匿名化グループ間の要素数の変化を比較可能な匿名化手法の実現

豊田　由起†　宮川　伸也†　側高　幸治†　伊東　直子†

†NEC サービスプラットフォーム研究所
211-8666　神奈川県川崎市中原区下沼部 1753

あらまし　個人のデータを活用する際に生じるプライバシー侵害の問題を解決する手法として、属性を組み合わせたときに個人が特定できないようにデータを加工する匿名化がある。匿名化手法の一つに、ある匿名性の指標を満たすために属性を抽象化する方法がある。情報損失を最小にする従来の手法では、抽象化の対象となるレコードの数を最小にするため、匿名化後の属性の値が異なる場合がある。その結果、属性の値ごとのレコード数の差分を正しく比較ができない問題がある。さらに、データの増加に伴って匿名化を行う際に、抽象化後の属性の値が同じレコード数の増分を正しく把握できない問題もある。本稿では、情報の値が異なるレコード数の差分や、属性の値が同じレコード数の増分を把握できるような匿名化手法を提案して評価を行う。

An anonymization method allowing the comparison of the numbers of records in anonymized group

Yuki Toyoda† Shinya Miyakawa† Koji Sobataka† Naoko Ito†

†Service Platforms Research Laboratories, NEC Corporation
1753 Shimonumabe, Nakahara-Ku, Kawasaki 211-8666, JAPAN

Abstract Anonymization plays a significant role as a solution to preserve user's privacy when utilizing the user data. In order to prevent the breach of the privacy information, some techniques convert an input data table into one satisfying k-anonymity by processing the data. One of the anonymization methods is Generalization. It generalizes the attributes to satisfy the indicator of anonymity. The Generalization process is optimized so that the number of generalized record and generalization level are minimized at individual process. Thus, multiple records with same attribute value and the same records over the multiple processes can be generalized differently. This makes it difficult to compare the difference in the number of the records with the same attributes value after the generalization process. Moreover, it is unable to trace the increased number of record which has same attribute value in incremental datasets. In this paper, we propose and evaluate an algorithm that enables the comparison of the increment and the difference of records in incremental data set.

1　はじめに

企業や病院等によって収集されたユーザの情報をより活用するための一つの方法として、第三者へ公開することで二次利用することが考えられる。

ユーザの情報には、背景知識などを考慮したり組み合わせたりすることで個人を特定しうる情報である準識別子と、他人に知られたくない情報であるセンシティブ属性が付随していることが多い。例えば、郵便番号や性別、年齢などが準識別子であり、病名や年収などがセンシティブ属性である。公開する情報に病歴や病状のセンシティブな情報が含まれる場合、個人のプライバシーに留意しなければならない。

プライバシーを保護するための手法の一つとし

て匿名化がある。匿名化とは、複数のユーザ情報が含まれるデータに対して、同じ準識別子の組み合わせのユーザ情報を特定の条件を満たすように加工する処理のことをいう。また、加工する方法の一つに属性を抽象化する方法がある。

従来手法では、抽象化するレコードの数や抽象度を最小にする匿名化を行う。そのため、レコードによって、属性の値が同じにも関わらず匿名化後の抽象度が異なる。その結果、属性の値ごとのレコード数の差分を正しく比較ができない問題がある。

また、従来手法ではデータの増加に伴って匿名化を行う度に、抽象化の対象を最小にして情報損失を最小にする。そのため、レコードによって抽象度が毎回異なる可能性があり、抽象化後の属性の値が同じレコード数の増分を正しく把握できない問題がある。

例えば、患者のカルテ情報には年齢・性別・喫煙歴・病名が記載されていているとする。また、「30代・女性・喫煙者」に該当する患者数は匿名性を満たす基準以下とする。

従来手法では、「30 代・女性・喫煙者」に該当する患者数が匿名性の基準未満の場合、個人が特定される可能性が高い。その解決の一例として、匿名性の基準以上の「30 代・女性・非喫煙者」に該当する患者と、匿名性の基準未満の「30 代・女性・喫煙者」に該当する患者のカルテ情報を「30代・女性」と抽象化する。しかし、匿名性の基準以上の「30 代・女性・非喫煙者」に該当する患者の少なくとも一部は、匿名性の基準未満の「30 代・女性・非喫煙者」に該当する患者のレコードと抽象化するため、患者数の増分を把握できなくなる問題がある。

また、匿名性の基準未満の「30代・女性・喫煙者」に該当する患者のプライバシー保護のために、これらの患者のレコードを切り落とす手法もある。しかし、匿名性の基準未満の全てのレコードを切り落としてしまうと統計情報が変わる問題がある。

本稿では、増加するデータを対象に、属性の値ごとのレコード数の変化の保持が可能な匿名化アルゴリズムを提案して評価する。

2 関連研究

k-匿名性[1]とは、抽象化などにより準識別子の値が同じレコードの数を k 以上にするための指標である。*k*-匿名性を満たすために以下の手法がある。

準識別子の値が同じレコードの数が k 未満のレコードを削除する切り落とし[1]がある。しかし、データを切り落としてしまうため、増加するデータに対して準識別子の値が同じレコードの増分を把握できない問題がある。

また、準識別子の値が同じレコードの数を k 以上にするために、準識別子の値を抽象化する手法がある。抽象化には、グローバルリコーディング[2]とローカルリコーディング[3]がある。

グローバルリコーディングでは、匿名化処理により準識別子の値が同じレコードの数が k 未満の場合、そのレコードは切り落とされるか、あるレコードを抽象化する際に、その属性の値と同じ値の属性からなる全てのレコードの属性を抽象化する。そのため、情報損失が大きい問題がある。

一方、ローカルリコーディングは、あるレコードを抽象化する際にその属性の値と同じ属性からなる、一部のレコードの属性を抽象化することにより、情報の損失を抑制している。

また、増加するデータを匿名化処理する際、途中でレコードが追加される場合がある。そこでローカルリコーディングを適用した場合、匿名化を行う度に抽象化するレコードの数を最小にする。その結果、属性の値が同じレコード増分を把握できない問題がある。

また、データの更新後、新たな準識別子の挿入などにより準識別子の値が同じレコードの数が*k*未満の場合に、ダミーレコードを追加する手法である。更新したデータに対する代表的な匿名化手法の一つが、ダミー情報を追加することにより*m*-不変性[4]を満たす手法である。

この手法は、センシティブ情報が ℓ 種類存在するようにレコードをまとめてグループとし、準識別子の値を抽象化する ℓ-多様性[5]を拡張した手法である。更新されたデータを再匿名化した場合にも、更新前と同じ種類数でかつ、全く同じセンシティブ属性になることを保証する。更新されたグループ

のセンシティブ属性が m-不変性を満たさない場合はダミーデータを追加する。そのため、データの総数が増加してしまい、統計情報が変化してしまう問題がある。

3 問題定義

増加するデータを対象に変化を保持する匿名性を定義する。

3.1 匿名性

時刻 t におけるデータを $T(t)$、匿名化後のデータを $T^*(t)$とする。また、$T(t)$に属するレコードを r とし、レコード r の準識別子の値を $QI_t(r)$、$T^*(t)$に属するレコード r の準識別子の値を $QI_t^*(r)$とする。

定義 1: (匿名性)

$T^*(t)$に属する全ての種類の $QI_t^*(r)$において、$QI_t^*(r_1)=QI_t^*(r_2)=\ldots=QI_t^*(r_n)$を満たすレコードの数が $n \geqq k$ である時、$T^*(t)$は匿名性を満たすとする。

3.2 異なる属性値間での差分比較

抽象化後のデータで、属性の値が同じレコードの数と、抽象化前のデータで、抽象化後の属性の値が同じレコードの差を、異なる属性の値で比較可能とする。

定義 2:(異なる属性値間での差分比較)

匿名化前後の $T(t)$と $T^*(t)$で属性の値が同じになるレコード $QI_t(r_n)=QI_t^*(r_n)$において、$QI_t(r_i)=\ldots=QI_t(r_n)=\ldots=QI_t(r_j)$ $(i \leqq n \leqq j)$を満たすレコードの数を α_tとする。また $QI_t^*(r_k)=\ldots=QI_t^*(r_n)=\ldots=QI_t^*(r_l)$ $(k \leqq n \leqq l)$を満たすレコードの数を α_t'する。

また、異なる属性の値で、かつ、抽象化前後で属性の値が同じになるレコード $QI_t(r_n) \neq QI_t(r_m)=QI_t^*(r_m)$において、$QI_t(r_p)=\ldots=QI_t(r_m)=\ldots=QI_t(r_q)$ $(p \leqq m \leqq q)$を満たすレコードの数を β_tとする。また $QI_t^*(r_s)=\ldots=QI_t^*(r_m)=\ldots=QI_t^*(r_t)$ $(s \leqq m \leqq t)$を満たすレコードの数を β_t'とする。このとき、$|\alpha_t|-|\alpha_t'|=|\beta_t|-|\beta_t'|$を満たせば、属性値が異なるレコードの数の差分比較が可能とする。

3.3 同じ属性値の増分比較

時刻 t の抽象化前のデータにおいて、抽象化後の属性の値が同じレコードの数(α_t)と、データが増加後の抽象化前のデータにおいて、抽象化後の属性の値が同じレコードの数(α_{t+n})の差と、抽象化後のデータにおいて、属性の値が同じレコードの数(α_t')と、データが増加後の抽象化後のデータにおいて、属性の値が同じレコードの数(α_{t+n}')の差を比較し、レコード数の増分が比較可能である。

定義 3:(同じ属性値の増分比較)

時刻 $t+n$ で、抽象化前後で属性の値が同じになる $QI_{t+n}(r_n)=QI_{t+n}^*(r_n)$において、$QI_{t+n}(r_i)=\cdots=QI_{t+n}(r_n)=\cdots=QI_{t+n}(r_j)$ $(i \leqq n \leqq j)$を満たすレコードの数を α_{t+n} とする。また、$QI_{t+n}^*(r_k)=\cdots=QI_{t+n}^*(r_n)=\cdots=QI_{t+n}^*(r_l)$ $(k \leqq n \leqq l)$を満たすレコードの数を α_{t+n}'とする。この時、$|\alpha_{t+n}|-|\alpha_t|=|\alpha_{t+n}'|-|\alpha_t'|$を満たせば、属性値同士のレコード数の増分の比較が可能とする。

3.4 変化を保持する匿名性

定義 4:(変化を保持する匿名性)

匿名化後の $T^*(t)$と $T^*(t+n)$が、定義 1 を満たした上で定義 2 と定義 3 を満たすことを、変化を保持する匿名性とする。

4 提案アルゴリズム

変化を保持する匿名性を満たすためのアルゴリズムについて述べる。

4.1 アルゴリズム

提案アルゴリズムはローカルリコーディングを基にする。匿名性を満たした上で、異なる属性値のレコード数の増分を比較可能にし、かつ、属性値のレコード数の増分比較を可能にするために、抽象化するレコード数を全ての属性の値および全ての時刻で一定にする。その結果、変化を保持する匿名性を満たす。

アルゴリズムは、1:抽象化レコード数の計算対象となるリーフの探索、2:抽象化レコード数の決定、3:抽象化 から構成される。

4.2 リーフの探索

タクソノミーツリーに基づいて抽象化するレコード数を決定する。タクソノミーツリーは単一継承の木構造であり、ツリーのノードには準識別子の値のラベルがついている。また、ツリーの上部は抽象的概念であり、ツリーの下部へ行くほど具体化される。

上部のノードは子ノードの概念を含む。

準識別子を一段階抽象化した際に、抽象化した準識別子の値(リーフ Y)が同じになるレコードの数を全てのリーフ$\{Y_1 \cdots Y_n\}$に対して数える。

一段階抽象化したら、準識別子の値が同じレコードの数が k 未満のリーフ Y_n が存在した場合、そのリーフ Y_n をさらにもう一段階抽象化した際に同じになるリーフ Z_i を起点に 4.1.2 節以降の処理を行う。

4.3 抽象化レコード数の決定

リーフ Z_i に対して、リーフ$\{Y_1 \cdots Y_n\}$の中から抽象化した際のレコード数を計算する方法を下記に述べる。

レコード数が k 未満となるリーフが存在する場合、それらのリーフのレコード数を加算する。加算した結果が k 未満の場合、k に不足する数を計算して、リーフのレコードが k 以上で、かつ、抽象化したらリーフ Z になるリーフの数で割って、抽象化するレコードの数(以下、p)を求める。これらの処理を全てのリーフについて行い$\{p_1 \cdots p_m\}$を算出する。

p_iを算出するアルゴリズムを図 1 に示す。

入力:タクソノミーツリー(*tree*),ユーザ情報(*user_info*),匿名化指標(*k*)
出力: 抽象化要素数($p_z \in Z$);

```
01  for( Y1から Yn){
02      if (レコード数が k 未満のリーフが存在){
03          レコード数が k 未満のリーフのレコードの和を
            求める
05          if(求めた和が k 未満){
06              x=k-(k 未満のリーフのレコード数の和)
07              pi=roundup(x÷非特異点集合の数)
08          }
09      }
10  }
```

図 1 p_iを算出するアルゴリズム

4.4 抽象化

識別子の値を、タクソノミーツリーに基づいて一つ抽象度上げて、ラベルを付けかえる。ラベルを付けかえるレコードの数は、4.1.2 節で求めた抽象化レコードの数にする。

準識別子の値が同じレコードの数が $k+p$ 未満の場合は、該当するレコードの準識別子の値の全てを対象にする。この処理をタクソノミーツリーの最下層ノードから行う。

定理:抽象化レコードの数が全ての準識別子の値で一定ならば、属性値間のレコードの数の差分の比較が可能である。

証明:

抽象化前のデータにおいて、抽象化後に属性の値が同じになるレコードの数を α_t、抽象化後のデータにおいて、属性の値が同じレコードの数をα_t'とする。同様に、異なる属性の値について β_t,β_t'とする。

抽象化するレコードの数を一定にするので、

$|\alpha_t'|-p=|\alpha_t|$・・・(式 1)

同様に、

$|\beta_t'|-p=|\beta_t|$・・・(式 2)

よって、(式 1)と(式 2)より、

$(|\alpha_t|+p)-(|\beta_t|+p)=|\alpha_t'|-|\beta_t'|$

$|\alpha_t|-|\beta_t|=|\alpha_t'|-|\beta_t'|$

よって、抽象化レコードの数が全ての準識別子の値で一定ならば、属性値間のレコードの数の差分の比較が可能であることが証明される■

また、時間 n が経過してデータが増加した後の 2 回目以降の匿名化処理では、初回に計算した p を用いて匿名化処理を行う。尚、2 回目の匿名化処理では、p を初回と同じにするだけではなく、選択するレコードも同じにする。

定理:すべての時刻において、常に抽象化レコード数が一定ならば、ある属性の値をもつレコードの数の増分の比較が可能である

証明:

時刻 t において、抽象化前のデータで、抽象化したら属性の値が同じになるレコードの数をα_t、抽象化後のデータで、属性の値が同じレコードの数を α_t'とする。同様に、時刻 $t+n$ において、α_{t+n}、α'_{t+n}とする。

上記の処理により、

$|\alpha_t|-p=|\alpha'_t|$・・・(式 3)

$|\alpha_{t+n}|-p=|\alpha'_{t+n}|$・・・(式 4)

よって、(式 3)(式 4) より、

$|\alpha_{t+n}|-|\alpha_t|=|\alpha_{t+n}'|-|\alpha_t'|$

よって、すべての時刻において、常に抽象化レコード数が一定ならば、ある属性の値をもつレコードの数の増分の比較が可能である■

また、準識別子が同じレコードの数が基準$(k+p)$以上になった場合、準識別子の値をタクソノミーツ

リーに沿って一つ具体化する。

抽象化アルゴリズムを図 2 に示す。

入力:タクソノミーツリー(*tree*),ユーザ情報(*user_info*),匿名化指標(*k*),対象ノード(Z_i),抽象化要素数(*p*),抽象化回数(*c*)
出力: 抽象化済みユーザ情報(*ano_user_info*)

```
01   for(  全ての属性の値  ){
02       QI(r1)=…QI(rn)となるレコードの数を計算
03       if(n < k+p ){
04           該当するレコードの属性を全て抽象化
05       }else{
06           if(初回){
07               該当するレコードの中から p 個の要素を選択
08               選択した要素の準識別子を抽象化
09           }else{
10               初回時に選択した p 個のレコードを選択
11               選択した要素の準識別子を抽象化
12           }
13       }
14   }
```

図 2　抽象化アルゴリズム

4.5　具体例

まず *p* の算出例を述べる。図 3 のように、自宅最寄り駅である「中目黒」「自由が丘」「緑が丘」と、それらをさらに抽象化した概念として、区名の「目黒区」が定義されているとする。以下の例では *k*=12 とする。また、匿名化前の「中目黒」「自由が丘」「緑が丘」に所属する人数を、200 人、100 人及び 8 人とする。

この中で、準識別子が「緑が丘」のレコードの数は 8 であるため、*k* まで満たない数は 12-8=4 である。そこで、*k* と準識別子が *k* 未満の準識別子の数は、「中目黒」「自由が丘」の 2 つなので、目黒区に付随する *p* は、4÷2＝2 となる。

次に抽象化の具体例を示す。各準識別子から抽象化要素を取り出した後に、基準値(*k*+*p*=12+2=)14 以上の人数を含む、「中目黒」と「自由が丘」の 2 人を「目黒区」に抽象化する。一方、「緑ヶ丘」の人数は 14 人以下なので 8 人全員を「目黒区」に抽象化する。

次に、「中目黒」「自由が丘」「緑が丘」の人数が、210 人、120 人及び 10 人に増加したとする。*p*=2 であるため「中目黒」「自由が丘」から「目黒区」に 2 人ずつ抽象化し、「緑が丘」が準識別子となるレコードの数は、(*k*+*p*=)14 を満たさないため、12 人全員を抽象化する。

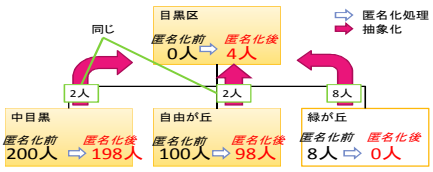

図 3　抽象化の具体例

4.6　効果

時刻 t_0 において、匿名化前後の「中目黒」と「自由が丘」の人数差は、それぞれ(200－100=)100 人及び (198－98=)100 人となり、匿名化前後で人数の差分を維持することができ、匿名化後のデータが定義 2 を満たす。

また、時刻 t_1 での匿名化前後の「中目黒」の人数の増加数は、それぞれ、(210－200=)10 人及び (208－198=)10 人となり、人数の増分を維持することができ、匿名化後のデータが定義 3 を満たす。

5　評価

各定義の性質をどの程度満たすかの評価を行うために実験と評価を行った。定義 2 については 5.1 節、定義 3 について 5.2 節で述べる。

尚、各実験に用いたユーザ情報は、疑似的に生成した。ユーザ 1 人につき 2 つの属性(最寄り駅と年収)を与えた。また、最寄り駅は準識別子、年収をセンシティブ属性とした。ユーザ情報は、それぞれ、乗降者人数、国勢調査を参考にした。また、匿名化処理をするときに必要なタクソノミーツリーは、地域メッシュ[6]の階層構造にしたがっている。匿名化指標の *k* は 10 にした。

5.1　異なる属性値間での差分比較評価

レコードを抽象化した際に、準識別子が同じ値をもつレコードの数を、異なる準識別子の値で比較可能かの評価を行うために実験をした。ユーザ数は 10,000 人生成して、従来手法のローカルリコーディングと比較した。

まず、*p*=5 にして実験と評価を行った。その後、*p* の値を 1≦*p*＜*k*(10)で変動させて、*p* の値ごとに匿名化を行い、評価した。

- *p*=5 に固定した実験と評価

匿名化処理後に、定義 2 を満たすノードの割合は、従来手法では約 96%だったのが、提案手法に

より100%まで改善することができた。

- p を変動させたときの実験と評価

次に p の値を $1 \leqq p \leqq 9$ で変動させて、定義2を満たす準識別子の値の割合の評価を行った。結果、p の値を変動させても、定義3を満たすノードの割合は100%を維持できることがわかる。

5.2 同じ属性値の増分比較評価

次に、ユーザ数を30,000人に増やして、データが増加した場合に定義3を満たす割合の評価を行った。結果、従来手法は98%、提案手法は100%の割合で定義3を満たした。

5.3 プライバシー性と有用性

匿名化処理を行うにあたり、考慮しなければならない情報損失として、(1) 変化を保持する匿名性を満たなくなることによる情報損失 (2)準識別子の抽象化による情報損失 (3)切り落としによる情報損失 がある。

(1)は、提案手法により、定義2と定義3はそれぞれ 100%満たすことが可能になり情報損失を最小にできた。

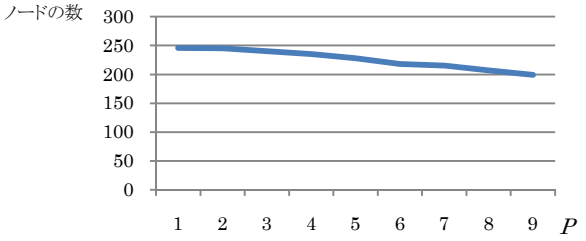

図4 p 変動時のノード数

(2)は、表1に示すように、従来手法の方が匿名化後のデータで、準識別子の値の種類が多くなった。これは、匿名化により、準識別子が抽象化されてまとめられるためである。つまり、準識別子の値の種類が多いことは、抽象度が低いことを示す。したがって、従来手法の方が情報損失が少ない。また、p を変動させた場合の評価を図4に示す。このグラフより、p の値が小さいほど、(2)の情報損失が少なくなったことがわかる。

表1 p=5 の時の情報損失

	従来手法	提案手法
(1)比較できない準識別子の値の数	12 個	0 個
(2)匿名化後の準識別子の値の数	267 個	228 個
(3)切り捨てされる準識別子の値の数	0 個	0 個

(3)は、表1よりどちらも、切り捨てされるレコードの数が0個であるため、従来手法でも提案手法でも変わらないことがわかる。

(1)(2)(3)に分類した情報損失の考察の結果、レコードの抽象化後に、準識別子が同じ値をもつレコードの数と、準識別子の別の値になるレコードの数の比較時に、提案手法が有効であることがわかった。

本稿の評価では p が大きくなるほど、情報が抽象化された。しかし、子ノードの数が少ないノードが多数存在するタクソノミーツリーを用いての匿名化時に、同じ属性の値のレコード数が k 未満となる属性の種類が少なく、かつ、十分に小さい場合、p の値を小さくすると、切り捨てされる。

6 おわりに

本研究では、属性値間のレコード数の差分比較と、属性値のレコード数の増分比較を定義して、変化を保持する匿名化アルゴリズムを提案した。また、疑似データを用いて提案アルゴリズムの有効性を評価した。その結果、全ての準識別子の値で変化を保持できることがわかった。

謝辞

本研究の一部は、総務省「平成22年度大規模仮想化サーバ環境における情報セキュリティ対策技術の研究開発」の一環として実施した。

参考文献

[1]. P.Samarati, Protecting Respondents'Identities in Microdata Release, IEEE Trans. on Knowl. And Data Eng. 13(6), pp.1010-1027, 2001.

[2]. K.LeFevre, Incognito:Efficient Full-Domain KAnonymity, ACM SIGMOD Int'l Conf. on Management of Data, pp.49-60, 2005

[3]. J.Xu, Utility-Based Anonymization Using LocalRecoding, ACM SIGKDD Int'l Conf. on Kowledge discovery and datamining, pp.785-790, 2006

[4]. X Xiao, m-Invariance: Towards Privacy apreserving Re-publication of Dynamic Datasets, ACM SIGMOD'07

[5]. A Machanavajihala,, L-diversity: Privacy beyond k-anonymity, Journal ACM Transaction on Knowledge Discoverty frin Data, Vol1, Issue1, March 2007

[6]. 総務省統計局, "地域メッシュ", http://www.stat.go.jp/data/mesh/index.htm