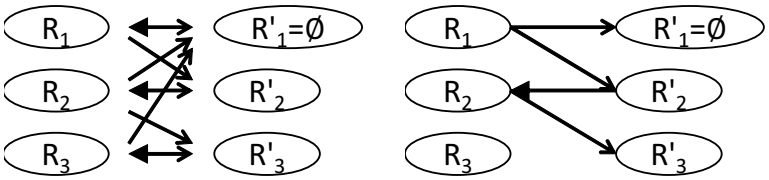
図 4 に例を示す．**図 4e** のように $m_n \neq n$ のレコード群対応の存在が重要であるが，**図 4b** のように R_1 から各 $R'_i \quad (i = 1, 2, \ldots, n-1)$ に至る経路があれば，そのようなレコード群対応は必ず存在する．

また，遅延なしストリーム k-匿名化方式もストリーム k-匿名性を達成する．一般化関数は過去の（匿名化前）レコードからなるミクロデータで k-匿名性を達成するような一般化をするため，逐次方式と同様に達成の理由を説明できる．

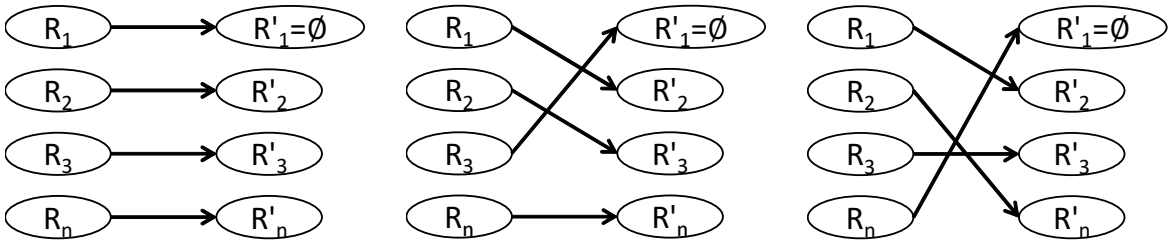
同様に，遅延なしストリーム k-匿名化方式は一般化関数を更新してもストリーム k-匿名性を達成する．

5.2 従来方式の課題

逐次方式の課題は，計算量である．ミクロデータの k-匿名化は，レコード数を n とすると $\Omega(n)$ のオーダーの計算量が必要なため，逐次方式は n レコードを処理するのに



(a) 有向グラフ　　(b) R_1 から各 R'_i $(i = 1, 2, \ldots, n-1)$ に至る経路



(c) レコード群対応 $M_1 = (1, 2, 3, 4)$　　(d) レコード群対応 $M_2 = (3, 1, 2, 4)$　　(e) レコード群対応 $M_3 = (4, 1, 3, 2)$

図 4: ストリーム k-匿名性を達成する様子（$k = 2, n = 4, C'_n = \{2, 4\}$）

Fig. 4 Achieving Stream-k-Anonymity ($k = 2, n = 4, C'_n = \{2, 4\}$)

$\Omega(n^2)$ のオーダーの計算量が必要である．

元の遅延なしストリーム k-匿名化方式の課題は，情報劣化である．初期データに含まれていない値は何人分ストリーム入力しても一般化されてしまう．大量の初期データを用意できたとしても，時間と共に傾向が変わるデータの場合，やはり初期データに含まれていない値が入力される可能性が高い．

提案方式は，遅延なしストリーム k-匿名化方式の一般化関数を自動更新することで，これらの課題を軽減する．一般化関数で高速にストリーム k-匿名化し，時間のかかる（ミクロデータの k-匿名化を伴う）一般化関数の生成はそれと並行して実施する．

5.3 一般化処理

一般化処理の処理手順は次の通りである．

(1) ストリームからのレコード R の入力を待ち受け．
(2) 一般化関数を最新版があれば更新．
(3) 一般化関数を R に適用し，R' を取得．
(4) $R \neq R'$ なら，R と通知 u を保存．
(5) R' を出力して (1) に戻る．

保存された R は，生成処理が最新の一般化関数を生成する際に使用される．$R \neq R'$ の場合，R を一般化する必要がないという情報がすでに現在の一般化関数に反映されているということである．そのため，この場合は R は一般化関数の更新に貢献しないため，保存しない．

一般化処理は高速に処理できる．一般化関数の更新と適用以外はレコード数に非依存な高速処理をおこなえる．一般化関数の更新と適用は，遅延なしストリーム k-匿名化方式であり，山岡ら [16] により高速に処理できることが実験で示されている．

5.4 生成処理

生成処理の処理手順は次の通りであり，一般化処理と並行して処理する．

(1) 通知 u を待ち受け，確認したら削除．
(2) 保存されているレコード群を取得し，ミクロデータの k-匿名化をおこない，一般化関数を生成．
(3) 生成した一般化関数を一般化処理に渡して，(1) に戻る．

一般化関数を生成は，遅延なしストリーム k-匿名化方式における初期データから一般化関数を生成する処理と同じである．ミクロデータの k-匿名化方式は，前述の通り，制限はあるものの，深さ 1 の一般化木を設定することで多くの k-匿名化方式を使用できる．

生成処理は計算量が多いため，一般的には生成処理の処理中に，一般化処理で新しいレコードや通知 u が何度も保存される．しかし，生成処理が処理中のときには通知により新しい処理が引き起こされることはないため，通知のたびに必要とする計算資源が増加するようなことはない．

6. 実験

提案方式を実際のデータに適用した結果，情報劣化を抑えながら高速にストリーム k-匿名化できることを確認した．以下，詳しく説明する．

適用対象のデータとして，k-匿名化のベンチマークとして使われている，UCI Machine Learning Repository[7] の Adult を用いた．レコードは，訓練データとテストデータをこの順に連結した，計 48,842 レコードを用いた．属性は，カテゴリ属性として良く使われる 5 つの属性 sex, race,

marital-status, native-country, workclass を用いた．一部のレコードは欠損値を含んでいる．本稿では，簡単のため，欠損値と抑制値を同一視した．欠損値は少ないため，これによる影響はほとんどないと考える．

データをレコードのストリームと見立てるため，1 秒あたり平均 100 レコードのポアソン過程とみなし，1 レコードずつ入力した．

比較対象として，逐次方式を実装した．逐次方式は計算量に課題があるが，情報劣化はストリーム k-匿名化において最小と考えらえる．

提案方式と逐次方式のそれぞれで使用する k-匿名化方式として，Xu らの方式（Top-Down 法）[13] を実装した．Xu らの方式は，カテゴリ属性を対象として高速性と情報量を両立する，代表的な k-匿名化方式と考えたためである．提案方式と逐次方式で同じ k-匿名化方式を使うことで，提案方式の効果が明確になりやすいと考えた．

適用環境は一般的な PC（Intel Core i7-2600 CPU @ 3.4GHz, RAM 16GB, 64bit Windows 7 Professional SP1）で，実装と適用は Java 8 でおこなった．

適用結果を**図 5** に示す．図 5 で，提案方式は「this paper」，逐次方式は「sequential」である．**図 5a**, **図 5b** より，提案方式は逐次方式に比べ十万倍ほど速く処理できたことがわかる．なお，レイテンシ―とは，レコードを入力してから出力が得られるまでの時間である．また，**図 5c**, **図 5d** より，情報劣化は提案方式と逐次方式にほとんど差がない（図では重なって見える）ことがわかる．つまり，提案方式は情報劣化をほぼ最小に抑えられたことになる．

以上より，提案方式は高速で情報劣化が小さいためストリーム k-匿名化に効果的といえる．

7. まとめ

本稿では，山岡ら [16] の遅延のないストリーム k-匿名化方式の一般化関数を自動更新する方式を提案した．一般化関数の適用を高速に保つため，時間のかかる処理である一般化関数の生成を，一般化関数の適用に並行しておこなうのが特徴である．元の方式と違い，最新データにより一般化関数の生成は繰り返しおこなわれるため，ストリーム開始時から適用でき，長期運用することで情報劣化の少ない一般化を実現できる．k-匿名化のベンチマークとして使われているデータ Adult を使って実験し，提案方式は高速で情報劣化が小さいことを確認した．

参考文献

[1] Fung, B. C. M., Wang, K., Chen, R. and Yu, P. S.: Privacy-preserving data publishing: A survey of recent developments, *ACM Comput. Surv.*, Vol. 42, pp. 14:1–14:53 (online), DOI: http://doi.acm.org/10.1145/1749603.1749605 (2010).

[2] Ghinita, G., Karras, P., Kalnis, P. and Mamoulis, N.: Fast Data Anonymization with Low Information Loss, *Proceedings of the 33rd International Conference on Very Large Data Bases*, VLDB '07, VLDB Endowment, pp. 758–769 (2007).

[3] Golle, P.: Revisiting the Uniqueness of Simple Demographics in the US Population, *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, WPES '06, New York, NY, USA, ACM, pp. 77–80 (online), DOI: 10.1145/1179601.1179615 (2006).

[4] Kim, S., Sung, M. K. and Chung, Y. D.: A framework to preserve the privacy of electronic health data streams, *Journal of biomedical informatics*, Vol. 50, pp. 95–106 (2014).

[5] LeFevre, K., DeWitt, D. J. and Ramakrishnan, R.: Incognito: efficient full-domain K-anonymity, *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, SIGMOD '05, New York, NY, USA, ACM, pp. 49–60 (online), DOI: 10.1145/1066157.1066164 (2005).

[6] LeFevre, K., DeWitt, D. J. and Ramakrishnan, R.: Mondrian Multidimensional K-Anonymity, *Proceedings of the 22nd International Conference on Data Engineering*, ICDE '06, Washington, DC, USA, IEEE Computer Society, pp. 25– (online), DOI: 10.1109/ICDE.2006.101 (2006).

[7] Lichman, M.: UCI Machine Learning Repository (2013).

[8] Machanavajjhala, A., Kifer, D., Gehrke, J. and Venkitasubramaniam, M.: L-diversity: Privacy beyond k-anonymity, *ACM Trans. Knowl. Discov. Data*, Vol. 1 (online), DOI: http://doi.acm.org/10.1145/1217299.1217302 (2007).

[9] Mohammadian, E., Noferesti, M. and Jalili, R.: FAST: Fast Anonymization of Big Data Streams, *Proceedings of the 2014 International Conference on Big Data Science and Computing*, BigDataScience '14, New York, NY, USA, ACM, pp. 23:1–23:8 (online), DOI: 10.1145/2640087.2644149 (2014).

[10] Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression, *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, Vol. 10, pp. 571–588 (online), DOI: 10.1142/S021848850200165X (2002).

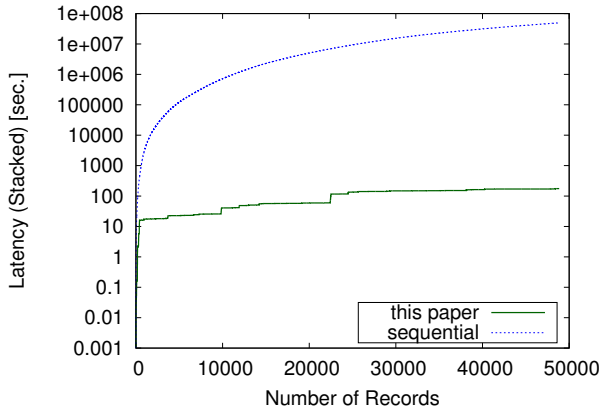
[11] Sweeney, L.: k-anonymity: a model for protecting privacy, *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, Vol. 10, pp. 557–570 (online), DOI: 10.1142/S0218488502001648 (2002).

[12] Wong, W. K., Mamoulis, N. and Cheung, D. W. L.: Non-homogeneous Generalization in Privacy Preserving Data Publishing, *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, New York, NY, USA, ACM, pp. 747–758 (online), DOI: 10.1145/1807167.1807248 (2010).

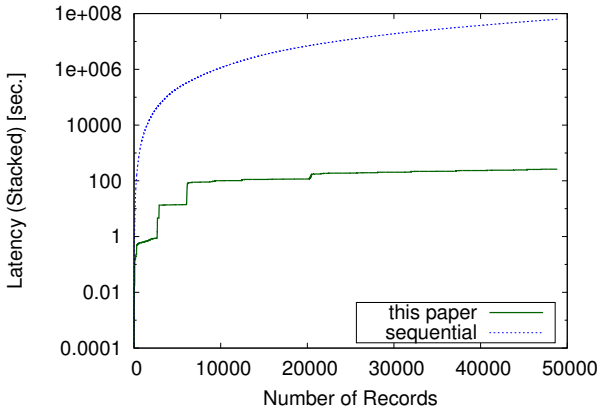
[13] Xu, J., Wang, W., Pei, J., Wang, X., Shi, B. and Fu, A. W.-C.: Utility-based Anonymization Using Local Recoding, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, New York, NY, USA, ACM, pp. 785–790 (online), DOI: 10.1145/1150402.1150504 (2006).

[14] Zhou, B., Han, Y., Pei, J., Jiang, B., Tao, Y. and Jia, Y.: Continuous Privacy Preserving Publishing of Data Streams, *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, New York, NY, USA, ACM, pp. 648–659 (online), DOI: 10.1145/1516360.1516435 (2009).

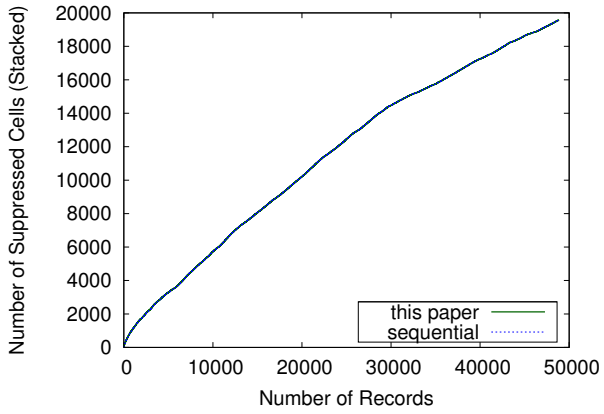
[15] 総務省：匿名データの作成・提供に係るガイ



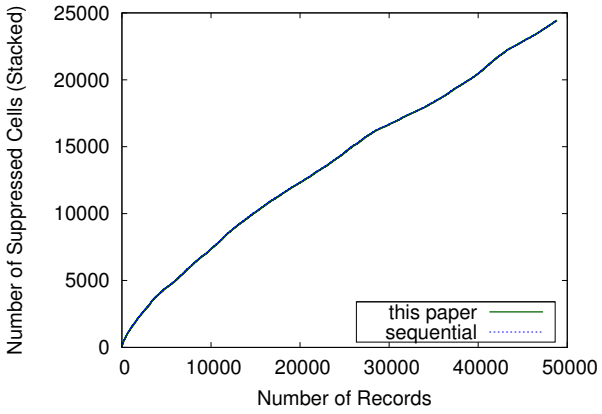
(a) レコード数とレイテンシー（累積，対数軸）の関係（$k = 3$）



(b) レコード数とレイテンシー（累積，対数軸）の関係（$k = 5$）



(c) レコード数と抑制値（セル）数（累積）の関係（$k = 3$）



(d) レコード数と抑制値（セル）数（累積）の関係（$k = 5$）

図 5: Adult への適用結果

Fig. 5 Result on Adult

ドライン，`http://www.soumu.go.jp/main_content/000398971.pdf`. 2018 年 8 月 4 日参照.

[16] 山岡裕司，伊藤孝一：ストリームデータに対する遅延のない k-匿名化方式，2016 年暗号と情報セキュリティシンポジウム概要集 (2016).