類似検索におけるプライバシ保護のためのクエリ監査法

荒井　ひろみ† 津田　宏治‡ 佐久間　淳§

†理化学研究所 情報基盤センター
hiromi.arai@riken.jp

‡産業技術総合研究所 生命情報工学研究センター
tsuda@cbrc.jp

§筑波大学 システム情報工学系
jun@cs.tsukuba.ac.jp

あらまし 本論文ではプライベート情報を含むデータベースに対する類似検索において，データベースの応答がプライバシ保護要件を満たしているかを判定する (クエリ監査) 手法を提案する．データベースは任意の複数クエリレコードそれぞれに対し類似するレコードの ID 集合を返すとする．我々はこのようなクエリ監査を定式化し，プライバシ基準としてデータベースのプライベート情報が推測される確率を導入した．さらにプライバシ基準の計算を数え上げ問題として記述し監査を高速かつ正確に行うアルゴリズムを導入した．実データを用いた実験で本手法の計算効率及び監査結果の検証を行った．

Query auditing for privacy preserving similarity search

Hiromi Arai† Koji Tsuda‡ Jun Sakuma§

†RIKEN Advanced Center for Computing and Communication
‡Computational Biology Research Center
National Institute of Advanced Industrial Science and Technology
§Department of Computer Science, University of Tsukuba

Abstract In this paper, we propose a query auditing method for similarity searches that examines whether database responces satisfy privacy preserving requirements. We assume that the database answers a set of similar records's IDs against each query. We introduce the probability of a certain private value given database responsesas as a privacy measure. We describe auditing with such a privacy measure as an enumeration problem and apply the efficient and accurate algorithm. The computational efficiency and the result of this auditing method is examined on the real world dataset.

1 はじめに

近年，医療データ，購買履歴，SNS などに代表される個人情報データベースの増化，巨大化に伴い，プライバシを保護したデータ利用技術の重要性が高まっている．プライベート情報を含むデータベースの利用において，そのデータベース由来情報，例えば検索結果，分析結果の開示におけるプライバシの保証は非常に重要である．

データベースに対する類似検索は既病歴や購買履歴などの様々な記録，文章や画像などから特徴ベクトルを抽出した検索など様々な応用対象が挙げられる．しかし，個人情報を含むデータベースの利用においては，類似検索結果の開示においてデータベースに含まれるプライベート情報の保護が必要である．例えば疾患データベースに対する類似患者検索，購買履歴などの

個人情報のデータベースを用いた類似ユーザー検索などが挙げられる．

類似検索結果の開示において，プライベート情報の漏洩の危険性の想定される例をあげる．例えば，類似患者検索において，データベースが問い合わせ患者情報から既病歴の類似した患者を検索し，患者の連絡先を返すとする．わずかに異なる２つのクエリに対する応答を考える．２回目のクエリが最初のクエリに加え，既病歴に乳がんを含むという条件が追加されたとする．そうすると，２番目の応答にのみ類似であると回答されたレコードはおそらく乳がんを既病歴に持つ患者であろう．

データベースの応答において，プライベート情報の漏洩があったか否かをチェックするクエリ監査 (query auditing) と呼ばれる方式がある．クエリ監査ではこれまでに集約クエリや SQL クエリについて研究がなされてきたが，本研究で扱う類似検索に利用できる方式はまだ提案されていない [7].

本稿では，そもそも類似検索においてプライバシは漏洩するのか，といった疑問を立脚点とする．我々は類似検索のクエリ監査を定式化し，その監査を効率的に行う方法を提案した．さらに，これまでのクエリ監査は実際の監査結果の検証はほとんど行われてこなかったためどのような問い合わせによってプライベート情報が漏洩するのか，といった検証がほとんどされていなかった．我々はクエリモデルを仮定し，それに対する実データを用いた監査結果の検証を行った．

1.1 関連研究

データベース問い合わせにおける応答におけるプライバシを扱う研究は以下のようである．データベースの情報出版におけるプライバシ保護全般については，その応答をプライバシを保護するためにランダム化させる方式が提案されてきた．主な研究としては差分プライバシ [3] などがある．しかし，このような方式ではユーザーに渡す情報にランダムノイズが乗るため，正確な応答ができない．

クエリ監査はデータベースが特定のユーザーのクエリを監視し，これまでの応答からプライベート漏洩の有無を判定する方法である．従来研究では応答が正確な場合を扱い，プライベート情報の漏洩が，データベース応答を得たユーザーが攻撃者となりプライベート情報を推論すると想定し，プライバシ基準として，この推論が成功する確率を用いていた [7]．我々の提案方式でもこれを採用し，類似検索のクエリ監査のための推論の定量的評価方法を提案する．

クエリ監査の方式はクエリの集合をバッチで処理する offline 監査，逐次的にクエリを受け付け判定を行う online 監査に大別できる．これらの方式ではレコードに実数値または boolean の値を持つデータベースを扱う．また，プライバシ漏洩のチェックに加え，プライバシが漏洩する場合にはクエリを拒否することでプライバシを防ぐ方式も提案されている．

offline 監査では，1 次元のデータベースに対する複数の集約クエリについて，推論攻撃が完全に成功する (full disclosure) 場合を判定する方式が提案されている．これまでに，複数のクエリの攻撃評価は計算量が大きく NP-hard であること，一部の問題は多項式時間で評価できることが示されている [2]．この解法は本稿で問題にする類似検索には応用できない．また，推論攻撃の定量評価を行う方式が提案されているが，具体的な評価方法は示されていない [4].

online 監査では，逐次的に offline 監査同様の方式でプライベート情報漏洩のチェックを行うことも可能である．しかし，そのようなチェックに基づいてプライバシ保護のためのクエリ拒否を行うと，プライバシ漏えいが起きる危険がある．クエリを拒否することで，そのようなクエリから情報漏洩がおきるプライベート情報の値を推定されてしまうのである．そのような情報漏洩を防ぐため，クエリのパターンから情報漏洩の可能性を評価し，確率的に応答を行うことでプライバシ保護を実現する simulatable auditing [6] が提案されている．これらは逐次監査が可能であることで実用上のメリットが大きいが，1 次元データベースにおける sum クエリついてのみ方式が提案されているにとどまっている．また，この方式では監査のための計算量が大きく，確率評価のために近似手法を用いている．

以上のようにクエリ監査分野ではこれまで 1 次元のデータベースに対する aggregate query が中心であった．bit-vector で表現されるような高次元レコード間の類似検索におけるクエリ監査は本研究が初めての試みである．クエリ監査では，これまでに所与の応答を与えうるデータベースを求めるアプローチで推論攻撃の評価を行ってきた．我々の扱う類似検索において，このような評価を行う方式はこれまで提案されていない．

本論文の構成は以下のようである．まず類似検索モデルを導入し（２章），そのような類似検索の応答を用いたプライバシ基準を定式化する（３章）．さらに，４章でプライベート情報推定を定式化し，5 章でその理論的検証を行い，プライベート情報漏洩の可能性を示す．6 章においてデータベースが監査を行うための Binary Decision Diagram(BDD) を導入，BDD を用いた監査法を提案する．7 章でクエリモデルを導入し，提案の監査法を用いた実データにおける実験を 8 章で示し，その結果について考察する．

2 類似検索モデル

データベースを行列 $X =\in \{0,1\}^{n\times m}$ とする．データベースの各 i 行が各レコード i に対応しているとする．第 i 行をビットベクトルを $\mathbf{x}_i = (x_1, x_2, \ldots, x_m) \in \{0,1\}^m$ と表現する．X の各 j 列は属性 j に対応し，属性値 $x_{ij} = 1$ は，レコード i が属性 j を持つことを表すとする．

クエリをデータベースレコードと同じ attribute を持つビット列 $\mathbf{q} = (q_1, q_2, \ldots, q_m) \in \{0,1\}^m$ とし，データベースは受け付けたクエリについて類似するレコード ID を返すとする．例えば，類似患者検索において，検索は患者の既病歴の比較で行い，返答にはレコード ID を返すとする．ユーザーはレコード ID を患者の公開している連絡先情報等にアクセスするなどに利用すると想定する．データベースは複数のクエリを受け付けるとし，あるユーザーから受け付けたクエリの列を $Q = (\mathbf{q}_1, \ldots, \mathbf{q}_c)$, それらに対する回答の列を $A = (a_1, \ldots, a_c)$ とする．データベースの回答 a_ℓ は，クエリ $\mathbf{q}_\ell$ と類似するデータベースレコードの ID 集合とする．

レコード間類似度　ここでクエリと類似するレコードを算定するために，bit ベクトル対の類似度を導入する．bit ベクトル $\mathbf{x}_i, \mathbf{x}_j$ 間の類似度を一般的に $\mathrm{score}(\mathbf{x}_i, \mathbf{x}_j)$ と表記する．

類似度として，今回は common neighbor

$$CN(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$$

を用いる．他にも bit ベクトル対の類似度はハミング距離やコサイン距離などがあるが，これら他の類似度を用いた場合の詳細な解析は今後の課題とする．

本稿では a_ℓ として $\mathbf{q}_\ell$ との類似度が高いデータベースの Top-k レコードの ID 集合を返すとする．順位付けは同順位を扱う midrank を用いる．データベースはユーザーに a_ℓ に加え，Top-k レコード中の最下位レコードとクエリの類似度 θ_ℓ を応答するとする．すなわち a_ℓ は類似度がしきい値 θ_ℓ より大きいレコード ID の集合

$$a_\ell = \{i | \mathrm{score}(\mathbf{x}_i, \mathbf{q}_\ell) \geq \theta_\ell\}$$

である．ここで，$\Theta = (\theta_1, \ldots, \theta_c)$ とする．

3 データベースプライバシモデル

クエリ監査に用いるプライバシの定量のために，クエリ Q に対する出力 A，Θ を得たユーザーがあるプライベート情報 x_{ij} の値を確率的推論する攻撃を考える．推論は以下のように行われるとする．事前知識として $v \in \{0,1\}$ について $\Pr[x_{ij} = v] = 1/2$ を想定する．応答を受け取った後の事後確率は

$$\Pr[x_{ij} = v | A, \Theta]$$

となる．

ある情報のプライバシ基準として以下を定義する．

Definition 1 (α, β)-*private :* データベース X について，検索者はクエリ Q に対する正確な応答 A，Θ を保持している．データベースのあるレコードのあるプライベートな属性値 x_{ij} につ

いて，応答を受け取った後の事後確率が開区間 (α, β) について

$$\Pr[x_{ij} = 1|A, \Theta] \in (\alpha, \beta)\ (0 \leq \alpha \leq \beta \leq 1) \tag{1}$$

であるとき，x_{ij} は応答 A について (α, β)-*private* であるとよぶ．

あるプライベート情報 x_{ij} が (α, β)-private を満たさないとき，x_{ij} について (α, β)-privacy breach であるとする．これは，閉区間や半開区間についても同様に定義できるとする．ここで，α，β が $1/2$ に近いほど，高い確率で属性を推定されにくく情報が漏洩していないと捉えることができる．

4　プライベート情報の値の推定

データベースの応答から推測される属性値の確率は，その応答をしうるデータベースが全て等確率に存在すると仮定し，それらの場合の数によって評価する．

クエリ Q に対して出力 A，Θ を与えうるデータベース $X' = \{\mathbf{x}_i | i = 1, \ldots, n\}$ の集合を

$$S(Q) = \{X' | A, \Theta, X' \in \{0,1\}^{n \times m}\}$$

とおく．また，S のうち $x'_{ij} = v \in \{0,1\}$ となる部分集合を

$$S(Q, x_{ij} = v) = \{X' | X' \in S \wedge x'_{ij} = v\}$$

とおく．これらを用いると，

$$\Pr[x_{ij} = v|A, \Theta] = \frac{|S(Q, x_{ij} = v)|}{|S(Q)|} \tag{2}$$

となる．

ここで，データベースの回答 a_ℓ の定義より，

$$S_i(Q) = \{\mathbf{x}_i | A, \Theta, \mathbf{x}_i \in \{0,1\}^m\},$$
$$S_i(Q, x_{ij} = v) = \{\mathbf{x}_i | \mathbf{x}_i \in S_i \wedge x_{ij} = v\}$$

とすれば，$S(Q)$ や $S(Q, x_{ij} = v)$ はこれらの直積集合として

$$S(Q) = \prod_{h \in 1 \ldots n} S_h(Q),$$
$$S(Q, x_{ij} = v) = S_i(Q, x_{ij} = v) \times \prod_{h \in 1 \ldots n \setminus i} S_h(Q)$$

となる．よって式 (2) は

$$\Pr[x_{ij} = x|A, \Theta] = \frac{|S_i(Q, x_{ij} = v)|}{|S_i(Q)|} \tag{3}$$

とできる．

また，$S_i(Q)$ は，各クエリ $\mathbf{q}_\ell$ への応答が満たす集合 $S_i(\mathbf{q}_\ell) = \{\mathbf{x}_i | a_\ell, \theta_\ell, \mathbf{x}_i \in \{0,1\}^m\}$ の積集合として，

$$S_i(Q) = \cap_{\ell \in 1 \ldots c} S_i(\mathbf{q}_\ell) \tag{4}$$

と書ける．

5　プライベート情報漏洩の検討

データベースのプライバシが $[\alpha, \beta]$-privacy breach の意味で漏洩する可能性を理論的に検証する．

ある応答 a_ℓ，θ_ℓ を返すデータベースエントリ $\mathbf{x}_i$ の満たすべき条件は

$$\mathrm{score}(\mathbf{x}_i, \mathbf{q}_\ell) - \theta_\ell = \begin{cases} \geq 0 & (\text{if } i \in a_j) \\ < 0 & (\text{if } i \notin a_j) \end{cases} \tag{5}$$

である．これをもとに，クエリ $\mathbf{q}_\ell$ によるプライベート情報漏洩を検証する．

まず，$|S_i|$ を検証する．式 (5) より，クエリ $\mathbf{q}_\ell$ に対して a_ℓ，θ_ℓ を取りうるエントリ i に関するスコアの集合 $V(i, \ell)$ を考える．

$$V(i,\ell) = \begin{cases} \{v | \mathbf{q}_\ell, \mathbf{x}'_i \in \{0,1\}^m, \mathrm{score}(\mathbf{x}'_i, \mathbf{q}_\ell) = v, v \geq \theta_\ell\} \\ \qquad\qquad (\text{if } i \in a_\ell) \\ \{v | \mathbf{q}_\ell, \mathbf{x}'_i \in \{0,1\}^m, \mathrm{score}(\mathbf{x}'_i, \mathbf{q}_\ell) = v, v < \theta_\ell\} \\ \qquad\qquad (\text{otherwise.}) \end{cases}$$

とする．ここで，S_i はある応答をしうるデータベースの集合であったので，

$$S_i(v) = \{\mathbf{x}' | \mathrm{score}(\mathbf{x}_i, \mathbf{q}_\ell) = v, \mathbf{x}' \in \{0,1\}^m\}$$

とすると，

$$S_i(\mathbf{q}_\ell) = \cup_{v \in V(i,\ell)} S_i(v) \tag{6}$$

と書ける．$S_i(\mathbf{x}_i, \mathbf{q}_\ell)$ についても同様である．

クエリ $\mathbf{q}_\ell$ が問い合わせ対象としている属性の集合を

$$Q_\ell = \{j | q_{\ell j} = 1\}$$

と定義する．

類似度が common neighbor の場合，$q = |Q_\ell|$, $v = CN(\mathbf{x}_i, \mathbf{q}_\ell)$ とおいて

$$|S_i(v)| = {}_q\mathrm{C}_v \cdot 2^{m-q},$$
$$|S_i(x_{ij} = 1, v)| = \begin{cases} {}_{q-1}\mathrm{C}_{v-1} \cdot 2^{m-q} & (\text{for } j \in Q_\ell) \\ {}_q\mathrm{C}_v \cdot 2^{m-q-1} & (\text{otherwise.}) \end{cases} \quad (7)$$

特に，$V(i, \ell) = \{v\}$ である場合は

$$P(x_{ij} = 1 | CN(\mathbf{x}_i, \mathbf{q}_\ell) = v) = \begin{cases} \frac{v}{q} & (\text{for } j \in Q_\ell) \\ \frac{1}{2} & (\text{otherwise.}) \end{cases}$$

となる．ここで，例えば $q = 1$ のとき，$V(i, \ell) \in \{0, 1\}$ となるので，

$$P(x_{ij} = 1 | A, \Theta) = \begin{cases} 1 & (\text{for } j \in Q_\ell) \\ \frac{1}{2} & (\text{otherwise.}) \end{cases}$$

となり，$j \in Q_\ell$ なる x_{ij} について $(0, 1)$-privacy breach が常に起きる．

複数クエリの場合を考える．式 (7) より，クエリ $\mathbf{q}_\ell$ の応答からは $j \in Q_\ell$ に関する情報が得られる．より詳しくみると，複数クエリの応答をしうるデータベースは式 (3)，(6) より，

$$\begin{aligned} S_i(Q) &= (\cup_{v_1 \in V(i,1)} S_i(v_1)) \cap (\cup_{v_2 \in V(i,2)} S_i(v_2)) \\ &= \cup_{v_1 \in V(i,1)} \cup_{v_2 \in V(i,2)} (S_i(v_1) \cap S_i(v_2)) \end{aligned}$$

である．

クエリ列 $Q = (\mathbf{q}_1, \mathbf{q}_2)$ を考える．簡単のために，$Q_1 \subset Q_2$，$q_\ell = \|\mathbf{q}_\ell\|$，$q_\Delta = q_2 - q_1$ とし，$S_i(v_1) \cap S_i(v_2)$ を考える．

$$|S_i(v_1) \cap S_i(v_2)| = {}_{q_1}\mathrm{C}_{v_1} \cdot {}_{q_\Delta}\mathrm{C}_{v_2 - v_1} \cdot 2^{m-q_2}. \quad (8)$$

よって，例えば同じ属性を含むクエリを複数回問い合わせや，わずかに異なるクエリの問い合わせによりより属性値を高確率で推定することが可能であると考察される．

以上のように，単数もしくは複数のクエリからプライベート情報漏洩が起きる可能性が示された．よって，データベース応答からのプライベート情報漏洩の有無を調べるためにクエリ監査を行う必要がある．

実際にクエリ監査を行うためには式 (7) で表されるようなサイズが指数オーダーである集合の積集合のサイズを評価する必要がある．これは実直に実行すると指数オーダーの計算時間を要する．

ここで，$|S_i(Q)|$ の評価は全ての ℓ について式 (5) で表される不等式制約を満たしうるデータベース集合を数え上げる線形計画問題と捉えることもできる．次章以降，このアプローチに則り数え上げを行う計算アルゴリズムを導入し，クエリ監査を行う方式を示す．

6 BDD を用いた推定確率の計算

データベース応答によるプライバシ漏洩尺度である式 (3) を評価するために，Binary Decision Diagram (BDD) を導入する．BDD はグラフ構造によって論理関数を表現するデータ構造である．BDD を用いると，論理関数の値をすべての変数について場合分けした結果を二分決定木で表し，これを縮約してコンパクトに表現することができる．BDD は基本的には計算機の主記憶上にすべてのデータを置いて処理するために，高速に処理を行うことができる．

BDD の論理表現を応用し，入力が 2 値 $0, 1$ のベクトルである整数線形計画問題の解を表現するような BDD を多項式時間で構築する方式が提案されている [1]．さらに，二つの BDD を入力とし，それらの 2 項論理演算 (AND,OR 等) の結果を表す BDD を直接生成することが可能であるため，複数の論理関数の制約下の bit ベクトルを表すことが可能である．

この方式を用い，式 (3) の評価を BDD を用いて行うことを考える．クエリ Q に対して応答 A，Θ を返す可能なデータベースレコード集合 $S_i(Q)$ は，全ての ℓ についての式 (5) を満たす線形計画問題として記述できる．さらに，$S_i(Q, x_{ij} = 1)$ は，全ての ℓ についての式 (5) に

加え $x_{ij} = 1$ を満たす線形計画問題である．以上により，[1] を用いて $S_i(Q)$ 及び $S_i(Q, x_{ij} = 1)$ を表す BDD を構築できる．この提案方式を BDD 類似検索監査 (BDD similarity search auditing, BSSA) とする．

7 クエリモデル

データベースの種類に応じてユーザーの典型的なクエリモデルを導入する．

履歴モデル データベースが，例えばユーザーの購買履歴や既病歴のような履歴を表す場合，各ユーザーの持つ属性は時間の経過に伴って単調に増加する．これをモデル化して，履歴モデルではクエリ列 $Q = (\mathbf{q}_1, \ldots, \mathbf{q}_c)$ は，

$$\|\mathbf{q}_1\|_1 \leq \|\mathbf{q}_2\|_1 \leq \cdots \leq \|\mathbf{q}_c\|_1,$$
$$\|\mathbf{q}_{\ell+1} - \mathbf{q}_\ell\|_1 = \|\mathbf{q}_{\ell+1}\|_1 - \|\mathbf{q}_\ell\|_1 = \Delta$$

となるように設定する．

8 実験

dataset. プライベート情報を含むデータベースのモデルデータとして，以下に示す ML100k データセットを作成した．GroupLens Research により公開されている 1000 users，1700 movies の 100,000 ratings からなる 100k データセット [5] をもとに映画視聴データ $X = \{0,1\}^{1000\times1700}$ を作成した．X の行は user，列は movie を示し，user i が movie j を評価していれば $x_{ij} = 1$，していなければ $x_{ij} = 0$ とした．すなわち各行 $\mathbf{x}_i$ は各ユーザーの映画視聴履歴レコードを表す．ここで，各ユーザーがある特定の映画を視聴したことがそのユーザーのプライベート情報とし，保護すべき対象と考える．すなわち，全ての i, j について x_{ij} がデータベースのプライベート情報とする．

settings. 実験には Intel(R) Core(TM) i3 (2core) 3.20GHz (CPU)，2GB (RAM) の Ubuntu 12.04.1 を用いた．パラメータは Top-k の $k = 10$ と設定し，他のパラメタ Δ，$|Q|$ やクエリの性質に対するプライベート情報漏洩の変化を調べた．

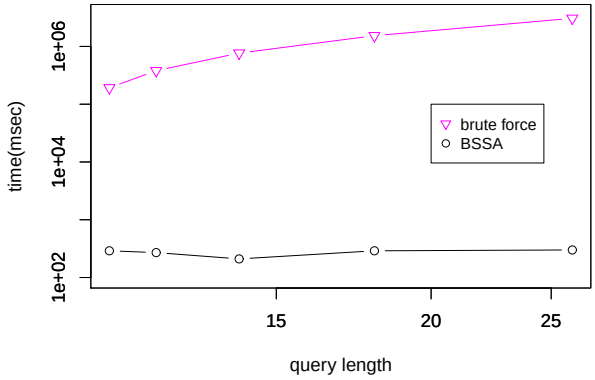

図 1: 履歴モデルで $|Q| = 2$ とした場合の $\|\mathbf{q}_2\|$ の大きさに対する計算時間の変化．

提案法 BSSA における線形制約問題の解を表す BDD の構築には Behle らの azove [1] を用いた．

データベース応答の設定は以下のようである．Top-10 レコードを返す類似検索のクエリ監査を行う．また，類似度は common neighbor のみを扱う．なお，今回はレコードがある属性値を持つことがプライバシと想定し，$[0-\beta)$-privacy breach がプライベート情報漏洩ととらえることにする．

クエリは以下のように 4 通りのパターンについて検討する．$\mathbf{q}_\ell \in \{0,1\}^{1700}$ なるクエリの集合 Q をユーザーのクエリモデルに従ってにランダムに生成する．MovieLens データセットは attribute によって bit が 1 となる頻度に差がある．プライベート情報漏洩の度合いがこの頻度に依存する可能性を考慮して，attribute を密度の高いところ，中間，低いところ，全領域と以下のように領域を定義する: $r_{dense} = [1, 200]$，$r_{mid} = [500, 700]$，$r_{sparse} = [1400, 1600]$ ，$r_{random} = [1, 1600]$. それぞれの attribute ドメインにおいて，bit が 1 平均頻度は 0.15, 9.3×10^{-2}, 4.8×10^{-3} ,7.0×10^{-2} である．これらの領域から属性をサンプルしてクエリを生成する．

8.1 計算時間

提案法 BSSA の要する計算時間を実験的に評価し，ナイーブな数え上げの場合と比較する．

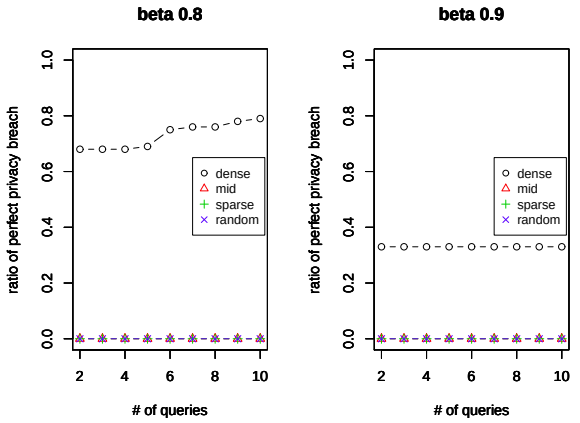

図 2: 履歴モデルでランダムに生成したクエリ Q の応答におけるプライベート情報漏洩．$\Delta = 1$ とした場合にある i，$j = Q_2 \backslash Q_1$ について属性値が $[0, \beta)$-private breach となる割合．(左)$\beta = 0.8$, (右))$\beta = 0.9$ の場合を示す．

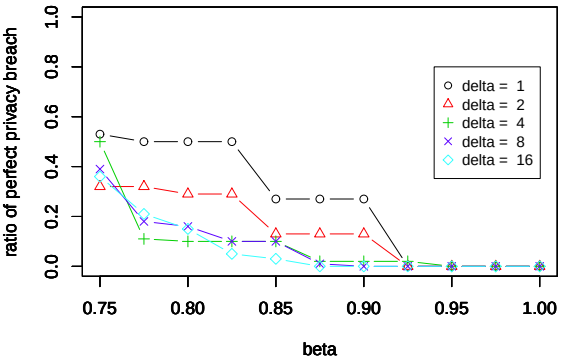

図 3: $|Q| = 2$，r_{dense} に関するクエリの場合の Δ に対するプライベート情報漏洩の β に対する変化．100 試行における $[0, \beta)$-private breach となる割合．

ここで，ナイーブな数え上げを brute force として，存在しうる $\{0,1\}^{\|\mathbf{q}_\ell\|_1}$ 全 $2^{\|\mathbf{q}_\ell\|_1}$ 個について全ての線形制約 (5) 及び $x_{ij} = 1$ を満たすかどうかで判定すればよい．本実験ではこの brute force の実装を C++で行った．なお，この方式は類似度が common neighbor の場合クエリの長さ $\|\mathbf{q}_c\|$ 及びクエリの数 c に線形に依存すると考えられる．

BDD を用いた数え上げには線形制約を満たす bit ベクトルを表す BDD の構築及び解の数の評価を要する．そのため，計算時間はクエリの長さ及び制約式の数に影響されると考えられる．

ここで，クエリの長さ $\|\mathbf{q}_1\| = 10$，$\|\mathbf{q}_2\| \in \{11, 12, 14, 18, 26\}$ とした場合の，$Q = (\mathbf{q}_1, \mathbf{q}_2)$ の応答についての監査に要する時間を調べた．これは，ある一つの属性値 x_{ij} についての時間であり，BSSA については BDD の構築及び解の数え上げの時間を測定した．r_{dense} を対象とした問い合わせの結果は図 1 のようになる．なお，他の領域 r_{mid}，r_{sparse}，r_{random} からクエリの属性をサンプリングした場合もほぼ同様の結果を得た．結果，BSSA は brute force と比較して，$O(10^{-3})$ 以下の計算時間ですみ，1 属性値について $O(1)$ 秒以下で監査が行えることがわかった．brute force の場合はクエリの長さに伴い計算時間が増加するが，BDD の構築時間はクエリの長さとの相関はあまり見られない．また，クエリの数 c を増やしても BDD の構築時間はほとんど変化がなかった．これにより，BSSA は brute force と比較して十分小さく現実的な計算時間でクエリ監査を実行できること，クエリの数や問い合わせ対象の属性数の増加に対して所要時間が大きく増えないことが考察される．

8.2 BSSA によるクエリ監査結果

$\|\mathbf{q}_1\| = 10$，$\Delta = 1$ とした場合，ある $i \in a_1$，$j \in Q_2 \setminus Q_1$ なる x_{ij} について，$P(x_{ij} = 1 | Q, \Theta)$ が $[0, \beta)$-private breach となる割合は図 2 のようになる．ここで，100 試行における割合を示している．この結果より，今回用いた履歴データベースでは r_{dense}，すなわちレコードが属性値を持つ頻度が高い属性に対する問い合わせからプライベート情報が比較的漏洩しやすいこと，クエリ回数が増えると $[0, \beta)$-privacy breach がおきる割合は増加する傾向が見て取れる．一方，頻度がある程度低い属性値に対する問い合わせからはプライベート情報は全く漏洩しなかった．これにより，属性値 1 が疎なデータベースにおいて類似検索におけるプライベート情報の漏洩は起きにくいことが考察される．

$[0, \beta)$-privacy breach がおきる割合の β に対する変化は図 3 に示すようになる．$|Q| = 2$，r_{dense} に関するクエリの結果の 100 試行における割合を示している．また，$\Delta \in \{2^0, 2^1, \ldots, 2^5\}$ についての結果をそれぞれ示す．これより，差分が大きくなるほど小さな β を設定してもプライバシが保護される傾向があること，$0.95 \leq \beta$ では差分が最小でもプライバシが保護されることが見て取れる．

9 終わりに

本稿では類似検索におけるクエリ監査問題を定式化し，監査法の導入およびプライベート情報の漏洩を検証した．その結果，我々の提案する BDD を用いたクエリ監査法は十分短い計算時間で実行でき，さらに，実データを用いた実験から類似検索においてスパースなデータベースに対するクエリや複数のクエリにおいてもその差分が大きければプライベート情報の漏洩が起きにくいことが示唆された．

差分プライバシのようなランダム化方法ではスパースなデータベースではランダム化の影響が大きくなる難点があったが，そのような場合に，ランダム化なしにプライバシ保護の保証を与え，かつデータの有用性を保った利用のためのアプローチとして本論文で扱ったような監査方法の利用が期待される．

本方式の応用として，明らかにプライベート情報漏洩の可能性が高いクエリは前もって受け付けないなどのデータベースの出版ポリシーの設定によりプライベート情報があまり漏洩しないようにすることが考えられる．また，監査結果を利用したクエリ拒否の方法に，漏洩しやすいクエリの応答優先順位を下げるなどの基準を導入することにより，ユーザーに応答可能な情報を増やすような応用も期待される．

10 謝辞

本研究は，理化学研究所における基礎科学特別研究員制度及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] Markus Behle and Friedrich Eisenbrand. 0/1 vertex and facet enumeration with bdds. In *Proceedings of the 9th Workshop on Algorithm Engineering and Experiments*, pages 158–165, 2007.

[2] Francis Chin and Gultekin Ozsoyoglu. Auditing for secure statistical databases. In *Proceedings of the ACM'81 conference*, pages 53–59. ACM, 1981.

[3] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, pages 265–284, 2006.

[4] Alexandre Evfimievski, Ronald Fagin, and David Woodruff. Epistemic privacy. *Journal of the ACM (JACM)*, 58(1):2, 2010.

[5] J. Herlocker, J. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Research and Development in Information Retrieval*. American Association of Computing Machinery, American Association of Computing Machinery, 8/1999 1999.

[6] K. Kenthapadi, N. Mishra, and K. Nissim. Simulatable auditing. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 118–127. ACM, 2005.

[7] S.U. Nabar, K. Kenthapadi, N. Mishra, and R. Motwani. A survey of query auditing techniques for data privacy. *Privacy-Preserving Data Mining*, pages 415–431, 2008.