(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6973636号
(P6973636)
(45)発行日　令和3年12月1日(2021.12.1)　(24)登録日　令和3年11月8日(2021.11.8)

(51)Int.Cl.　F I
G06F 21/62 (2013.01)　G06F 21/62 345

請求項の数 6 (全 11 頁)

(21)出願番号　特願2020-518220(P2020-518220)
(86)(22)出願日　平成31年4月17日(2019.4.17)
(86)国際出願番号　PCT/JP2019/016447
(87)国際公開番号　WO2019/216137
(87)国際公開日　令和1年11月14日(2019.11.14)
審査請求日　令和2年10月26日(2020.10.26)
(31)優先権主張番号　特願2018-89641(P2018-89641)
(32)優先日　平成30年5月8日(2018.5.8)
(33)優先権主張国・地域又は機関
日本国(JP)

(73)特許権者　000004226
日本電信電話株式会社
東京都千代田区大手町一丁目５番１号
(74)代理人　100121706
弁理士　中尾　直樹
(74)代理人　100128705
弁理士　中村　幸雄
(74)代理人　100147773
弁理士　義村　宗洋
(72)発明者　長谷川　聡
東京都千代田区大手町一丁目５番１号　日本電信電話株式会社内

審査官　宮司　卓佳

最終頁に続く

(54)【発明の名称】安全性評価装置、安全性評価方法、およびプログラム

(57)【特許請求の範囲】
【請求項１】
複数のレコードからなる元データベースと上記元データベースを秘匿した秘匿データベースとを記憶するデータベース記憶部と、
上記秘匿データベースの各レコードについて、上記元データベースに対する近傍探索により所定の近傍数の近傍レコード集合を取得する近傍レコード探索部と、
上記秘匿データベースの各レコードについて上記近傍レコード集合の各レコードとの距離を計算し、当該レコードとの距離に基づいて最近傍レコードを取得する最近傍レコード計算部と、
上記秘匿データベースの各レコードについて、当該レコードに対応する上記元データベースのレコードが上記最近傍レコードと一致するか否かに基づいて当該レコードの再識別率を計算する再特定判定部と、
上記秘匿データベースの各レコードについて計算した再識別率に基づいて上記秘匿データベースの再識別率を計算する再識別率計算部と、
を含む安全性評価装置。
【請求項２】
請求項１に記載の安全性評価装置であって、
上記近傍レコード集合のレコード数が所定の閾値よりも多い場合に上記近傍レコード集合中の重複レコードを排除する重複排除部をさらに含む、
安全性評価装置。

【請求項３】
請求項２に記載の安全性評価装置であって、
上記近傍レコード探索部は、上記近傍数を上記元データベースのレコード件数の対数として上記近傍レコード集合を取得するものであり、
上記重複排除部は、上記閾値を上記近傍数の２倍として上記近傍レコード集合中の重複レコードを排除するものである、
安全性評価装置。
【請求項４】
請求項１から３のいずれかに記載の安全性評価装置であって、
上記近傍レコード探索部は、kd木を用いる近傍探索により上記近傍レコード集合を取得するものである、
安全性評価装置。
【請求項５】
データベース記憶部に、複数のレコードからなる元データベースと上記元データベースを秘匿した秘匿データベースとが記憶されており、
近傍レコード探索部が、上記秘匿データベースの各レコードについて、上記元データベースに対する近傍探索により所定の近傍数の近傍レコード集合を取得し、
最近傍レコード計算部が、上記秘匿データベースの各レコードについて上記近傍レコード集合の各レコードとの距離を計算し、当該レコードとの距離に基づいて最近傍レコードを取得し、
再特定判定部が、上記秘匿データベースの各レコードについて、当該レコードに対応する上記元データベースのレコードが上記最近傍レコードと一致するか否かに基づいて当該レコードの再識別率を計算し、
再識別率計算部が、上記秘匿データベースの各レコードについて計算した再識別率に基づいて上記秘匿データベースの再識別率を計算する、
安全性評価方法。
【請求項６】
請求項１から４のいずれかに記載の安全性評価装置としてコンピュータを機能させるためのプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
この発明は、データベースに対して決定的手法もしくは確率的手法により個別データを秘匿したデータベースの安全性を評価する技術に関する。
【背景技術】
【０００２】
データベース（以下、「元データベース」と呼ぶ）に対して決定的手法により個別データを秘匿する技術として、k-匿名法（非特許文献１および２参照）がある。また、確率的手法により秘匿する技術として、Pk-匿名法（非特許文献３および４参照）がある。これらの秘匿処理を施したデータベース（以下、「秘匿データベース」と呼ぶ）の安全性を評価するために、レコードリンケージと呼ばれる手法（非特許文５および６参照）が用いられる。レコードリンケージとは、あるレコードを再特定しようとすることで、どれだけそのレコードが秘匿できているかを測定する方法である。従来技術では、再特定を試みる秘匿データベースの対象レコードと元データベースの全レコードとの距離を計算し、最近傍レコードと対象レコードとが一致したら再特定できたとして、最近傍レコード数の逆数を対象レコードの再識別率とする。これを秘匿データベースの全レコードについて実施し、各レコードの再識別率を合計した値をデータベースの再識別率として評価する。
【先行技術文献】
【非特許文献】
【０００３】

【非特許文献１】Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan, "Incognito: Efficient full-domain k-anonymity", Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 49-60, 2005.
【非特許文献２】Florian Kohlmayer, Fabian Prasser, Claudia Eckert, Alfons Kemper, and Klaus A Kuhn, "Flash: efficient, stable and optimal k-anonymity", Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom), pp. 708-717, 2012.
【非特許文献３】五十嵐大，千田浩司，高橋克巳，“数値属性における，k-匿名性を満たすランダム化手法”，コンピュータセキュリティシンポジウム2011，pp. 450-455，2011年
【非特許文献４】五十嵐大，千田浩司，高橋克巳，“k-匿名性の確率的指標への拡張とその適用例”，コンピュータセキュリティシンポジウム2009，pp. 1-6，2009年
【非特許文献５】Vicenc Torra, John M Abowd, and Josep Domingo-Ferrer, "Using mahalanobis distance-based record linkage for disclosure risk assessment", International Conference on Privacy in Statistical Databases, pp. 233-242, 2006.
【非特許文献６】Josep Domingo-Ferrer and Vicenc Torra, "Distance-based and probabilistic record linkage for re-identification of records with categorical variables", Butlleti de IACIA, Associacio Catalana dIntelligencia Artificial, pp. 243-250, 2002.

【発明の概要】
【発明が解決しようとする課題】
【０００４】
近年ビッグデータの利活用が注目されており、匿名化の対象となるデータも大規模データとなることが想定される。従来技術では、レコードリンケージの際に、レコード数が増えるに連れて処理時間が増えることが問題であった。より具体的には、レコード数の線形な増加に伴い、処理時間が２乗で増えてしまう。したがって、大規模なデータに対し、実用的な処理時間でレコードリンケージを行うことが課題であった。
【０００５】
この発明は、上記のような技術的課題に鑑みて、大規模なデータを秘匿したデータベースの安全性を効率的に評価することを目的とする。
【課題を解決するための手段】
【０００６】
上記の課題を解決するために、この発明の一態様の安全性評価装置は、複数のレコードからなる元データベースと元データベースを秘匿した秘匿データベースとを記憶するデータベース記憶部と、秘匿データベースの各レコードについて、元データベースに対する近傍探索により所定の近傍数の近傍レコード集合を取得する近傍レコード探索部と、秘匿データベースの各レコードについて近傍レコード集合の各レコードとの距離を計算し、当該レコードとの距離に基づいて最近傍レコードを取得する最近傍レコード計算部と、秘匿データベースの各レコードについて、当該レコードに対応する元データベースのレコードが最近傍レコードと一致するか否かに基づいて当該レコードの再識別率を計算する再特定判定部と、秘匿データベースの各レコードについて計算した再識別率に基づいて秘匿データベースの再識別率を計算する再識別率計算部と、を含む。
【発明の効果】
【０００７】
この発明によれば、レコードリンケージを行う際に、従来技術では$O(N^2)$の計算量を要する処理が、近傍数を$\log N$とした場合には$O(N \log N)$の計算量となる。そのため、大規模なデータに対し、実用的な処理時間でレコードリンケージを行うことができる。したがって、大規模なデータを秘匿したデータベースの安全性を効率的に評価することができる。
【図面の簡単な説明】

【０００８】
【図１】図１は、本発明で対象とするデータベースの定義を説明するための図である。
【図２】図２は、従来のレコードリンケージを説明するための概念図である。
【図３】図３は、本発明のレコードリンケージを説明するための概念図である。
【図４】図４は、実施形態の安全性評価装置の機能構成を例示する図である。
【図５】図５は、実施形態の安全性評価方法の処理手続きを例示する図である。
【発明を実施するための形態】
【０００９】
以下、この発明の実施の形態について詳細に説明する。なお、図面中において同じ機能を有する構成部には同じ番号を付し、重複説明を省略する。
【００１０】
［記号］
ある属性の集合を大文字Xと表現し、属性Xの値を小文字$x \in X$と表現する。
【００１１】
データベースの１レコードを横ベクトルとして表現する。例えば、M属性あるデータベースのi番目のレコードは、$\vec{x}_i=\{x_{i1}, \cdots, x_{ij}, \cdots, x_{iM}\}$とする。
【００１２】
複数のレコードからなる集合をデータベース$\vec{X}$とする。例えば、レコード数Nのデータベースは、$\vec{X}=\{\vec{x}_1, \cdots, \vec{x}_N\}$とする。
【００１３】
データベース$\vec{X}$の各レコードを決定的手法もしくは確率的手法により秘匿したレコードからなる集合を秘匿データベース$\vec{Y}$とする。例えば、レコード数Nの秘匿データベースは、$\vec{Y}=\{\vec{y}_1, \cdots, \vec{y}_N\}$とする。
【００１４】
秘匿データベース$\vec{Y}$はレコードの順番がシャッフルされている場合もある。そこで、秘匿データベース$\vec{Y}$の行番号と元データベース$\vec{X}$の行番号（以下、「真の行番号」と呼ぶこともある）とを対応付ける行番号対応関数$f_y:R \rightarrow R$を定義する。
【００１５】
図１に、元データベース$\vec{X}$、秘匿データベース$\vec{Y}$、および行番号対応関数f_yの例を示す。元データベース$\vec{X}$は、M属性からなる平文のレコードをNレコード含むデータベースである。秘匿データベース$\vec{Y}$は、元データベースの各レコードが秘匿され、かつ、順番がシャフルされたデータベースである。行番号対応関数f_yは、元データベースの行番号と秘匿データベースの行番号との対応が表された参照表である。
【００１６】
［処理の概要］
本発明の安全性評価技術では、元データベース$\vec{X}$と秘匿データベース$\vec{Y}$と行番号対応関数f_yとを用いて、データベース全体の再識別率を計算し、安全性の評価を行う。本発明では、あるレコードの再識別率を計算するにあたり、大まかに以下の２つの処理を行う。
【００１７】
処理１．近傍探索の対象となるレコードの近傍レコードを、指定した近傍数分取得する。近傍レコードの探索は、木構造を用いたもの（参考文献１参照）や、ハッシングを用いたもの（参考文献２参照）があり、それらを用いて近傍レコードを取得する。木構造としては、例えばkd木等が挙げられる。
【００１８】
〔参考文献１〕Jon Louis Bentley, "Multidimensional binary search trees used for associative searching", Communications of the ACM, Vol. 18, No. 9, pp. 509-517, 1975.
〔参考文献２〕Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions", In Proceedings of the twentieth annual symposium on Computational geometry, pp. 253-262,

2004.

【００１９】

処理２．近傍レコードに基づくレコードリンケージの対象レコードと近傍レコードとの距離を計算し、最も距離が近い近傍レコードの行番号とレコードリンケージの対象レコードの真の行番号とが一致したら再特定できたとする。計算する距離としては、例えば、ユークリッド距離、ハミング距離、マンハッタン距離等、適切な距離を用いることができる。レコードの属性値が重複している場合、近傍探索で指定した近傍数以上の近傍レコードが取得される。その場合には、近傍レコード中の重複レコードを排除した上で、近傍レコードとの距離を計算する。

【００２０】

上記処理１，２を秘匿データベースの各レコードについて行い、各レコードの再識別率の合計値をデータベースの再識別率とし、秘匿データベース全体の安全性を評価する。

【００２１】

図２は、従来技術によるレコードリンケージを表す概念図であり、図３は、本発明によるレコードリンケージを表す概念図である。従来技術は秘匿データベースのあるレコードについて元データベースの全レコードとの距離を計算し、最も近いレコードの行番号がそのレコードの真の行番号と一致した場合に、再特定できたものと判定する。一方、本発明は秘匿データベースのあるレコードについて元データベースから近傍探索により取得した所定の近傍数の近傍レコードとの距離を計算し、最も近いレコードの行番号がそのレコードの真の行番号と一致した場合に、再特定できたものと判定する。

【００２２】

本発明では、木構造を用いた近傍探索もしくはハッシングを用いた近傍探索のどちらかを用いることとする。本発明の具体的な処理を<Algorithm 1>に示す。上記の処理１（近傍探索）は２～７行目に対応し、処理２（近傍レコードに基づくレコードリンケージ）は８～22行目に対応する。なお、|・|は集合・の要素数を表す。

【００２３】

<Algorithm 1>近傍探索を用いたレコードリンケージ

Input: レコード数Nの元データベース$\vec{X}=\{\vec{x}_1, \cdots, \vec{x}_N\}$，レコード数Nの秘匿データベース$\vec{Y}=\{\vec{y}_1, \cdots, \vec{y}_N\}$，行番号対応関数$f_y:R\rightarrow R$，近傍数K（1<K<N），許容範囲ε（ε>1）

Output: 再識別率r

1: r←0
2: for i=1 to N do
3: $\vec{y}_i$に対する元データベース$\vec{X}$の近傍レコード集合$\vec{X}_i^{near}=\{\vec{x}_j\}$（ただし、$|\vec{X}_i^{near}|\geqq K$）を近傍探索により取得する
4: end for
5: if $|\vec{X}_i^{near}|>\varepsilon K$となるiが存在する場合 then
6: $|\vec{X}_i^{near}|>\varepsilon K$となるiに対して、$\vec{X}_i^{near}$のうち重複を除いたレコード集合$\vec{X}_i^{uniq}=\{\vec{x}_j\}$とし、各$\vec{x}_j$に対応する重複レコードの行番号集合を返す関数$f^i_{dup}$を保持する
7: end if
8: for i=1 to N do
9: if $|\vec{X}_i^{near}|>\varepsilon K$となる場合 then
10: $\vec{y}_i$と$\vec{X}_i^{uniq}$の各レコードとの距離を求め、$\vec{y}_i$に最も距離の近いレコードを最近傍レコード$\vec{Z}$とする
11: if $\vec{Z}$に$f_y(k)=i$（ただしk:$\vec{x}_k\in\vec{Z}$）となるレコードが存在する場合 then
12: for $\vec{x}_j\in\vec{Z}$ do
13: $r\leftarrow r+1/(|f^i_{dup}(j)||\vec{Z}|)$
14: end for

15: end if
16: else
17: $\vec{y}_i$と$\vec{X}_i^{near}$の各レコードとの距離を求め、$\vec{y}_i$に最も距離の近いレコードを最近傍レコード$\vec{Z}$とする
18: if $\vec{Z}$に$f_y(k)=i$（ただしk:$\vec{x}_k \in \vec{Z}$）となるレコードが存在する場合 then
19: $r \leftarrow r+1/|\vec{Z}|$
20: end if
21: end if
22: end for

【００２４】

まず、秘匿データベース$\vec{Y}$のレコード$\vec{y}_i$ごとに近傍探索を用いて近傍レコード集合$\vec{X}_i^{near}$を取得する（２～４行目に対応）。元データベース$\vec{X}$中に重複したレコードが少なければ、取得した近傍レコード集合$\vec{X}_i^{near}$が指定した近傍数K以下となる。しかしながら、元データベース$\vec{X}$中に重複するレコードが多い場合、取得した近傍レコード集合$\vec{X}_i^{near}$が指定した近傍数Kを超えることがあり、近傍探索した効果がなくなってしまう。そこで、近傍レコード集合$\vec{X}_i^{near}$がεK件を超えた場合は、近傍レコード集合$\vec{X}_i^{near}$中の重複するレコードを排除したレコード集合$\vec{X}_i^{uniq}$を生成し、重複するレコードの行番号を返す関数f^i_{dup}を保持する（５～７行目に対応）。なお、許容範囲εと近傍数Kは、例えば、$\varepsilon=2.0$, $K=\log N$などに設定するとよい。

【００２５】

次に、秘匿データベース$\vec{Y}$のレコード$\vec{y}_i$ごとに近傍レコード集合の各レコードとの距離を計算し、最も距離が近いレコードの真の行番号（すなわち、元データベース上の行番号）が現在のレコードの真の行番号（すなわち、現在のレコードの秘匿データベース上の行番号に対応付けられた元データベース上の行番号）と一致した場合、再識別成功として再識別率rを加算する。

【００２６】

より具体的には、近傍レコード集合$\vec{X}_i^{near}$のレコード数がεK件を超えていた場合は、まず重複を排除した近傍レコード集合$\vec{X}_i^{uniq}$の各レコードとの距離計算を行う。そして、最も距離が近いレコードの重複するレコードの真の行番号を探索し、現在のレコードの行番号と一致した場合、重複レコード間で平均して再識別できたとして、1/重複レコード数（$1/|f^i_{dup}|$）を再識別率として加算する。その際、最近傍レコードが複数ある場合は、それらも平均して（$1/|\vec{Z}|$）再識別できたとして加算する（９～15行目に対応）。

【００２７】

もし近傍レコード集合$\vec{X}_i^{near}$のレコード数がεK件以下であった場合は、まず近傍レコード集合$\vec{X}_i^{near}$の各レコードとの距離計算を行う。そして、最も距離が近いレコードの重複するレコードの真の行番号を探索し、現在のレコードの行番号と一致した場合、１を再識別率として加算する。その際、最近傍レコードが複数ある場合は、それらを平均して（$1/|\vec{Z}|$）再識別できたとして加算する（16～21行目に対応）。

【００２８】

［実施形態］

実施形態の安全性評価装置および方法は、上記<Algorithm 1>を実行して秘匿データベースの安全性を評価する。実施形態の安全性評価装置１は、図４に例示するように、データベース記憶部１０、近傍レコード探索部１１、重複排除部１２、最近傍レコード計算部１３、再特定判定部１４、および再識別率計算部１５を備える。この安全性評価装置１が、図５に例示する各ステップの処理を行うことにより実施形態の安全性評価方法が実現される。

【００２９】

安全性評価装置１は、例えば、中央演算処理装置（CPU: Central Processing Unit）、主記憶装置（RAM: Random Access Memory）などを有する公知又は専用のコンピュータに

特別なプログラムが読み込まれて構成された特別な装置である。安全性評価装置１は、例えば、中央演算処理装置の制御のもとで各処理を実行する。安全性評価装置１に入力されたデータや各処理で得られたデータは、例えば、主記憶装置に格納され、主記憶装置に格納されたデータは必要に応じて中央演算処理装置へ読み出されて他の処理に利用される。安全性評価装置１の各処理部は、少なくとも一部が集積回路等のハードウェアによって構成されていてもよい。安全性評価装置１が備える各記憶部は、例えば、RAM（Random Access Memory）などの主記憶装置、ハードディスクや光ディスクもしくはフラッシュメモリ（Flash Memory）のような半導体メモリ素子により構成される補助記憶装置、またはリレーショナルデータベースやキーバリューストアなどのミドルウェアにより構成することができる。

【００３０】

以下、図５を参照して、実施形態の安全性評価装置１が実行する安全性評価方法について説明する。

【００３１】

データベース記憶部１０には、平文のレコード$\vec{x}_i$（i=1，…，N、N≧2）からなる元データベース$\vec{X}=\{\vec{x}_1, \cdots, \vec{x}_N\}$と、元データベース$\vec{X}$を秘匿した秘匿データベース$\vec{Y}=\{\vec{y}_1, \cdots, \vec{y}_N\}$と、元データベース$\vec{X}$の行番号と秘匿データベース$\vec{Y}$の行番号とを対応付ける行番号対応関数$f_y$が記憶されている。

【００３２】

ステップＳ１１において、近傍レコード探索部１１は、秘匿データベース$\vec{Y}$の各レコード$\vec{y}_i$について、元データベース$\vec{X}$に対する近傍探索により所定の近傍数Kの近傍レコード集合$\vec{X}_i^{near}=\{\vec{x}_j\}$（$j\in\{1, \cdots, N\}$）を取得する。このとき、近傍数Kは、例えば、元データベース$\vec{X}$のレコード件数Nの対数log Nとする。近傍探索は、木構造もしくはハッシングを用いた近傍探索のどちらかを用い、例えば、kd木を用いる手法を用いる。近傍レコード探索部１１は、取得した近傍レコード集合$\vec{X}_i^{near}$を重複排除部１２へ出力する。

【００３３】

ステップＳ１２において、重複排除部１２は、近傍レコード集合$\vec{X}_i^{near}$のレコード数が所定の閾値εKよりも多い場合に、近傍レコード集合$\vec{X}_i^{near}$中の重複レコードを排除して、重複排除済み近傍レコード集合$\vec{X}_i^{uniq}$を生成する。このとき、閾値εKは、例えば、近傍数の２倍（すなわち、許容範囲ε=2.0）とする。重複排除部１２は、重複排除済み近傍レコード集合$\vec{X}_i^{uniq}$を最近傍レコード計算部１３へ出力する。近傍レコード集合$\vec{X}_i^{near}$のレコード数が閾値εK以下だった場合は、近傍レコード集合$\vec{X}_i^{near}$を最近傍レコード計算部１３へ出力する。

【００３４】

ステップＳ１３において、最近傍レコード計算部１３は、秘匿データベース$\vec{Y}$の各レコード$\vec{y}_i$について、近傍レコード集合$\vec{X}_i^{near}$のレコード数が閾値εKよりも多かった場合には、重複排除済み近傍レコード集合$\vec{X}_i^{uniq}$の各レコードとの距離を計算し、近傍レコード集合$\vec{X}_i^{near}$のレコード数が閾値εK以下だった場合には、近傍レコード集合$\vec{X}_i^{near}$の各レコードとの距離を計算し、当該レコード$\vec{y}_i$との距離が最も近い最近傍レコード$\vec{Z}=\{\vec{x}_k\}$（$k\in\{1, \cdots, N\}$）を取得する。最近傍レコード計算部１３は、取得した最近傍レコード$\vec{Z}$を再特定判定部１４へ出力する。

【００３５】

ステップＳ１４において、再特定判定部１４は、秘匿データベース$\vec{Y}$の各レコード$\vec{y}_i$について、当該レコード$\vec{y}_i$に対応付けられた元データベース$\vec{X}$のレコード$\vec{x}_j$が最近傍レコード$\vec{Z}$中に存在するか否かに基づいて当該レコード$\vec{y}_i$の再識別率r_iを計算する。レコード$\vec{y}_i$に対応付けられたレコード$\vec{x}_j$は行番号対応関数f_yを用いて求めることができる。再特定判定部１４は、計算したレコード$\vec{y}_i$の再識別率r_iを再識別率計算部１５へ出力する。

【００３６】

ステップＳ１５において、再識別率計算部１５は、秘匿データベース$\vec{Y}$の各レコード$\vec{y}_i$について計算した再識別率r_iに基づいて秘匿データベース$\vec{Y}$の再識別率rを計算する。例えば、秘匿データベース$\vec{Y}$の各レコード$\vec{y}_i$の再識別率r_iの総和$\Sigma_{i=1}^{N} r_i$を秘匿データベース$\vec{Y}$の再識別率rとする。再識別率計算部１５は、秘匿データベース$\vec{Y}$の再識別率rを安全性評価装置１の出力とする。

【００３７】

本形態のポイントは、レコードリンケージに対して近傍探索を単に組み合わせただけでは解決できない課題、すなわち、近傍レコードが大量に出現した場合の問題を解決したことである。具体的には、大量の近傍レコードに対して重複排除処理を加えることで、処理時間を抑えたことである。近傍レコードを取得する際に属性値が重複等している場合、指定した近傍数以上の近傍レコードを取得するため、そのまま処理を行うと実行時間が長くなる。最悪の場合、近傍レコードがデータベース中のレコード数分出力されてしまい、結果として近傍レコードを探索した効果がなくなってしまう。本形態では、近傍レコードを取得する際に重複排除の処理を加えていることから、上記問題を回避でき、高速な実行が可能となっている。

【００３８】

以上、この発明の実施の形態について説明したが、具体的な構成は、これらの実施の形態に限られるものではなく、この発明の趣旨を逸脱しない範囲で適宜設計の変更等があっても、この発明に含まれることはいうまでもない。実施の形態において説明した各種の処理は、記載の順に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的にあるいは個別に実行されてもよい。

【００３９】

［プログラム、記録媒体］

上記実施形態で説明した各装置における各種の処理機能をコンピュータによって実現する場合、各装置が有すべき機能の処理内容はプログラムによって記述される。そして、このプログラムをコンピュータで実行することにより、上記各装置における各種の処理機能がコンピュータ上で実現される。

【００４０】

この処理内容を記述したプログラムは、コンピュータで読み取り可能な記録媒体に記録しておくことができる。コンピュータで読み取り可能な記録媒体としては、例えば、磁気記録装置、光ディスク、光磁気記録媒体、半導体メモリ等どのようなものでもよい。

【００４１】

また、このプログラムの流通は、例えば、そのプログラムを記録したDVD、CD-ROM等の可搬型記録媒体を販売、譲渡、貸与等することによって行う。さらに、このプログラムをサーバコンピュータの記憶装置に格納しておき、ネットワークを介して、サーバコンピュータから他のコンピュータにそのプログラムを転送することにより、このプログラムを流通させる構成としてもよい。

【００４２】

このようなプログラムを実行するコンピュータは、例えば、まず、可搬型記録媒体に記録されたプログラムもしくはサーバコンピュータから転送されたプログラムを、一旦、自己の記憶装置に格納する。そして、処理の実行時、このコンピュータは、自己の記憶装置に格納されたプログラムを読み取り、読み取ったプログラムに従った処理を実行する。また、このプログラムの別の実行形態として、コンピュータが可搬型記録媒体から直接プログラムを読み取り、そのプログラムに従った処理を実行することとしてもよく、さらに、このコンピュータにサーバコンピュータからプログラムが転送されるたびに、逐次、受け取ったプログラムに従った処理を実行することとしてもよい。また、サーバコンピュータから、このコンピュータへのプログラムの転送は行わず、その実行指示と結果取得のみによって処理機能を実現する、いわゆるASP（Application Service Provider）型のサービスによって、上述の処理を実行する構成としてもよい。なお、本形態におけるプログラムには、電子計算機による処理の用に供する情報であってプログラムに準ずるもの（コンピ

ュータに対する直接の指令ではないがコンピュータの処理を規定する性質を有するデータ等）を含むものとする。

【００４３】

また、この形態では、コンピュータ上で所定のプログラムを実行させることにより、本装置を構成することとしたが、これらの処理内容の少なくとも一部をハードウェア的に実現することとしてもよい。

【図１】

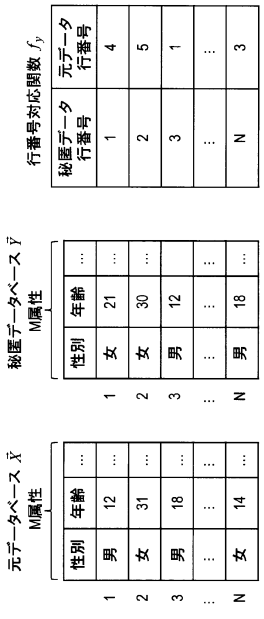

図1

【図２】

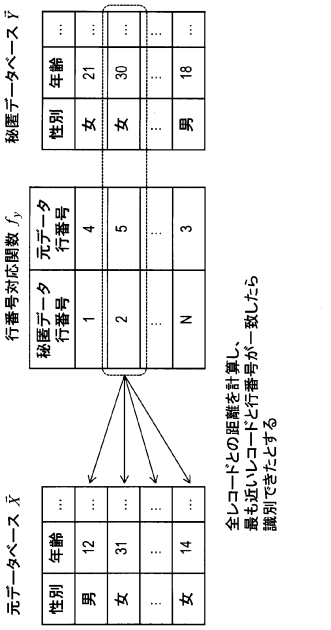

図2

【図３】

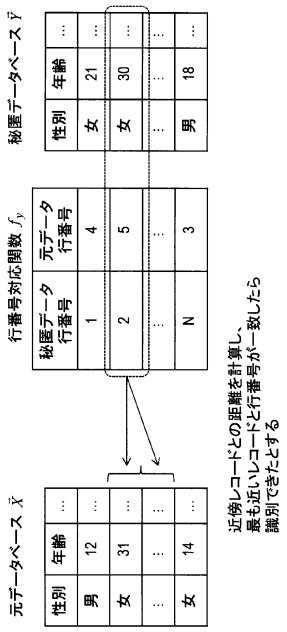

図3

【図４】

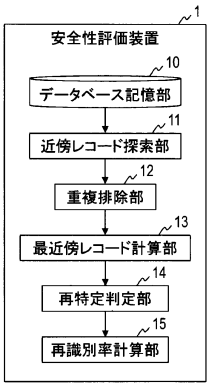

図4

【図５】

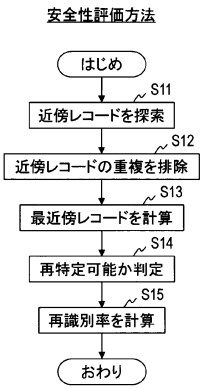

図5

フロントページの続き

(56)参考文献 特開２０１８－４９４３７（ＪＰ，Ａ）
国際公開第２００７／１３２５６４（WO，Ａ１）
伊藤聡志,菊池浩明，ユークリッド距離を用いた再識別手法とＰＷＳＣｕｐ２０１５の匿名加工データを用いた評価，電子情報通信学会技術研究報告 Ｖｏｌ．１１６ Ｎｏ．６５ IEICE Technical Report，日本，一般社団法人電子情報通信学会，2016年05月19日，第116巻,第65号，p.145-p.152

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ２１／６２