情報処理学会論文誌　データベース　Vol.6　No.3　50–60 (June 2013)

推薦論文

プライベート問合せにおける問合せ頻度を用いた制約緩和手法

川本 淳平[1,a)]　ギレット パトリシア[2]　佐久間 淳[1]

受付日 2012年12月20日, 採録日 2013年4月6日

概要：本論文では，プライベート問合せにおける問合せの頻度を用いた制約緩和手法を提案する．プライベート問合せとは，データベースの利用者が何を問い合わせているのか，すなわち検索意図を隠したまま目的のアイテムを取得する問合せ手法である．既存手法の多くは，i) サーバは実際の値を知ることはできないが問合せ処理のみ実行できる形式に問合せおよび問合せ結果を符号化する，ii) サーバは問合せ処理時にすべてのアイテムを走査する，という 2 条件を用いてプライベート問合せを実現している．その結果，サーバが攻撃者となる場合であっても，どのアイテムが実際に問い合わせられたのかを隠すことができる．しかし，この 2 つ目の条件によりサーバにおける問合せ処理コストはサーバが保持するアイテムの総数を n として $O(n)$ となる．本論文で提案する制約緩和手法は，この 2 つ目の条件を緩和し多くの場合でデータベースの一部分の走査でプライベート問合せを実現する．提案手法は，一次元データベースに対する一致問合せだけでなく，範囲問合せや二次元データベースに対する一致問合せにも利用できる．

キーワード：データベース，セキュリティ，プライバシ，問合せ処理

Frequency-based Constraint Relaxation for Private Query

Junpei Kawamoto[1,a)]　Patricia L. Gillett[2]　Jun Sakuma[1]

Received: December 20, 2012, Accepted: April 6, 2013

Abstract: We introduce a frequency-based constraint relaxation methodology for *private queries*. Private queries undergo processing in order to allow people to obtain data from a database without exposing which data the user is interested in (*search intention*). Most existing protocols for private querying achieve this by the following two constraints: i) queries and query results are encoded so that the database server cannot actually decode queries but can handle query processes; ii) the server is forced to check all data in the server when computing query results. Because of these constraints, even database servers cannot distinguish which data are selected from the database. However, this second constraint compels servers to spend $O(n)$ computational cost for each query processed, where n is the number of data entries on the server. Our relaxation methodology relaxes this constraint and allows private querying while only examining a portion of the data in most cases. Our methodology is also flexible and applies not only to exact match queries in one dimensional data but also to range queries in one dimensional data and exact match queries in two dimensional data.

Keywords: database, security, privacy, query processing

1. はじめに

プライベート問合せ [1], [2], [3], [4], [5], [6] は利用者が何を問い合わせているのか（**検索意図**）をサーバに隠したまま目的のデータを取得する仕組みであり，サービスとして提供されているデータベース（DaaS; Database as a Service）を安全に利用するために用いられる．利用者の検索意図は商業的に重要な情報である一方，利用者のプライバシ情報

[1] 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305–0006, Japan
[2] モントリオール理工科大学
École Polytechnique du Montréal, Montreal, Quebec, Canada
[a)] junpei@mdl.cs.tsukuba.ac.jp

本論文の内容は 2012 年 11 月の WebDB2012 にて発表され，同シンポジウムプログラム委員会により情報処理学会論文誌データベースへの掲載が推薦された論文である．

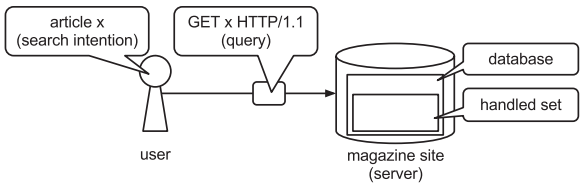


図 1　検索意図，問合せおよび走査集合
Fig. 1　Search intention, query, and handled set.

を含んでいる可能性があるからである．たとえば，データベースサービスを利用したアプリケーションとしてニュースサイトを考える．攻撃者が，利用者がどんな記事を問合せ閲覧していたのかを取得することで，その利用者の趣味や政治観，宗教観などを推測することが可能となる．そして，これらは利用者によっては秘密にしておきたい種類の情報である場合が少なくない．また，より深刻な場合として，医薬品データベースの例が考えられる．利用者が，病院から処方された薬に関する情報を得るために医薬品データベースを利用することを考える．このとき，利用者の検索意図，すなわちどんな医薬品名を問い合わせたのかという情報から，その利用者の病名を推測することが可能である．こうしたリスクを回避するためにプライベート問合せは用いられる．

本論文では，検索意図，問合せそして走査集合という 3 つの概念を用いてプライベート問合せの性質を議論する．**検索意図**（*search intention*）とは，利用者が実際に何をサーバに問い合わせたのかそのものをいう．**問合せ**（*query*）とは，検索意図に合致する情報をサーバから取得するために利用者が送信するデータのことをいう．この問合せは，一般的にサーバと利用者が用いるプロトコルで定義された形式で書かれている．**走査集合**（*handled set*）とは，サーバが受け取った問合せを処理し問合せ結果を作成するために走査しなければならない最小のアイテム集合をいう．最小のアイテム集合であるので，サーバがとりうる最善の方法で問合せを処理した場合に走査するアイテム集合が走査集合である．**図 1** は，あるニュースサイトにおける 3 つの概念を図示したものである．あるニュースサイト（magazine site）の利用者（user）が，記事 x を閲覧したいと考えている．この利用者の検索意図は x になる．今，このニュースサイトを利用するために簡易的な HTTP プロトコルを用いるとすると，この利用者は記事 x を取得するために問合せ “GET x HTTP/1.1” をサーバに送る．このニュースサイトのサーバは問合せを受け取り，サーバが持っているデータベースの一部を走査し記事 x を発見する．そして，利用者へ問合せ結果として記事 x を送信する．この問合せ処理においてサーバが走査する必要のあるデータ集合は，データベースの設計と使用プロトコルに依存する．たとえば，データベースが x の探索用に効率的なハッシュ索引を用意可能な場合，x のみを走査し問合せ結果を作成できる．しかし，いっさいの索引が利用できずアイテム間に順序も定義されない場合はすべてのアイテムを走査する必要がある．このニュースサイトのサーバが効率的な索引を用意可能とすると，検索意図 x に対する走査集合は $\{x\}$ となる．

多くのプロトコル [3], [7] は，次の事柄によってプライベート問合せを実現している．i) サーバが問合せ処理を実行することはできるが解読することができないような符号化を問合せと問合せ結果に施す．ii) 走査集合がつねにデータベース全体となることを強制する．すなわち，サーバはどのような方法を用いて問合せ処理を効率化してもデータベース全体の走査が必要となる．1 つ目の条件を実現するためには暗号化などの技術が用いられる．そして，2 つ目の条件すなわちサーバに課された制約により，既存のプロトコルはデータベースサーバに対してもデータベース全体のうちどのデータが実際に求められていたのかを隠している．一方，この制約によって，データベース内の総アイテム数を n として，サーバは $O(n)$ の計算コストをあらゆる問合せに対して負うことになる．そのため，既存のプロトコルを巨大なデータベースへ適用することは難しい．

一方で，我々は利用者の検索意図をデータベース全体の中から秘匿しなくてもよい場合があると考えている．たとえば，ニュースサイトにおいては利用者がすべての記事の中からどの記事を閲覧したのかまでを隠さなくても，一部の記事の中から閲覧記事を隠すだけで，その利用者の政治観や宗教観を秘匿可能な場合が考えられる．このような場合では，先ほどの制約を緩和しデータベースの部分集合から検索意図を隠すことができる．bbPIR [5] は，この制約緩和をナイーブに実現したプロトコルであり，検索意図 x をデータベース全体から隠すのではなくアイテム番号が x に近い $k-1$ 個のアイテムの中から隠す．すなわち，bbPIR における走査集合は x を含む k 個のアイテムとなる．しかし，このナイーブな方法は安全ではない．たとえば，ニュースサイトの例では，利用者の検索意図である記事 x が人気記事で，その他の $k-1$ 記事があまり問い合わせられていないことを攻撃者が知っていたとする．このとき，利用者の検索意図が x であることは高確率で推測されうる．

本論文では，検索意図の頻度をもとにした安全な制約緩和手法を提案する．提案手法の基になるアイデアは，検索意図が人気のアイテムである場合，検索意図は高い確率で推測されうるため，多くのアイテムの中から隠す必要がある．一方で，検索意図があまり人気のないアイテムの場合，それほど大きなアイテム集合の中から隠さなくても検索意図が推定される確率は大きくないというものである．制約緩和を行わないプロトコルを**完全なプロトコル**と呼ぶと，提案手法は上記のアイデアを基に多くの場合で走査集合を完全なプロトコルに比べて削減する．しかし，同時に安全性についても確保しており，提案手法では制約緩和による

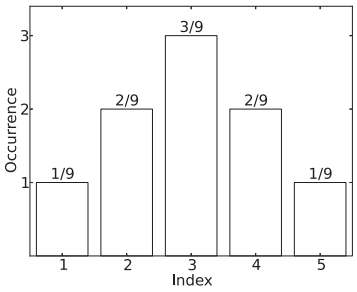


図 2　検索意図の頻度に関する例
Fig. 2　A small example of frequency.

リスクが完全なプロトコルにおける最大のリスクを超えないことを保証する．一般的に，完全なプロトコルは安全であると考えられているため，その最大リスクを超えない本提案手法も安全であるといえる．本論文では，一次元データベースにおける一致検索に対する制約緩和手法を導入した後，範囲問合せに対する制約緩和手法と二次元データベースに対する一致検索に拡張する．それゆえ，提案手法は様々な種類のデータベースに適用することが可能である．

我々が提案する，検索意図の頻度分布を用いた制約緩和手法の概要は次のとおりである．例として 5 個のアイテムからなり，利用者の問合せ回数および検索意図の頻度が**図 2** のようなデータベースを考える．完全なプロトコルでは，検索意図が 1 のときでも 3 のときでも走査集合は $[1,5]$ である．しかし，図 2 に示す検索意図の頻度を攻撃者が知っている場合，走査集合が $[1,5]$ であっても，検索意図が 1 および 3 と推測できる確率，すなわち問合せリスクはそれぞれ 1/9 と 3/9 と考えることができる．本論文で提案する緩和手法は，いずれの検索意図に対してもリスクが 3/9 以下という条件の下，走査集合を最小化する．この例では，検索意図が 1 の場合は $[1,2]$ が緩和された走査集合になる．このときのリスクは $1/(1+2) = 1/3$ である．また，検索意図が 3 の場合は緩和されず走査集合は，$[1,5]$ のままである．

本論文の構成は次のとおりである．まず，2 章で本論文で用いる基本的な概念について定義する．その後，3 章で制約緩和アルゴリズムについて述べ，4 章で評価実験を行う．さらに，5 章では，検索意図と制約緩和された走査集合の対応関係を知る強い攻撃者に対する拡張について述べる．最後に，6 章で関連研究について述べた後，7 章でまとめを述べる．

2. 基本事項

本論文では，データベース D は n 個のアイテム $\{t_1, t_2, \cdots, t_n\}$ からなり，利用者の目的は x 番目のアイテム t_x を求めることであるとする[*1]．このとき，利用者の検索意図は x である．たとえば，1 章で用いたニュースサイトの場合，このサイトが n 本の記事を掲載しており利用者に記事番号と見出しの組からなる目次を提供しているとする．読者，すなわちこのサイトの利用者は，目次から興味ある見出しを選びその記事番号 x をサーバに問い合わせる．

cPIR [7] や IPP 法 [3] などといった既存のプライベート問合せを実現するプロトコルでは，平文の検索意図 x はサーバに直接送信されず符号化された問合せが送られる．サーバは受け取った問合せを基にデータベース D の中のいくつかのアイテムに対して演算を行い問合せ結果を計算する．サーバが検索意図 x に対する問合せ結果の計算に用いなければならない最小のアイテム集合，すなわち走査集合を $H(x)$ と書くと，$H(x) \subseteq D$ を満たす．完全なプロトコルにおけるこの走査集合はつねにデータベース全体であり $\forall t_x \in D : H_{\mathrm{comp.}}(x) = D$ となる．

次に，検索意図 x の頻度を $\mathrm{Freq}(x)$ と書く．本論文では，この頻度が正規化され $\sum_{t_x \in D} \mathrm{Freq}(x) = 1$ を満たすとする．この検索意図の頻度は，サーバ，利用者および攻撃者が知りうる公開情報であるとする．この頻度を用いて，プライベート問合せにおける**問合せリスク**を定義する．サーバは，検索意図が x である問合せに対して走査集合 $H(x)$ に含まれるアイテムを走査し問合せ結果を作成する．よって，我々はこの検索意図 x に対する問合せリスクを $H(x)$ から x を推測できる確率として定義する．

定義 2.1　検索意図 x に対する走査集合 $H(x)$ の問合せリスク $\mathrm{Risk}(x|H(x))$ は，

$$\mathrm{Risk}\,(x|H(x)) = \frac{\mathrm{Freq}(x)}{\sum_{t_y \in H(x)} \mathrm{Freq}(y)}.$$

本論文では，公開されている検索意図の頻度を用いた攻撃を行う攻撃者を考える．また，攻撃者は問合せから直接利用者の検索意図 x を知ることはできないが，サーバが問合せ処理に用いた走査集合 $H(x)$ は知ることができるとする．そして，攻撃者の目的は利用者の検索意図 x を走査集合 $H(x)$ と検索意図の頻度 $\mathrm{Freq}(x)$ からの推測とする．我々は，任意の検索意図 x に対して，その問合せリスクが完全なプロトコルにおける最大の問合せリスク $\max_{t_y \in D} \mathrm{Risk}\,(y|H_{comp.}(y)) = \max_{t_y \in D} \mathrm{Risk}\,(y|D)$ を下回っているとき，走査集合 H は安全であるという．

定義 2.2　$\forall t_x \in D$ に対して，

$$\mathrm{Risk}\,(x|H(x)) \leq \max_{t_y \in D} \mathrm{Risk}\,(y|D) \qquad (1)$$

が満たされるとき H は安全であるという．

この定義は，我々の制約緩和手法がある検索意図に対する問合せリスクの増加を許すことを意味する．しかし，その問合せリスクは完全なプロトコルにおける最大のリスクを超えない．一般に制約緩和を行わない完全なプロトコルは安全であると認められている．そのため，その最大リス

[*1] 範囲問合せでは，利用者の目的は，ある範囲 $[x, y]$ のアイテム $\{t_x, t_{x+1}, \cdots, t_y\}$ の取得とする．範囲問合せに関しては 3.1 節で改めて議論する．

クを超えなければ制約緩和を行いリスクが部分的に増大しても安全であると考えることができる．

最後に，問合せの処理コストを定義する．本論文では，検索意図 x を表す問合せの処理コストを $C(x)$ と書き，走査集合の大きさで評価する．すなわち，$C(x) = |H(x)|$ である．また，検索意図全体に対する問合せ処理コストの期待値を考え C と書くことにする．したがって，$C = \sum_{t_x \in D} \mathrm{Freq}(x) C(x)$ となる．完全なプロトコルでは，サーバは検索意図によらずつねにすべてのアイテムを走査する．つまり $\forall t_x \in D$: $H_{\mathrm{comp.}}(x) = D$ であるため $C_{\mathrm{comp.}}(x) = n$ となり $C_{\mathrm{comp.}} = n$ となる．

3. 制約緩和

本論文で我々が提案する制約緩和手法によって，サーバは問合せ処理のためにデータベース全体を走査する必要はなくなり，その一部のみを走査する．本章では，まず一次元データベースにおける一致検索に対する制約緩和手法を導入する．提案手法を用いると，利用者の検索意図が x のとき，制約緩和された走査集合は連続するアイテムの集合 $H_1(x) = \{t_i, t_{i+1}, \cdots, t_j\}$ となる．ここで，$i \le x \le j$ であり，$t_x \in H_1(x)$ を満たす．$H_1(x) \subseteq H_{comp.}(x)$ であるから，サーバは多くの場合計算コストを削減できる．この制約緩和により問合せリスクは増加する．しかし，前章までで述べたように，提案手法はこの増加した問合せリスクが完全なプロトコルにおける最大のリスクを超えないことを保証する．すなわち $H_1(x)$ は式 (1) を満足し，定義 2.2 の安全性を満たす．この条件を満足する走査集合 $H_1(x)$ を求める問題は次のようなる．

問題 3.1　与えられた検索意図 x に対して，(i) $t_x \in H_1(x)$，(ii) $\mathrm{Risk}(x|H_1(x)) \le \max_{t_y \in D} \mathrm{Risk}(y|D)$，(iii) $|H_1(x)|$ を最小化する，(iv) 条件 (iii) の下で $\sum_{t_y \in H_1(x)} \mathrm{Freq}(y)$ を最大化するという条件を満足し連続するアイテムからなる集合 $H_1(x) = \{t_i, t_{i+1}, \cdots, t_j\} \subseteq D$ を求める．

条件 (ii) によって見つかった走査集合が攻撃者に対して安全であると保証しており，条件 (iii) によって問合せコストを最小化している．さらに，上記の 2 条件を満足する候補が複数存在する場合は，条件 (iv) によって問合せリスクを最小にする走査集合を選んでいる．

この問題 3.1 は，**図 3** に示すアルゴリズム Find H_1 によって解くことができ，その計算量は $O(n)$ である．さらに，得られた $H_1(x)$ は同じに検索意図 x 対して再利用することができる．また，アルゴリズム Find H_1 によって得られた走査集合 $H_1(x)$ に対するサーバ計算コストの期待値は，$C_1 = \sum_{t_x \in D} \mathrm{Freq}(x)|H_1(x)| \le \sum_{t_x \in D} \mathrm{Freq}(x)|D| = n$ であり，完全なプロトコルにおける計算コストより小さい．

この制約緩和を用いた問合せは次のとおりである．データベース利用者の検索意図が x であるとする．この利用者はまずアルゴリズム Find H_1 を用いて走査集合 $H_1(x)$ を

Require: D はデータベース，Freq は検索意図の頻度.
  $bestsofar \leftarrow |D|+1$, $best \leftarrow [0, |D|-1]$
  $bestsum \leftarrow \sum_{t_y \in D} \mathrm{Freq}(y)$ $i \leftarrow x$, $j \leftarrow x$
  $vsum \leftarrow \mathrm{Freq}(x)$, $\mu \leftarrow \mathrm{Freq}(x)/\max_{t_y \in D} \mathrm{Risk}(y)$
  while $vsum < \mu$ **and** $i \ge 1$ **do**
    $i \leftarrow i-1$, $vsum \leftarrow vsum + \mathrm{Freq}(i)$
  end while
  if $vsum \ge \mu$ **then**
    $bestsofar \leftarrow j-i+1$, $best \leftarrow [i,j]$, $bestsum \leftarrow vsum$
  end if
  while $i \le x$ **do**
    while $vsum \ge \mu$ **do**
      $j \leftarrow j+1$, $vsum \leftarrow vsum + \mathrm{Freq}(j)$
    end while
    if $vsum \ge \mu$ **then**
      if $j-i+1 < bestsofar$ **then**
        $bestsofar \leftarrow j-i+1$, $best \leftarrow (i,j)$
        $bestsum \leftarrow vsum$
      else if $j-i+1 = bestsofar$ **and** $vsum > bestsum$ **then**
        $bestsofar \leftarrow j-i+1$, $best \leftarrow (i,j)$
        $bestsum \leftarrow vsum$
      end if
    end if
    $i \leftarrow i+1$, $vsum \leftarrow vsum - \mathrm{Freq}(i-1)$
  end while
  return $best$

図 3　$H_1(x)$ の計算アルゴリズム Find H_1
Fig. 3　Algorithm for computing $H_1(x)$; Find H_1.

求める．次に，得られた $H_1(x)$ に含まれるアイテムのみからなる部分データベースを仮定し，その上で検索意図 x を表す問合せ q を作成する．そして，利用者は $H_1(x)$ と q をサーバに送信する．サーバは，受け取った走査集合に含まれるアイテムからなる部分データベース上で問合せ q を評価し結果を利用者に返却する．なお，問合せ q は本制約緩和手法とともに用いるプロトコルによって定まっているものとする．たとえば，cPIR に本手法を組み合わせて用いる場合は，cPIR における問合せを用いる．

3.1　一次元データベースに対する範囲問合せへの拡張

次に，本節では先ほど提案したアルゴリズムを一次元のデータベースに対する範囲問合せへ拡張する．一致問合せと範囲問合せの差異は，検索意図が単一の値 x ではなく区間 $[x_L, x_R]$（ただし，$1 \le x_L$ かつ $x_R \le n$）となることである．以降では，区間 $[1, n]$ の部分区間の全体を $\mathcal{I}$ と書き，この範囲検索における検索意図を $X \in \mathcal{I}$ で表すことにする．そして，検索意図の頻度および問合せリスクを，範囲検索における検索意図 X 上に拡張する．

$$\mathrm{Freq}(X) = \frac{\sum_{x \in X} \mathrm{Freq}(x)}{\sum_{I \in \mathcal{I}} \sum_{y \in I} \mathrm{Freq}(y)}$$

$$\mathrm{Risk}(X|H(X)) = \frac{\mathrm{Freq}(X)}{\sum_{t_y \in H(X)} \mathrm{Freq}(y)}$$

また，問合せのコストと期待値についても同様に拡張し $C(X) = |H(X)|$，$C = \sum_{X \in \mathcal{I}} \mathrm{Freq}(X) C(X)$ とする．

Require: D はデータベース, Freq は検索意図の頻度, $X = [x_L, x_R]$.
  $bestsofar \leftarrow |D| + 1$, $best \leftarrow [0, |D| - 1]$
  $bestsum \leftarrow \sum_{t_y \in D} \mathrm{Freq}(y)$, $i \leftarrow x_L$, $j \leftarrow x_R$
  $vsum \leftarrow \mathrm{Freq}(X)$, $\mu \leftarrow \mathrm{Freq}(X)/\max_{t_y \in D} \mathrm{Risk}(y)$
  while $vsum < \mu$ **and** $i \geq 1$ **do**
    $i \leftarrow i - 1$, $vsum \leftarrow vsum + \mathrm{Freq}(i)$
  end while
  if $vsum \geq \mu$ **then**
    $bestsofar \leftarrow j - i + 1$, $best \leftarrow [i, j]$, $bestsum \leftarrow vsum$
  end if
  while $i \leq x_L$ **do**
    while $vsum \geq \mu$ **do**
      $j \leftarrow j + 1$, $vsum \leftarrow vsum + \mathrm{Freq}(j)$
    end while
    if $vsum \geq \mu$ **then**
      if $j - i + 1 < bestsofar$ **then**
        $bestsofar \leftarrow j - i + 1$, $best \leftarrow (i, j)$
        $bestsum \leftarrow vsum$
      else if $j - i + 1 = bestsofar$ **and** $vsum > bestsum$ **then**
        $bestsofar \leftarrow j - i + 1$, $best \leftarrow (i, j)$
        $bestsum \leftarrow vsum$
      end if
    end if
    $i \leftarrow i + 1$, $vsum \leftarrow vsum - \mathrm{Freq}(i - 1)$
  end while
  return *best*

図 4　$H_{1\mathrm{R}}(X)$ の計算アルゴリズム Find $H_{1\mathrm{R}}$
Fig. 4　Algorithm for computing $H_{1\mathrm{R}}(X)$; Find $H_{1\mathrm{R}}$.

範囲検索における完全なプロトコルの走査集合は，一致検索の場合と同様にデータベースすべてであり $\forall X \in \mathcal{I} : H_{comp.}(X) = D$ となる．一方，提案手法における走査集合 $H_{1\mathrm{R}}(X)$ は，連続するアイテム集合からなるデータベースの一部分となる．この走者集合 $H_{1\mathrm{R}}(X)$ を発見する問題は，先の問題 3.1 を拡張して次のようになる．

問題 3.2　与えられた検索意図 $X = [x_L, x_R]$ に対して，(i) $\{t_{x_L}, t_{x_L+1}, \cdots, t_{x_R}\} \subseteq H_{1\mathrm{R}}(X)$，(ii) $\mathrm{Risk}(X|H_{1\mathrm{R}}(X)) \leq \max_{t_y \in D} \mathrm{Risk}(y|D)$，(iii) $|H_{1\mathrm{R}}(X)|$ を最小化する，(iv) 条件 (iii) の下で $\sum_{t_y \in H_{1\mathrm{R}}(X)} \mathrm{Freq}(y)$ を最大化するという条件を満足する $H_{1\mathrm{R}}(X) = \{t_i, t_{i+1}, \cdots, t_j\} \subseteq D$ を求める．

この問題は，**図 4** に示すアルゴリズム Find $H_{1\mathrm{R}}$ によって求めることができ，その計算量は，アルゴリズム Find H_1 と同様に $O(n)$ である．

3.2　二次元データベースに対する一致問合せへの拡張

最後に，二次元データベースに対する一致問合せへの拡張を行う．本論文では，二次元データベースを $l \times m = n$ の l 行 m 列行列として考える．利用者の目的は，この行列のうち e 行 g 列目のアイテムを取得することとする．すなわち，利用者の検索意図は $x = (e, g)$ と書ける．完全なプロトコルでは，任意の検索意図 x に対してその走査集合 $H_{comp.}(x)$ は，$D[[1, l]; [1, m]]$ である[*2]．したがって，検索意図 x によらず $C_{comp.}(x) = |H_{comp.}(x)| = n$ であり，サーバはつねにすべてのアイテムを走査する必要がある．

[*2] 行列 D の i_a 行目から i_b 行目および j_a 列目から j_b 列目からなる部分行列を $D[[i_a, i_b]; [j_a, j_b]]$ と書く．

Require: $D' \in \mathbb{R}^{l \times m}$ は問合せ頻度行列 ($D'_{i,j} = \mathrm{Freq}(i \times m + j)$).
  $\mu \leftarrow D'_{e,g}/\max_{t_y \in D} \mathrm{Risk}(y)$, $C_{e,g} \leftarrow D'_{e,g}$
  $vsum \leftarrow \sum_{i,j} D'_{i,j}$, $size \leftarrow l \times m$, $best \leftarrow (0, 0, l, m)$
  $h \leftarrow 1$, $w \leftarrow 1$, $maxw \leftarrow m$, $C \leftarrow \mathrm{NegOnes}(h, w)$
  $rangeC \leftarrow (e, e, g, g)$
  while $h \leq \min(l, size)$ **do**
    $M \leftarrow \mathrm{NegOnes}(h, w)$
    $M_{i,g} \leftarrow C_{i,g} \quad \forall rangeC[0] \leq i \leq rangeC[1]$
    $rangeM \leftarrow (rangeC[0], rangeC[1], g, g)$
    while $w \leq maxw$ **do**
      $(i^{new}, j^{new}) \leftarrow \mathrm{argmax}(M), v^{new} \leftarrow M_{i^{new}, j^{new}}$
      if $(v^{new} > vsum)$ **or** $(v^{new} \geq \mu$ **and** $h \times w < size)$ **then**
        $vsum \leftarrow v^{new}$, $size \leftarrow h \times w$
        $best \leftarrow (i^{new}, j^{new}, h, w)$, $maxw \leftarrow \min(m, \lfloor \frac{size}{h} \rfloor)$
      end if
      $w \leftarrow w + 1$
      if $w \leq maxw$ **then**
        $(C, rangeC, M, rangeM)$
          $\leftarrow \mathrm{updateM}(h, w, C, rangeC, M, rangeM, D')$
      end if
    end while
    $h \leftarrow h + 1, w \leftarrow 1$
    if $h \leq \min(l, size)$ **then**
      $maxw \leftarrow \min(m, \lfloor \frac{size}{h} \rfloor)$
      $rangeC[2] \leftarrow \max(rangeC[2], g - maxw + 1)$
      $rangeC[3] \leftarrow \min(rangeC[3], g + maxw - 1)$
      $(C, rangeC) \leftarrow \mathrm{updateC}(h, C, rangeC, D')$
    end if
  end while
  return *best*

図 5　$H_2(x)$ の計算アルゴリズム Find H_2
Fig. 5　Algorithm for computing $H_2(x)$; Find H_2.

一次元データベースの場合と同様に，提案手法ではこの制約を緩和し安全性の条件である式 (1) を満足する走査集合 H_2 としてデータベースを表す行列の部分行列を求める．ある検索意図 $x = (e, g)$ に対して，走査集合 $H_2(x)$ を発見する問題は，次のように定義できる．

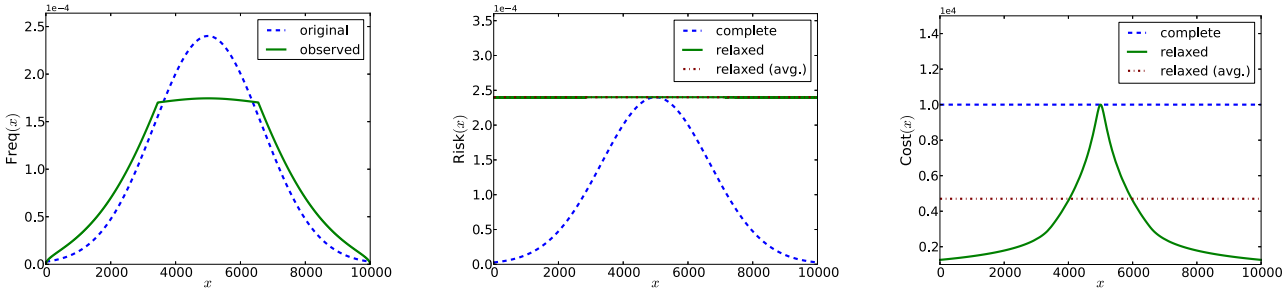
問題 3.3　与えられた検索意図 $x = (e, g)$ に対して，(i) x を含む，すなわち $i \leq e < i + h$ かつ $j \leq g < j + w$，(ii) $\mathrm{Risk}(x|H_2(x)) \leq \max_{t_y \in D} \mathrm{Risk}(y|D)$，(iii) $h \times w$ を最小化する，(iv) 条件 (iii) の下で $\sum_{t_y \in H_2(x)} \mathrm{Freq}(y)$ を最大化するという条件を満足する，データベース行列 D の部分行列 $H_2(x) = D[[i, i + h - 1]; [j, j + w - 1]]$ を発見する．

この問題は**図 5** に示すアルゴリズム Find H_2 によって解くことができる．アルゴリズムの計算量は $O(n^2)$ となるが，詳細は付録に記す．

3.3　近似問合せ頻度の利用

2 章で述べたように，本論文では検索意図の頻度 $\mathrm{Freq}(x)$ は公開情報と仮定していた．本節では，検索意図の頻度順と一部の頻度しか公開されていない場合について議論する．

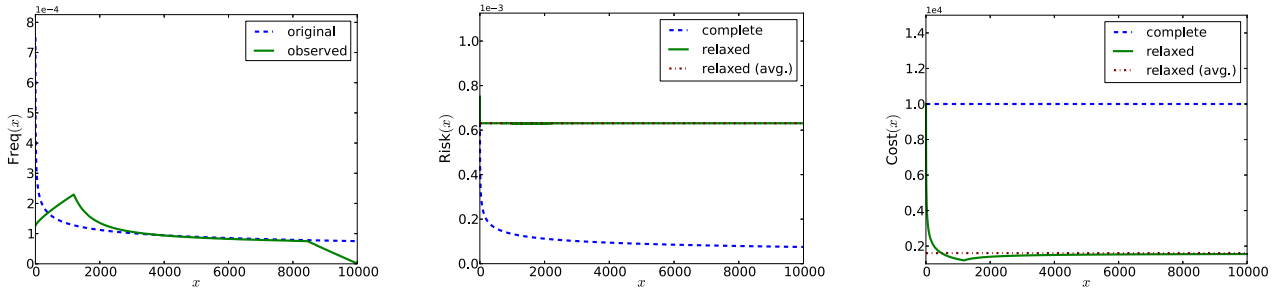
まず，本来の頻度分布と頻度順は同じで頻度差の小さい



(a) 検索意図の頻度　(b) 問合せリスク　(c) サーバ計算コスト

図 6　ガウス分布に従う問合せ頻度に対する実験結果（$n = 10000$，$\mu = n/2$，$\sigma = n/6$）

Fig. 6　Experimental results for queries distributed according to a Gaussian distribution ($n = 10000, \mu = n/2, \sigma = n/6$).



(a) 検索意図の頻度　(b) 問合せリスク　(c) サーバ計算コスト

図 7　Zipf 分布に従う問合せ頻度に対する実験結果（$n = 10000$，$s = 0.25$）

Fig. 7　Experimental results for queries distributed according to a Zipf distribution ($n = 10000, s = 0.25$).

近似的な頻度分布 $\widetilde{\mathrm{Freq}}(x)$ を考える．すなわち，$x, y \in D$ に対して $\mathrm{Freq}(x) \leq \mathrm{Freq}(y)$ ならば $\widetilde{\mathrm{Freq}}(x) \leq \widetilde{\mathrm{Freq}}(y)$ かつ，$|\mathrm{Freq}(x) - \mathrm{Freq}(y)| \geq |\widetilde{\mathrm{Freq}}(x) - \widetilde{\mathrm{Freq}}(y)|$ とする．このような近似頻度分布は，データベース内のアイテムが頻度順に並んでいる場合や本来の頻度分布が正規分布と見なせる場合，容易に作成ができる．直感的には，高頻度のアイテムに対する頻度を少なめに稀なアイテムに対する頻度を多めに見積もった分布といえる．なお，一様分布はこの近似頻度分布の特殊な場合である．

この近似頻度分布を用いて計算した完全なプロトコルにおける最大リスクの近似値 $\max_{y \in D} \widetilde{\mathrm{Risk}}(y|D)$ は，$x_f = \arg\max_{t_x \in D} \mathrm{Freq}(x)$ とすると $\max_{t_y \in D} \widetilde{\mathrm{Risk}}(y|D) = \widetilde{\mathrm{Freq}}(x_f) \leq \mathrm{Freq}(x_f) = \max_{t_y \in D} \mathrm{Risk}(y|D)$ を満たす．したがって，近似頻度分布を用いた場合でもあっても実際の頻度分布による完全なプロトコルの最大リスクを超えることはない．そのため，実際の頻度分布の一部しか公開されていない場合，本来の分布よりも頻度差の小さい近似分布を仮定すれば提案の制約緩和を行える．また，検索意図の頻度が時間とともに変化するような場合であっても，頻度順位に変化がなければ同じ近似頻度分布を用いることができる．

4. 評価実験

提案手法の効果を 2 つの側面から評価する．実験に用いたプログラムはすべて Python 2.7 で実装し，AMD FX™-8120 Processor, 32GB RAM で OS が Ubuntu 12.04 LTS である計算機上で実行した．

1 つ目は，アイテム数 $n = 10000$ の一次元データベース D に対して $\forall t_x \in D$ を問い合わせた場合を考え，提案手法適用後の見かけの検索意図頻度，問合せリスクおよびサーバ計算コストを計測した．この実験では，元々の検索意図頻度としてガウス分布と Zipf 分布の 2 種類を用意した．ガウス分布のパラメータは，$\mu = n/2$ および $\sigma = n/6$ とし，

$$\mathrm{Zipf}(x; n, s) = \frac{1/x^s}{\sum_{k=1}^{n} 1/k^s}$$

で定義される Zipf 分布のパラメータとして $s = 0.25$ を選んだ．Zipf 分布の場合，パラメータによって出力に大きな変化は見られなかった．そこで，グラフの変化が判別しやすいパラメータを選んだ．また，この実験ではガウス分布と Zipf 分布という 2 つの分布を選んだが，一般的にどのような検索意図分布であっても，アイテム番号の並べ替えにより検索意図頻度をこれらの分布に近づけられる．

図 6 (a) と**図 7** (a) は，それぞれガウス分布および Zipf 分布を元々の検索意図頻度とした場合（complete）に，提案手法を用いた問合せから求めた検索意図の頻度，すなわちサーバが観測した見かけの検索意図頻度（relaxed）を示したものである．なお，それぞれの頻度は正規化してある．見かけの検索意図頻度は次の式で求めた．

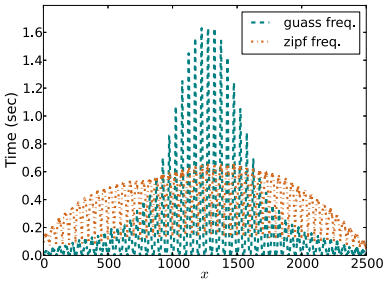
$$\mathrm{Freq}_{\mathrm{observed}}(x) = \frac{\sum_{t_y \in D} \sum_{t_x \in H(y)} \mathrm{Freq}(y)}{\sum_{t_y \in D} \mathrm{Freq}(y) \times |H(y)|}$$

図 6 (a) および図 7 (a) の横軸は検索意図 x を表しており，縦軸は検索意図の頻度を表している．ともに，提案手法により頻度のピークが別の形に変わっていることが分かる．

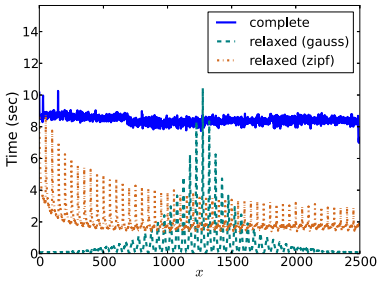
図 6 (b) と図 7 (b) は，横軸に検索意図 x をとり，縦軸にその検索意図に対する問合せリスクを示したものである．完全なプロトコルにおける問合せリスク（complete）と提案手法を用いて制約を緩和した場合における問合せリスク（relaxed）に加え，$\sum_{t_x \in D} \mathrm{Freq}(x) \times \mathrm{Risk}(x|H(x))$ で計算できる平均問合せリスク（relaxed (avg.)）もあわせて記してある．この平均問合せリスクと制約緩和後の問合せリスクとの間に大きな差はなかった．我々の提案手法は制約を緩和しているため，一般的に問合せリスクは完全なプロトコルに対して増加する．しかし，提案の制約緩和手法は，増加する問合せリスクが完全なプロトコルにおける最大のリスクを超えないことを保証している．図 6 (b) と図 7 (b) は実際にそのことを示している．具体的に見ていくと，図 6 (b) において，完全なプロトコルにおける最も高い問合せリスクは $x = 5000$ のときであり Freq(5000) となる．そして，制約緩和された場合における問合せリスクを見ていくと，任意の x に対してほぼ Freq(5000) に等しく，また超えることはない．図 7 (b) の場合は，最も問合せリスクが高いのは $x = 0$ のときであり Freq(0) となる．同様に，制約緩和された場合の問合せリスクは，任意の x に対してほぼ Freq(0) に等しく超えることはない．

図 6 (c) と図 7 (c) は横軸に検索意図 x をとり，縦軸にその検索意図に対するサーバの問合せ計算コストを記したものである．計算コストは完全なプロトコル（complete），提案手法によって制約を緩和した場合（relaxed），そして制約緩和した場合の期待値（relaxed (adv.)）の 3 つを記している．完全なプロトコルでは，サーバはつねにデータベース内すべてのアイテムを走査するため，計算コストは x に依存しない．一方で，制約緩和を行った場合，最も高い頻度の検索意図以外に対してはサーバにおける計算コストが削減されていることが分かる．最も高い頻度の検索意図に対しては，そのサーバ計算コストは完全なモデルと同じであるが，これは意図したとおりの動きである．また，問合せ処理コストの期待値より，提案手法が問合せ頻度の少ないアイテムに対する問合せ処理コストの削減だけではなく，全体として問合せ処理コストを削減していることが分かる．

2 つ目の評価実験では，実際にプライベート問合せを実現するサーバ・クライアントを用いて，問合せに要する時間を計測した．ここでは，プライベート問合せを実現する手法として，cPIR プロトコルを選び，アイテム数 $n = 2500$ のデータベースを構築した．また，1 つ目の実験と同じように，ガウス分布に従う検索意図頻度と Zipf 分布に従う検索意図頻度の 2 種類を用意した．ガウス分布におけるパラメータは，先ほどと同様に，$\mu = n/2$ かつ $\sigma = n/6$ とし，Zipf 分布におけるパラメータは $s = 0.25$ とした．



(a) 制約緩和された走査集合の計算時間



(b) サーバにおける問合せ処理時間

図 8　二次元属性に対する問合せの処理時間
Fig. 8　Query processing times in 2D data.

図 8 (a) は，各検索意図 x に対して走査集合 $H(x)$ を求めるのに要した時間を記している．グラフはガウス分布を用いた場合（gauss）と Zipf 分布を用いた場合（zipf）の 2 種類を記してある．ここでは，cPIR プロトコルを用いているので，検索意図は $l \times m$ 行列に対して $x = (e, g)$ 番目のアイテムを求めることを意味する．グラフでは横軸に x をとっているが，これは $x = e \times m + g$ によって計算したものである．図 8 (a) によれば，走査集合を求めるのに要した時間は大きく波打っている．現在のところ，なぜこのような現象が起きるのか分かってはいないが，走査集合を求めるのに要する時間はサーバにおける計算時間に比べ小さいことが分かる．図 8 (b) は，サーバにおける問合せ処理時間を比較したものである．図には，完全なプロトコルとして cPIR を用いた場合の処理時間（complete）と，検索意図の頻度がガウス分布および Zipf 分布に従うと仮定して提案手法を用いた場合（それぞれ gauss，zipf）の 3 つを記している．多くの場合，サーバにおける計算時間は制約緩和を行った方が完全なプロトコルが要する時間よりも短くて済む．一方，最も高い頻度で問い合わされている検索意図に対しては，完全なプロトコルの方が短い時間であった．これは，ガウス分布における x が $n/2$ 近辺と Zipf 分布における x が 0 近辺のことである．頻度が高い場合，提案手法でもほとんど制約を緩和せずデータベース全体を走

査することになる．したがって，送られてきた走査集合を解釈するのに要する時間の分だけ，提案手法の方が時間を要することになり，その差が現れているといえる．

5. 走査集合のランダム選択による拡張

前章までで，検索意図に対してサーバが走査すべきアイテム数を削減する制約緩和手法について議論した．提案の制約緩和手法は，任意の検索意図に対し定義 2.1 の問合せリスクが制約緩和を行わない完全なプロトコルにおける最大の問合せリスクを超えないことを保障している．一方で，検索意図と走査集合の対応関係を知っている強い攻撃者に対しては，制約緩和された走査集合 $H(x)$ から検索意図 x が高い確率で推測可能な場合がある．

例 5.1　図 2 の頻度分布に対して一致検索用の制約緩和を行うと，$H_1(1)=\{1,2\}$，$H_1(2)=\{1,2,3\}$，$H_1(3)=\{1,2,3,4,5\}$，$H_1(4)=\{3,4,5\}$，$H_1(5)=\{4,5\}$ となる．

この走査集合を用いた場合の問合せリスクは完全なプロトコルにおける最大リスクを超えないことが保障されているが，検索意図 x と走査集合 $H_1(x)$ が 1 対 1 に対応付いている．そのため，この対応関係を知る攻撃者は走査集合から検索意図を推測することができる．本章では，検索に対し用いる走査集合を確率的に選択することで，走査集合から検索意図の推測リスク完全なプロトコルにおける最大の問合せリスク以下となるランダム選択を導入する．

初めに，検索意図 x に対して包含走査集合族を定義する．

定義 5.1　検索意図 x の包含走査集合族 $\mathrm{Super}(x)$ とは，x を含む走査集合の族をいう．すなわち，

$$\mathrm{Super}(x)=\{H_{\mathrm{relaxed}}(y)|t_y\in D\wedge x\in H_{\mathrm{relaxed}}(y)\}$$

である．なお，H_{relaxed} は，問合せの種類に応じて 3 章の各アルゴリズムで求まる走査集合とする．

次に，走査集合 H が与えられたときに検索意図 x を推測する確率 $p(x|H)$ について考える．この条件付き確率 $p(x|H)$ が走査集合から検索意図が推測される**逆推測リスク**である．これが，完全なプロトコルにおける最大の問合せリスク以下であるとは，$\forall t_x\in D$，$\forall H\in\mathrm{Super}(x)$ に対して，

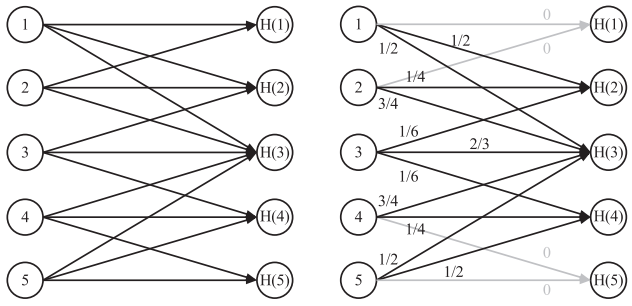
$$p(x|H)\le\max_{t_y\in D}\mathrm{Risk}(y|D) \tag{2}$$

が成り立つことである．また，$p(x)=\mathrm{Freq}(x)$ より

$$p(x|H)=\frac{p(H|x)\mathrm{Freq}(x)}{\sum_{t_y\in H}p(H|y)\mathrm{Freq}(y)} \tag{3}$$

と分解できる．ここで，$p(H|x)$ はある検索意図 x に対して走査集合 H を選択する確率である．我々は，条件 (2) を満たす $p(H|x)$ を求め，その確率を用いて走査集合を選択する．式 (3) より，$p(H|x)$ の決定問題は次のようになる．

問題 5.1　$\forall t_x\in D,\forall H\in\mathrm{Super}(x)$ に対し，



(a) 検索意図と包含走査集合　　(b) 走査集合の選択確率

図 9　走査集合の確率的選択を表すグラフ

Fig. 9　Graphs showing how the probabilities for handled sets can be calculated.

$$p(H|x)\mathrm{Freq}(x)\le\max_{t_z\in D}\mathrm{Risk}(z|D)\sum_{t_y\in H}p(H|y)\mathrm{Freq}(y)$$

を満たす $p(H|x)$ を求める．

この問題を解くために，検索意図と走査集合の関係を二部グラフを用いて表現する．この二部グラフは，検索意図の集合すなわち $[1,n]$ と，可能なすべての走査集合を V_h を頂点とする．そして，検索意図 x を表す頂点から，包含走査集合族 $\mathrm{Super}(x)\subset V_h$ に含まれる走査集合 $H\in\mathrm{Super}(x)$ を表す頂点に枝 $e_{x,H}$ を張る．この枝の集合を E と書くと，$G=([1,n],V_h,E)$ は二部グラフとなる．

例 5.2　例 5.1 における検索意図と走査集合の関係を表す二部グラフは**図 9** (a) のようになる．

$p(H|x)$ の計算は，この二部グラフをもとにしたフローネットワークを用いて各枝 $e_{x,H}$ に $p(H|x)\mathrm{Freq}(x)$ をフローとして割り当てることで行う．二部グラフ G をもとにしたフローネットワークは次のようになる．まず．ソース s とシンク t を追加し，ソース s と検索意図 x および走査集合 $H\in V_h$ とシンク t を枝で結ぶ．容量関数 c を次のように定める．ソース s と検索意図 x を結ぶ枝の容量はその検索意図の頻度 $c(e_{s,x})=\mathrm{Freq}(x)$ を用いる．検索意図 x と走査集合 H を結ぶ枝の容量はその検索意図の頻度 $c(e_{x,H})=\mathrm{Freq}(x)$ を用いる．そして，走査集合とシンクを結ぶ枝の容量は $c(e_{H,t})=1$ とする．次に，各枝の流量比関数 α を検索意図 x 流量比は $\alpha(x)=1$，走査集合 H 流量比は完全なプロトコルにおける最大リスク $\alpha(H)=\max_{x\in D}\mathrm{Risk}(x)$ と定める．このようにして得られた (s,t) フローネットワーク $G_f=((s,t),[1,n],V_h,E,c,\alpha)$ を考えると，$p(H|x)\mathrm{Freq}(x)$ の割当て問題 5.1 は，最大流量問題の一種である点平衡フローグラフ問題（vertex-balanced flow problem）[8] に帰着される．点平衡フローグラフ問題とは，最大流問題において頂点 u から v に流入する流量 $\mathrm{flow}(u,v)$ に対して，$\mathrm{flow}(u,v)\le\alpha(v)\sum_x\mathrm{flow}(x,v)$ という制約を付けた問題である．

例 5.3　例 5.1 に対して問題 5.1 の条件を満たす確率を求めると図 9 (b) のようになる．すなわち，検索意図が 1 の場合，走査集合として確率 1/2 で $H_1(2)$ あるいは $H_1(3)$

を用いて問い合わせる．この確率を用いることで，たとえば走査集合が $H_1(2)$ のとき，攻撃者による逆推測リスクは $p(1|H_1(2)) = p(2|H_1(2)) = p(3|H_1(2)) = 1/3$ となる．また，走査集合が $H_1(3)$ の場合は $p(1|H_1(3)) = p(5|H_1(3)) = 1/12$，$p(2|H_1(3)) = p(4|H_1(3)) = 1/4$，$p(3|H_1(3)) = 1/3$ である．すなわち，攻撃者による逆推測リスクも，完全なプロトコルにおける最大のリスク $\max_{y \in D} \mathrm{Risk}(y|D) = 1/3$ 以下に抑えられている．

前章までに導入した制約緩和手法に加えてこの走査集合のランダム選択を用いる場合，利用者はあらかじめ公開された問合せ頻度情報から各検索意図に対する走査集合および，走査集合の選択確率を計算する必要がある．走査集合の計算コストは，データベースに含まれるアイテム数 n に対して一次元データベースの場合 $O(n)$ であり二次元データベースの場合 $O(n^2)$ である．走査集合の選択確率の計算は，点平衡フロー問題を $O(n^2 \log M(n, n^2))$ で解くアルゴリズム [9] を利用することができる．ここで，$M(n, n^2)$ は，n 頂点 n^2 枝からなるグラフに対して一般化最大流問題を解くのに要する計算量を表す．よって，事前準備として必要な計算量は，一次元データベースの場合 $O(n^2 + n^2 \log M(n, n^2))$ であり，二次元データベースの場合 $O(n^3 + n^2 \log M(n, n^2))$ となる．一方，3.3 節で議論したように，近似的な検索意図の頻度分布を利用するなど，頻度分布が変化しないと仮定すると，走査集合および走査集合の選択確率はあらかじめ計算しておくことができる．

6. 関連研究

cPIR [7] は，計算量をもとに問合せから検索意図が推測できないことを保証するプロトコルである．cPIR における走査集合は，他の多くのプライベート問合せを実現するプロトコルと同様に，検索意図 x によらずつねにデータベース全体となる．cPIR は様々な研究で用いられており，たとえば Ghinita らは cPIR を位置情報サービスに適用し [10]，Nakamura らは匿名での認証に用いている [11]．bbPIR [5] は，cPIR に k-匿名性 [12] の概念を導入することで，サーバにおける問合せ処理時間を改善する制約緩和プロトコルの一種である．利用者は，問合せ時に検索意図 x を含む幅 k の矩形をサーバに送信する．以降はその矩形に含まれているアイテムのみからなる部分データベースに対して cPIR プロトコルを実行する．したがって，問合せ処理時には，アイテム総数は本来の n から k に削減されており，サーバにおける問合せ処理時間を短縮できる．ただし，1 章で議論したようにこの制約緩和手法は安全ではない．

暗号化データベース [13], [14], [15] も部分的にプライベート問合せを実現している．暗号化データベースでは，すべてのアイテムは暗号化されてデータベースに格納されている．研究の目的は，この暗号化されたアイテムの中から検索意図に合致するデータを発見する方法である．各アイテムは暗号化されており，問合せも何らかの符号化された値からなる．バケット化 [13], [16] は，アイテムのドメインをいくつかのバケットに分割しアイテムの番号の代わりにそのアイテムが属するバケットの名前を付加する．この手法の場合，ある検索意図 x に対する走査集合は，x が属するバケットに等しくなり制約緩和が行われていると見ることもできる．しかし，このバケットの選び方と検索意図頻度の関係，および問合せのリスクまでを議論している研究は我々の知る限りない．本論文の成果を応用することで安全なバケット分割方法を構築できる．

順序保存暗号 [17], [18] を用いた問合せ頻度の秘匿化手法 [19] は，本研究とは異なる方法でプライベート問合せを実現している．これらは，暗号化後も大小比較が行え，かつ本来の検索意図頻度とは異なる任意の問合せ頻度を達成できる暗号を用いている．一方で，どの記事が人気であるのかといった検索意図の頻度を攻撃者が知っている場合，暗号化された問合せから検索意図を高確率で推定できる問題は議論されていない．我々の提案手法とは，完全なプロトコルにおける最大のリスクを超えないことを保証している点で異なっている．また，Lu は順序保存暗号を用い暗号化データ上に木構造を構築することで高速な検索手法を提案している [20]．この手法はサーバにおける計算コストが $O(\log n)$ を達成している．しかし，攻撃者が問合せの頻度を知っている場合に彼の手法が安全でないと同論文中でも議論されている．我々の提案手法は，問合せの頻度分布を考慮して制約緩和を行っている．よって，攻撃者が問合せの頻度分布を知っていたとしても安全であり，同時にサーバにおける計算コストを削減することができる．

秘密通信技術 [21], [22] を用いることでも，プライベートに問合せ処理を行うことができる．これらの秘密得通信技術はどのクライアントがサーバに問い合わせているのかを秘匿する．したがって，ある利用者がサーバに検索意図 x を直接送信したとしても，サーバからは問合せ元を特定することができない．言い換えれば，サーバは検索意図を特定の利用者に結び付けることができず，この意味で利用者はプライベートな問合せ処理を行うことができる．一方，通常のプライベート問合せを実現するプロトコルでは，問合せ元の利用者ではなく検索意図を秘匿する．検索意図を秘匿する手法の利点は，サーバが利用者を識別でき認証処理などを行える点にある．たとえば，ニュースサイトの例で，利用者に一部の記事のみを閲覧できる無料利用者とすべての記事を閲覧できる有料利用者がいるような場合を考える．この場合，プライベートな問合せ処理だけではなく認証機能も必要となる．秘密通信技術を用いた場合では，このような要求に応えることはできない．

7. おわりに

本論文では，問合せの頻度情報を基にしたプライベート問合せのための制約緩和手法を提案した．既存のプライベート問合せでは，サーバにデータベース全体を走査させることで安全性を保証しているが問合せ処理コストが大きいという問題があった．提案手法はこの制約を緩和するが，緩和後の問合せリスクは制約緩和を行わない完全なプロトコルにおける最大のリスクは超えないことを保証している．すなわち，我々の緩和手法はサーバにおける問合せ処理コストを削減する一方で，問合せリスクは一般的に許容できると考えられている範囲に収めることができる．また，検索意図と走査集合の対応関係を知っている強い攻撃者に対しても，走査集合から検索意図の推測リスクを完全なプロトコルにおける最大リスク以下に抑える拡張方法についても提案した．

本論文では，ガウス分布と Zipf 分布に従う 2 種類の問合せを用いて提案手法を評価した．そして，問合せリスクが完全なプロトコルにおける最大のリスクを超えないことおよびサーバによる問合せ処理コストが削減されたことを確認した．特に，問合せ処理コストの期待値から，提案手法は問合せ頻度の少ないアイテムへの検索だけではなく全体として問合せ処理コストを削減していることが分かる．さらに，cPIR プロトコルを用いたデータベースを用意し実際に問合せ処理時間を計測することで，提案手法が多くの場合処理時間を短縮することを確認した．

謝辞　本研究は，中島記念国際交流財団および最先端研究開発プログラム「超巨大データベース時代に向けた最高速データベースエンジンの開発と当該エンジンを核とする戦略的社会サービスの実証・評価」の助成を受けました．

参考文献

[1] Pappas, V., Raykova, M., Vo, B., Bellovin, S.M. and Malkin, T.: Private search in the real world, *Proc. 27th Annual Computer Security Applications Conference*, New York, NY, USA, pp.83–92, ACM Press (2011).

[2] Chor, B., Goldreich, O., Kushilevitz, E. and Sudan, M.: Private Information Retrieval, *J. ACM*, Vol.45, No.6, pp.965–982 (1998).

[3] Kawamoto, J. and Yoshikawa, M.: Private Range Query by Perturbation and Matrix Based Encryption, *Proc. 6th IEEE International Conference on Digital Information Management*, Melbourne, Australia, pp.211–216, IEEE Computer Society (2011).

[4] Boneh, D., Kushilevitz, E., Ostrovsky, R. and Skeith, III, W.E.: Public Key Encryption that Allows PIR Queries, *Proc. 27th Annual International Cryptology Conference on Advances in Cryptology*, Santa Barbara, CA, USA, pp.50–67, Springer (2007).

[5] Wang, S., Agrawal, D. and Abbadi, A.E.: Generalizing PIR for Practical Private Retrieval of Public Data, *Proc. 24th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy*, Rome, Italy, pp.1–16, Springer (2010).

[6] Shang, N., Ghinita, G., Zhou, Y. and Bertino, E.: Controlling data disclosure in computational PIR protocols, *Proc. 5th ACM Symposium on Information, Computer and Communications Security*, New York, NY, USA, pp.310–313, ACM Press (2010).

[7] Kushilevitz, E. and Ostrovsky, R.: Replication Is Not Needed: Single Database, Computationally – Private Information Retrieval, *Proc. 38th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, IEEE Computer Society, pp.364–373 (1997).

[8] Nakayama, A.: NP-Completeness and Approximation Algorithm for the Maximum Integral Vertex-Balanced Flow Program, *Journal of the Operations Research Society of Japan*, Vol.34, No.1, pp.13–27 (1991).

[9] Nakayama, A. and Su, C.F.: A Binary Search Algorithm for the Generalized Maximum Balanced Flow Problem, 日本オペレーションズ・リサーチ学会秋期研究発表会アブストラクト集，pp.206–207 (2000).

[10] Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C. and Tan, K.-L.: Private Queries in Location Based Services: Anonymizers are Not Necessary, *Proc. 28th ACM SIGMOD International Conference on Management of Data*, Vancouver, BC, Canada, pp.121–132, ACM Press (2008).

[11] Nakamura, T., Inenaga, S., Ikeda, D., Baba, K. and Yasuura, H.: Anonymous Authentication Systems Based on Private Information Retrieval, *Proc. 1st International Conference on Networked Digital Technologies*, Ostrava, Czech Republic, pp.53–58, IEEE Computer Society (2009).

[12] Sweeney, L.: k-Anonymity: A Model for Protecting Privacy, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol.10, No.5, pp.1–14 (2002).

[13] Hacigumus, H., Iyer, B., Li, C. and Mehrotra, S.: Executing SQL over Encrypted Data in the Database-Service-Provider Model, *Proc. 21st ACM SIGMOD International Conference on Management of Data*, Madison, WI, USA, pp.216–227, ACM Press (2002).

[14] Wang, Z.-F., Dai, J., Wang, W. and Shi, B.-L.: Fast Query over Encrypted Character Data in Database, *Computational and Information Science*, Vol.4, No.4, pp.1027–1033 (2005).

[15] Wang, H. and Lakshmanan, L.V.S.: Efficient Secure Query Evaluation over Encrypted XML Databases, *Proc. 32nd International Conference on Very Large Data Bases*, Seoul, Korea, pp.127–138, VLDB Endowment (2006).

[16] Hore, B., Mehrotra, S. and Tsudik, G.: A Privacy-Preserving Index for Range Queries, *Proc. 30th International Conference on Very Large Data Bases*, Toronto, ON, Canada, pp.720–731, VLDB Endowment (2004).

[17] Boldyreva, A., Chenette, N., Lee, Y. and O'Neill, A.: Order-Preserving Symmetric Encryption, *Proc. 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Cologne, Germany, pp.224–241, Springer-Verlag (2009).

[18] Boldyreva, A., Chenette, N. and Neill, A.O.: Order-Preserving Encryption Revisited: Improved Security Analysis and Alternative Solutions, *Proc. 31st International Cryptology Conference*, Santa Barbara, CA, USA, pp.578–595 (2011).

[19] Agrawal, R., Kiernan, J., Srikant, R. and Xu, Y.: Order Preserving Encryption for Numeric Data, *Proc. 23rd*

ACM SIGMOD International Conference on Management of Data, New York, NY, USA, pp.563–574, ACM Press (2004).
[20] Lu, Y.: Privacy-Preserving Logarithmic-time Search on Encrypted Data in Cloud, *Proc. 19th Annual Network & Distributed System Security Symposium*, San Diego, CA, USA (2012).
[21] Reed, M.G., Syverson, P.F. and Goldschlag, D.M.: Anonymous connections and onion routing, *IEEE Journal on Selected Areas in Communications*, Vol.16, No.4, pp.482–494 (1998).
[22] Ren, J. and Wu, J.: Survey on anonymous communications in computer networks, *Computer Communications*, Vol.33, No.4, pp.420–431 (2010).

付　　録

図 5 に示したアルゴリズム Find $H_2(x)$ の計算量が $O(n^2)$ となることを示す．まず，NegOnes(h, w) はすべての要素が -1 の $h \times w$ 行列を作成する関数である．発見すべき部分行列は 4 つの変数，左上の位置を表す (i, j)，幅 w，および高さ h からなる．本アルゴリズムは，幅と高さを変えて可能な部分行列の形を列挙する．そして，各形ごとに左上の点を変え，検索意図 (e, g) を含む候補の中で最適な部分行列を探す．

計算量を削減するため，D' の部分行列の和を保持する C と M という 2 つの行列を用いる．C を列行列と呼び，$h \times 1$ の大きさからなる D' の部分行列の和からなり，$C_{i,j} = \sum_{k=i}^{i+h-1} D'_{k,j}$ である．すべての列の和を保持する代わりに，本アルゴリズムでは，$rangeC[0] \leq i \leq rangeC[1]$，$rangeC[2] \leq j \leq rangeC[3]$ で与えられる C の部分行列のみを保持する．なお，$rangeC$ は今考慮している C の部分行列を (左, 右, 上, 下) の各点からなる四つ組みで記録しており，$rangeC[i]$ は i 番目の値である．h が増加するごとに，各行の和を 1 要素として C は伸張する．C の更新手続きを updateC と書く．

M は，$h \times w$ の行列で D' の部分行列の和を記録する和行列である．C が D' の要素和から計算されるように，M は C の要素和から計算できる．C と同様に M に対しても考慮すべき D' の部分を $rangeM$ で表す．$M_{i,j}$ は (i, j) 成分を左上とし，高さ h，幅 w からなる D' の部分行列内の要素和を保持する．updateM は M の更新手続きである．C と異なり，w が増えると C に記録されている列の和を基に古い M に列を追加することで M を更新するが，h の更新時には M は一から再計算される．

計算結果を再利用できるため，h や w が更新されたときに updateC と updateM を計算するコストは $O(n)$ である．h と w の可能な組合せは n 通りであり，updateC と updateM は $O(n)$ 回呼ばれる．したがって，アルゴリズム Find $H_2(x)$ に要するコストは $O(n^2)$ となる．

川本 淳平　（正会員）

2007 年京都大学工学部情報科学科卒業，2012 年同大学大学院情報学研究科博士後期課程修了．京都大学博士（情報学）．2012 年 4 月より筑波大学システム情報系研究員．クラウドデータベースにおけるプライバシおよびプライバシ保護データマイニングの研究に従事．日本データベース学会，人工知能学会，ACM 各会員．

Patricia L. Gillett

Received a B.Math. in computational mathematics from the University of Waterloo in 2010. Since 2010, she has been a Ph.D. student at École Polytechnique du Montréal. From 2010 to 2012 she was a student member of a MITACS/Mprime project, and since 2010 she has been a student member of the multi-university research centre GERAD. Her research interests include combinatorial optimization, complementarity problems, and semidefinite programming.

佐久間 淳

1997 年東京工業大学生命理工学部生物工学科卒業，2003 年 3 月同大学大学院総合理工学研究科知能システム科学専攻博士後期課程修了．博士（工学）．同年 4 月日本アイ・ビー・エム株式会社入社，東京基礎研究所に配属．2004 年 7 月東京工業大学総合理工学研究科助手，2007 年 4 月同助教，2009 年 4 月筑波大学システム情報工学研究科コンピュータサイエンス専攻准教授，2009 年 10 月から 2013 年 3 月科学技術振興事業団さきがけ研究員兼任，現在に至る．機械学習と知識発見，セキュリティとプライバシの研究に従事．

（担当編集委員　金政 泰彦）