類似検索における秘密情報漏えいの評価及び差分プライバシの保証

荒井　ひろみ †　　佐久間　淳 ‡

† 理化学研究所生命情報基盤研究部門
基礎科学特別研究員
hiromi@base.riken.jp

‡ 筑波大学 大学院 システム情報工学研究科
科学技術振興機構 さきがけ
jun@cs.tsukuba.ac.jp

あらまし 各レコードが秘密情報であるデータベースにおけるクエリに対する類似度によるランキング情報からの情報漏えいを検討する．データベースのプライベートレコードおよびクエリが長さ ℓ のビットベクトルとする．データベースはクエリと各レコードとの類似度によって降順に並べられたレコード index のランキングを応答する．多くのデータはビットベクトルで表現でき，またランキングは情報検索でよく用いられる手法であるため，この設定は秘密情報の情報検索も含め幅広く応用できると期待される．本論文ではこのようなランキングにおけるプライバシ保護と差分プライバシを保証するランキングについて検討する．

Privacy Breach and Differential Privacy of Similarity Search

Hiromi Arai†　　Jun Sakuma‡

†RIKEN Bioinformatics and System Engineering division, RIKEN Spetial Postdoctoral Researcher
‡Graduate School of SIE, University of Tsukuba / Japan Science and Technology Agency

Abstract We examine the database privacy of similarity based ranking for a database of private records where each record and queries for similarity search are both represented by a bit vector of length ℓ. In our setting, the database answer record indexes are arranged in descending order according to similarity to the query. This setting can be widely applied, including for retrieval from private records, because ranking is often used for information retrieval and many types of data can be expressed as bit vectors. In this paper, we examine privacy breaches caused by ranked indexes and differentially private ranked indexes.

1　はじめに

近年医療情報などのプライベートな情報のデータベース上への蓄積が進み，その利用が期待されているが，これらデータベースへのクエリ応答が秘密情報の漏洩を招くことは避ける必要がある．レコード間類似度に基づくデータベースレコードのランキングは主要な情報検索技術の一つである．本稿は類似検索における類似度ランキングが引き起こす秘密情報の漏えいについて検討し, その対策について考察する．以下に, 類似度によるランキングが秘密情報の漏えいを引き起こす例をあげる．

類似患者検索．患者の疾病情報や治療履歴, 体験談が蓄積されたデータベースを考える．このような情報の公開は一般的に望まれないが，難病などの患者やその家族にとって類似疾病を持つ患者との情報共有は有用であり，医療上の属性に基づく類似患者検索 (あるいは推薦) は情報共有の機会として有望である．しかし，患者数の少ない病気については類似検索結果から患者の特定の属性値が推定されるリスクがある．検索者が類似検索クエリを自由にデザインできる

場合にはそのリスクはさらに大きくなる．

SNS における友人検索． SNS における, 友人関係や趣味，学歴などの類似性を用いた類似友人検索を考える．類似友人検索 (あるいは友人推薦) はネットワークの充実化を図る SNS 運営者にとっても SNS ユーザにとっても有用な機能であるが, これらの個人情報を一般に公開することを望まないユーザも多い．このような秘密情報を対象とした類似検索では, 類似検索結果から友人のレコードや人間関係のうち特定の値が, 推定されるリスクがある．登録情報の少ない利用者における類似友人検索ではそのリスクはさらに大きくなる．

以上の例に見るように，メディカルレコードや SNS のプロパティなど個々のレコード自体は秘匿していても，検索結果からレコードの情報が漏えいするリスクを認識する必要がある．秘密情報を対象とした類似検索が, 個々の秘密レコードの属性値の推測を容易にすることを妨げるには, 検索による秘密漏えいの可能性を定量評価し, これを低めるような類似度ランキングを用いる必要がある．

本稿ではこのような立場から, 類似度に基づくランキング結果の開示が引き起こす情報漏洩を, 属性推定の信頼度の観点から理論的に評価する．さらに, このような属性推定を妨げるために, 差分プライバシを満足するようなランダム化手法を導入し、その効果を実験的に評価する．

1.1 関連研究

本研究ではレコードにビットベクトルを, その類似度にビットベクトル同士の内積 (common neighbor) を用いる．同様の問題設定におけるプライバシ保護の研究はいくつか知られている．

ビットベクトルで記述されるデータベースとクエリとの内積を出力する統計データベースのプライバシは [2] で議論されている．[2] ではデータベースの各レコードが 1 ビットの秘密情報であり，ビットベクトルで表されるデータベースとクエリの内積を出力としている．これは我々の設定でのデータベースの各レコードの類似度に対応し，我々の問題に包含される．しかし我々の問題意識はランキングによる情報漏えいやランダム化であり，[2] では未検証の事項である．

順序のプライバシを扱った問題として順序保存暗号 [1] が提案されている．これは，ある数値列の大小関係を保って別の値に射影することで, もとの数値の漏えいを防ぐことを目的とした暗号系である．我々の研究は, 順序の開示自体が秘密情報の漏えいを招くという問題意識に立っており立場が異なる．

差分プライバシ [3] は出力プライバシ保護のよく知られた一つの基準である．本論文で扱うような類似度や順位のような離散値や non-numeric な出力値に対して差分プライバシを実現するようなランダム化は exponential mechanism で実現できることが知られている [6]．この方法ではそれらの出力値を連続値にマップする utility function を用いる．exponential mechanism は問題領域に応じて utility function を設計する必要があり, 今回対象とする類似度ベースランキングについては, 著者らが知る限り exponential mechanism の適用例はない．

本稿の構成は以下の通りである．まず，データベースのプライバシおよびクエリに対する類似度ベースランキングを定義する (2 章)．さらに, ランキングから漏えいするプライバシを定量するためのプライバシ基準を定義する (3 章)．4 章では，2 種のランキング方法について　検索結果からの情報漏えい度合について理論的に評価する．5 章では 4 章の評価に基づいて, ランキング結果の差分プライバシを保証するランダム化メカニズムを検討する．さらに，6 章で MovieLens データにおけるランキングによるプライバシの定量的評価とランダム化による順位及び類似度の分布の変化を考察する．

2 問題設定

本章ではデータベースレコードの類似度に基づくランキングとその情報漏洩を定式化する．まずデータベースとそのプライバシを定義し，次に二種類のランキングの出力形式を定義する．最後に出力されたランキングから推測可能な属性値について，　その推定の信頼度の観点からプライバシを定義する．

2.1 データベース

データベースとして行列 $A \in \{0,1\}^{n\times\ell}$ を考える．A の i 行が一つのレコード i とし，$\mathbf{a}_i$ と書く．データベースのレコード数を $n(n>1)$, レコードの index 集合を $I=\{1,\ldots,n\}$ とする．A の各列はレコードの属性を表す．この属性の index 集合を $H=\{1,\ldots,\ell\}$ とおく．ここでは, 検索者がこのデータベースに前もって決められた種類のクエリを発行し, その応答を得るものとする.

データベースの各レコード $\mathbf{a}_i$ は個人や企業などの情報を所有する主体 i の秘密情報であり, その漏えいや推測は望ましくないものとする. すなわち， 各レコード i の持つ秘密情報は a_{ih} $(h\in H)$ である．このとき, 検索者が得たクエリ応答から各レコード i の持つ秘密情報が推測可能かどうかが問題となる.

2.2 類似度に基づくランキング

本稿で想定する検索者のクエリ $\mathbf{q}$ は各レコード $\mathbf{a}_i$ と同様に長さ ℓ のビットベクトルで，データベース A はクエリ $\mathbf{q}$ と各レコードの類似度に基づくランキング情報を返す.

ビットベクトル同士の類似度定義は複数知られるが, 本稿では類似度として共著関係などの評価に用いる common neighbor について議論を行う．文書や画像を比較するためのコサイン類似度や文書や化合物の類似度評価に用いる Jaccard 係数でも同様の考え方で議論が可能であるが，詳細な解析は今後の課題とする.

common neighbor は集合 B と C が共有する要素の数として定義される．ビットベクトル $\mathbf{a}_i$, $\mathbf{q}$ の各要素が, 指示変数としてされていると解釈すれば, $\mathbf{a}_i$ と $\mathbf{q}$ の common neighbor は

$$\mathbf{a}_i\cdot\mathbf{q}=u_i$$

となる．ただし $\mathbf{a}\cdot\mathbf{q}$ は $\mathbf{a}$ と $\mathbf{q}$ の内積を表す.

検索者からの類似レコード問い合わせに応じてデータベースは検索者に $f(A,\mathbf{q})$ を返す．ただし f は以下に説明する類似度情報を返す関数である.

$f_u:\{0,1\}^{n\times\ell}\times\{0,1\}^{\ell}\to[0,\ell]^n$ は, $\mathbf{a}_i$ と $\mathbf{q}$ を引数にとり, 各レコードのクエリの類似度 u_i のベクトル (**スコアベクトル**)

$$\mathbf{u_q}=(u_1,u_2,\ldots,u_n)$$

を返す.

次に，類似度そのものではなく, 類似度の順位を返す関数を定義する. 関数 $f_\pi:\{0,1\}^{n\times\ell}\times\{0,1\}^{\ell}\to[1,n]^n$ は, $\mathbf{a}_i$ と $\mathbf{q}$ を引数にとり, 以下に説明する順位ベクトル $\pi_\mathbf{q}$ を返す.

集合 $\{u_i|i=1,\ldots,n\}$ の重複のない要素を降順に並べたベクトルを $\mathbf{v_q}=(v_1,v_2,\ldots,v_b)$，そのサイズを $b=|\mathbf{v_q}|$ とおく．ここで $v_i>v_{i+1}$ である．これをもとに，レコードのキーから順位を参照するための**順位ベクトル**を

$$\pi_\mathbf{q}=(\pi_1,\pi_2,\ldots,\pi_n),\quad \text{where } u_i=v_{\pi_i}$$

と定義し，これを出力とする．例として以下のデータベースとクエリを考える.

$$A=\begin{pmatrix}1&0&0&0&1\\1&1&1&0&0\\0&1&1&0&0\\0&0&0&1&1\end{pmatrix},\quad \mathbf{q}=(0,1,1,1,0),$$

$\mathbf{u_q}\cdot\mathbf{a}_1=0, \mathbf{u_q}\cdot\mathbf{a}_1=2$ などから, スコアベクトルは

$$\mathbf{u_q}=(u_1,u_2,u_3,u_4)=(0,2,2,1),$$

となる．これより順位ベクトルは

$$\pi_\mathbf{q}=(3,1,1,2).$$

となる. 順位ベクトルの場合, 同じ類似度を与えるレコードは, 同順位を得るため, $b\le n$ であることに注意されたい.

本稿で扱う類似度ランキングは，レコードそのものではなく, レコードを類似度で序列化し, その index の序列をクエリ応答として公開する. この出力対象であるスコアベクトル $f_u(A,\mathbf{q})$ および順序ベクトル $f_\pi(A,\mathbf{q})$ を ranked indexes と総称し，$f(A,\mathbf{q})$ とする．本稿の主な議論の対象は順序ベクトル f_π であるが，比較のためにスコアベクトル f_u についても議論を行う.

2.3 検索者によるデータベースへの攻撃

ranked indexes からのデータベースの秘密情報漏えいについて考察する．データベースへの攻撃は，クエリ $\mathbf{q}$ への応答を得た検索者がデータベースの a_{ih} の値を推測することと定義できる．検索者による a_{ih} の推測は，検索者が得た出力を実現しうるデータベースを列挙し，これより $a_{ih}=1$ を与える確率を評価することで行われる．具体的には，$a_{ih}=1$ あるいは $a_{ih}=0$ となる確率が，1 あるいは 1 に近いほど，情報漏えいの度合いは高いといえる．この点について，次の章でより詳しく議論を行う．

3 安全性定義

クエリ $\mathbf{q}$ に応答しデータベースがスコアベクトルや順位ベクトルを公開する場合のレコード $i \in I$ の属性値の安全性基準を定義する．データベースのクエリへの応答 $f(A,\mathbf{q})$ は一意に決まる．しかし，応答 $f(A,\mathbf{q})$ を与えるデータベース A は一意と限らない．

ケース 1 (スコアベクトル). データベースがスコアベクトル $\mathbf{u_q}$ を出力するとき，$\mathbf{u_q}$ を与えうる A の集合を $S_{\mathbf{u_q}}=\{A|f(A,\mathbf{q})=\mathbf{u_q}, A\in\{0,1\}^{n\times\ell}\}$ とおく．$S_{\mathbf{u_q}}$ のうち，レコード i が属性 h を持つデータベースの部分集合を $S_{ih,\mathbf{u_q}}=\{A|A\in S_{\mathbf{u_q}}\wedge a_{ih}=1\}$ とおく．$S_{\mathbf{u_q}}$, $S_{ih,\mathbf{u_q}}$ を用いて，攻撃者は $a_{ih}=1$ について以下の確信度を得る．

$$\mathrm{Conf}(a_{ih}=1|\mathbf{q},\mathbf{u_q})=\frac{|S_{ih,\mathbf{u_q}}|}{|S_{\mathbf{u_q}}|} \qquad (1)$$

ケース 2 (順位ベクトル). データベースが順位ベクトル $\pi_{\mathbf{q}}$ を出力する場合，$\pi_{\mathbf{q}}$ をとりうる $\mathbf{u_q}$ の集合を $U_{\pi_{\mathbf{q}}}$ とおく．攻撃者は $\pi_{\mathbf{q}}$ から $a_{ih}=1$ について以下の確信度を得る．

$$\begin{aligned}\mathrm{Conf}(a_{ih}=1|\mathbf{q},\pi_{\mathbf{q}})&=\frac{|\cup_{\mathbf{u_q}\in U_{\pi_{\mathbf{q}}}} S_{ih,\mathbf{u_q}}|}{|\cup_{\mathbf{u_q}\in U_{\pi_{\mathbf{q}}}} S_{\mathbf{u_q}}|}\\&=\frac{\sum_{\mathbf{u_q}\in U_{\pi_{\mathbf{q}}}}|S_{ih,\mathbf{u_q}}|}{\sum_{\mathbf{u_q}\in U_{\pi_{\mathbf{q}}}}|S_{\mathbf{u_q}}|} \qquad (2)\end{aligned}$$

上記で得た a_{ih} の値の確信度に基づいて，ranked indexes を返すデータベースに対するプライバシを以下のように定義する．

Definition 1. *α-privacy :* データベース A について，検索者は $\mathbf{q}$ と $f(A,\mathbf{q})$ を保持している．データベースのあるレコードのある属性値 a_{ih} について，検索者の確信度が $\mathrm{Conf}(a_{ih}=1|f)\le\alpha$ $(0\le\alpha\le 1)$ であるとき，a_{ih} はクエリ $\mathbf{q}$ に対して *α-private* であるとよぶ．

定義より a_{ih} が 1-privacy，0-privacy のときそれぞれ $a_{ih}=1$，$a_{ih}=0$ と特定できる．$\alpha=1$ となる場合を**完全な漏えい**と呼ぶ．

なお，α-privacy と一見類似した概念の $\rho_1-\rho_2$ privacy[4] では，集合のランダム化パラメタとして確率空間での確信度を扱っている．本稿で扱う確信度はデータベースの属性空間での議論であり彼らとは視点が異なることに注意されたい．

α-privacy から $a_{ih}=0$ の情報漏えいも簡単に議論できるため，以降は簡単のため $a_{ih}=1$ についてのケースのみを議論する．

4 ランキングからの情報漏えい

ranked indexes を用いたデータベースへの攻撃を Definition 1 より定義する．ここで，$\|\mathbf{a}_i\|_1=a_i$，$q=\|\mathbf{q}\|_1$ (L1 ノルム，ビットベクトル中の 1 の個数)，クエリ $\mathbf{q}$ によって指定される属性集合を $H_{\mathbf{q}}=\{h|q_h=1, h\in H\}$ とおく．検索者がデータベース A へ任意の $\mathbf{q}$ を用いて類似レコードの問い合わせを行い，応答 $f(A,\mathbf{q})$ を得るとする．検索者が各 a_{ih} についてクエリ $\mathbf{q}$ とそのデータベースからの応答 $f(A,\mathbf{q})$ を用いて確信度 $\mathrm{Conf}(a_{ih}=1|\mathbf{q},f(A,\mathbf{q}))$ を得ることを検索者によるデータベース A への**単純攻撃** (naive attack) とよぶ．

本稿では，一人の検索者による一回のクエリを通じて得たデータベース出力に基づく naive attack の影響を検討する．データベースの応答がスコアベクトル $\mathbf{u_q}$ の場合と順位ベクトル $\pi_{\mathbf{q}}$ の場合について調べ，両者におけるプライバシ保護について議論を行う．

スコアベクトル $\mathbf{u_q}$ がデータベース出力の場合

スコアベクトルが出力される場合の秘密情報の漏えいに関して以下の定理が成り立つ．

Theorem 1. クエリ $\mathbf{q}$ についてスコアベクトルを応答するデータベース A を考える．このとき，

a_{ih} は $\min(1, a_i/q)$-*private* である．

これは，a_{ih} の完全な漏えいは $u_i = q$ の時おきることを示している．属性情報の完全な漏えいの可能性について，特に以下の定理が成り立つ．

Theorem 2. スコアベクトルを応答するデータベース A を考える．このとき，どの a_{ih} についても，1-*privacy* を達成するクエリ $\mathbf{q}$ が必ず存在する．

以上より，任意のクエリに対してデータベースが正しくスコアベクトルを返す場合，秘密漏えいは避けられない事がわかる．

順位ベクトル $\pi_\mathbf{q}$ がデータベース出力の場合

順位ベクトル $\pi_\mathbf{q}$ とクエリ $\mathbf{q}$ から検索者が確信できるデータベース A の情報を議論する．式 (2) より，まず $\pi_\mathbf{q}$ をとりうる $\mathbf{u_q}$ の集合 $U_{\pi_\mathbf{q}}$ を考察する．$\pi_\mathbf{q}$ を与える $\tilde{\mathbf{v}}_\mathbf{q}$ の集合 $V_{\pi_\mathbf{q}}$ とする．あるクエリ $\mathbf{q}$ と任意のビット列がとりうる類似度の集合は $\{q, q-1, \ldots, 0\}$ である．$\max_{i=1,\ldots,n}(\pi_i) = b$ なる $\pi_\mathbf{q}$ が与えられたとき，$|\tilde{\mathbf{v}}_\mathbf{q}| = b$ となるので，

$$V_{\pi_\mathbf{q}} = \{\mathbf{v_q} \mid |\mathbf{v_q}| = b, \\ v_i > v_{i+1}, v_i \in \{q, q-1, \ldots, 0\}\} \quad (3)$$

となる．ここで

$$g(\pi_i, \mathbf{q}, \pi_\mathbf{q}) = \frac{v_{\pi_i}}{q} \cdot \frac{\sum_{\mathbf{u_q} \in U_{\pi_\mathbf{q}}} |S_{\mathbf{u_q}}|}{\sum_{\mathbf{u_q} \in U_{\pi_\mathbf{q}}} |S_{\mathbf{u_q}}|}.$$

とおく．この関数を用いて完全な秘密情報漏えいを起こすクエリが常に存在することが以下の定理で示される．

Theorem 3. クエリ $\mathbf{q}$ について順位ベクトル $\pi_\mathbf{q}$ を応答するデータベースを考える．このとき a_{ih} は $g(\pi_i, \mathbf{q}, \pi_\mathbf{q})$-*privte* である．

式 (3) より，$b - \pi_i \leq v_{\pi_i} \leq q - \pi_i + 1$ となるので，以下が成り立つ．

$$g(\pi_i, \mathbf{q}, \pi_\mathbf{q}) = \frac{\cdot v_{\pi_i}}{q} \cdot \frac{\sum_{\mathbf{u_q} \in U_{\pi_\mathbf{q}}} |S_{\mathbf{u_q}}|}{\sum_{\mathbf{u_q} \in U_{\pi_\mathbf{q}}} |S_{\mathbf{u_q}}|} \leq \frac{q - \pi_i + 1}{q}.$$

等号は $b = q + 1$ のとき成り立つ．よって，a_{ih} が完全に漏えいするのは $b = q + 1$ かつ $\pi_i = 1$ のときである．

データベース A について，$\{a_{1h}, \ldots, a_{nh}\} = \{0, 1\}$ なる属性 A_h の添え字集合を

$$H_A = \{h | A, \{a_{ih}, \ldots, a_{nh}\} = \{0, 1\}, h \in H\}$$

とおく．これを用い，以下の定理によってデータベースのある列について完全な情報漏えいを起こすクエリが常に存在することが示される．

Theorem 4. クエリ $\mathbf{q}$ に対し順位ベクトル $\pi_\mathbf{q}$ を応答するデータベース A を考える．このとき，どの a_{ih} についても，1-*privacy* を達成するクエリが必ず存在する．

以上より，データベースが順位ベクトルを返す場合にも，情報漏えいが起きうる事がわかる．しかし，スコアベクトルよりも順位ベクトルの方が漏えいする属性が少ない．

5 ランキングのプライバシ保護

データベースが任意のクエリに対する ranked indexes を返す場合，1-private となる情報漏えいがおきる可能性があることが分かった．本章ではこのような漏えいを避けるため，類似度に基づく ranked indexes の差分プライバシのための応答のランダム化を導入する．

5.1 差分プライバシ

差分プライバシは，直感的にはクエリの応答から，ある入力とよく似た入力の場合との区別がつかないことを指す．ある入力を用いてクエリに応答するアルゴリズムをメカニズム $\mathcal{M}$ とする．任意の入力データベース A について，$\mathrm{nbr}(A)$ を隣接する，すなわち A と 1 要素異なり $\|A - A'\|_1 = 1$ が成り立つデータベースの集合とする．

Definition 2. ϵ-差分プライバシ：ランダム化関数 $\mathcal{M}$ が ϵ-差分プライバシを与えるとは A と任意の $A' \in \mathrm{nbr}(A)$，任意の出力 $O \subseteq Range(\mathcal{M})$ について，以下が成り立つことである：

$$Pr[\mathcal{M}(A) \in O] \leq \exp(\epsilon) \times Pr[\mathcal{M}(A') \in O].$$

出力が離散値の場合に摂動によって差分プライバシを達成する方法にexponential mechanism[6]がある．メカニズム $\mathcal{M}$ は出力に以下の方法で摂動を加える．入力 A に対し，メカニズム $\mathcal{M}$ の出力値 r と真の応答との違いを表す入力関数として $W : \{0,1\}^{n\times\ell} \times \mathrm{Range}(r) \to \mathbb{R}$ を定義する．入力した応答が真の応答に近いほど，W の出力が大きい値をとるように W を定義する．また，$\Delta W = \max_{(A,A'\in \mathrm{nbr}(A))} W(A,r) - W(A',r')$ とする．このとき exponential mechanism を以下のように定義する．

Definition 3. *exponential mechanism :* $W(A,r)$ と *base measure* $\mu(r)$ について，メカニズムを以下のように定義する：

$$\mathcal{E}^{\epsilon}_{W}(d,r) := \text{choose } r \textit{ with probability } \frac{\exp(\epsilon\cdot W(d,r))\cdot\mu(r)}{\sum_{r\in\mathrm{Range}(r)}\exp(\epsilon\cdot W(d,r))\cdot\mu(r)}$$

このメカニズムは $2\epsilon\Delta W$-差分プライバシを保証する [6]．本稿では，base measure$\mu(r)$ は一様分布とする．

5.2 差分プライバシを満たすランキング

本節ではranked indexes の exponential mechanism を用いたランダム化によって差分プライバシを達成する方法を導く．スコアベクトルの場合のランダム化メカニズムを $\mathcal{M}_u(A,\mathbf{r_q})$ とする．入力関数，ランダム化後の値域をそれぞれ

$$W_u(A,\mathbf{r_q}) = -\|\mathbf{u_q}-\mathbf{r_q}\|_1$$
$$\mathrm{Range}(\mathbf{r_q}) = \{0,\ldots,q\}$$

とする．定義より，$\Delta W_u = 1$ となる．

順位ベクトルの場合のランダム化メカニズムを $\mathcal{M}_\pi(A,\xi_\mathbf{q})$ をとする．同様に

$$W_\pi(A,\mathbf{r_q}) = -\|\pi_\mathbf{q}-\xi_\mathbf{q}\|_1$$
$$\mathrm{Range}(\xi_\mathbf{q}) = \{1,\ldots,\max(\pi_\mathbf{q})\}$$

とすれば，定義より $\Delta W_\pi = 1$ となる．ここで，順序間の距離 $\mathrm{dist}(\pi_\mathbf{q},\pi'_\mathbf{q})$ には以下の Spearman footrule を用いた．

$$\mathrm{dist}(\pi_\mathbf{q},\pi'_\mathbf{q}) = \|\pi_\mathbf{q}-\pi'_\mathbf{q}\|_1.$$

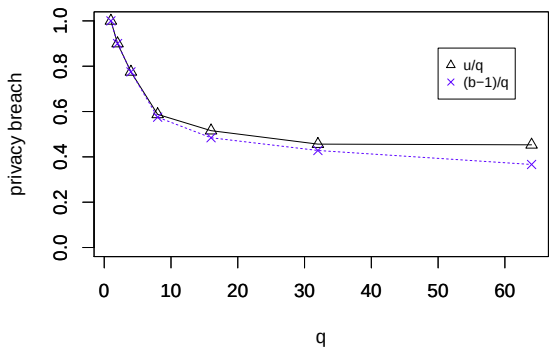

図 1: $\max_{i\in I} u_i/q$, $(b-1)/q$ の平均値

6 実データを用いた実験

Datasets. 秘密情報データベースのモデルとして，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 $A = \{0,1\}^{1000\times1700}$ を作成した．A の行はユーザすなわち本稿におけるレコード，列はムービーを示し，ユーザ i がムービー h を評価していれば $a_{ih}=1$，していなければ $a_{ih}=0$ とした．

出力による秘密情報漏洩

ランダムなクエリにより A の秘密情報の漏えい度合いを実験的に評価する．Theorem 1 より，スコアベクトルに関しては $\max_{i\in I} u_i/q$ を，Theorem 3 より，順位ベクトルに関しては $b-1/q$ を指標として用いる．これらの値が小さいほど，情報漏えいは少ない．

実験ではサイズ $q \in \{1,2,2^2,\ldots,2^6\}$ のランダムなクエリを，各 q について 20 ずつ用意した．$\max_{i\in I} u_i/q$, $(b-1)/q$ の各 q における平均値は図 1 のようになった．これより，クエリが含む 1 の数が大きいほど情報漏えいの平均的リスクは低くなることが考察される．

ランダム化による出力の変化

出力値のランダム化による変化を考察するため，ランダム化しない出力とランダム化した出力を比較した．ランダム化メカニズムのパラメタは $\epsilon = \log 2$ とした．ランダム化しない出力におけるクエリに類似したレコードを true, 類似でないレコードを false，ランダム化した出力のクエリに類似したレコードを positive として

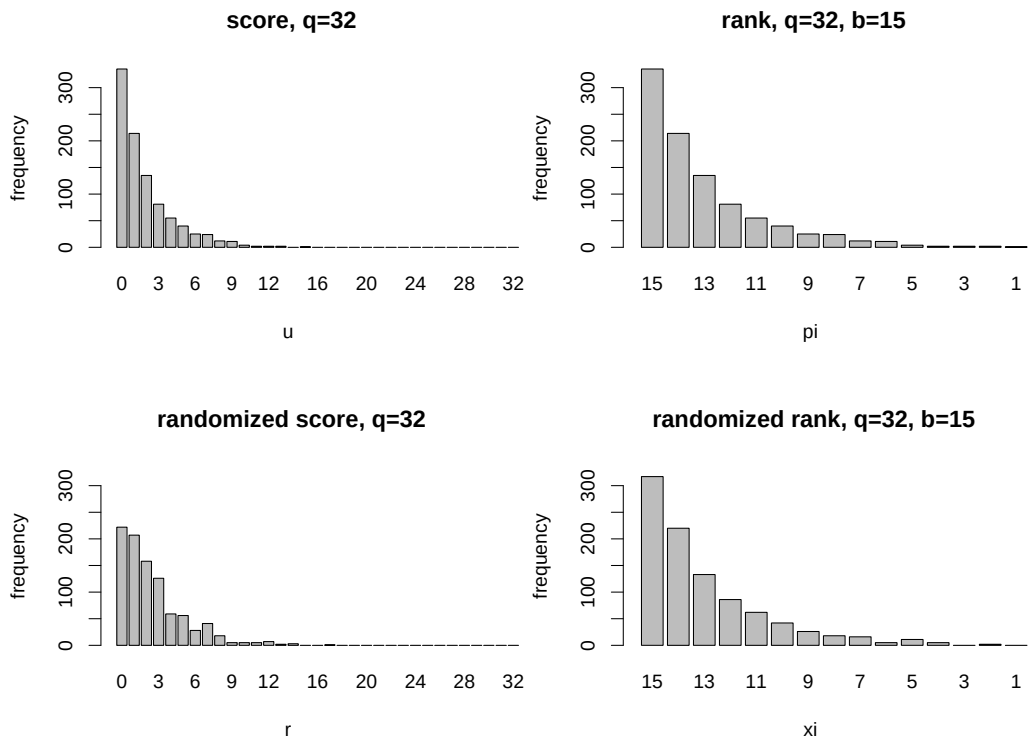

図 2: スコアベクトル，順位ベクトル及びランダム化したそれらの値の分布例

f	q	# of true	# of TP	# of FP
$\mathbf{u_q}$	4	8.35	2.7	72.5
$\mathbf{u_q}$	64	8.35	1.65	86.8
$\pi_\mathbf{q}$	4	7.35	5.55	1.25
$\pi_\mathbf{q}$	64	7.35	5.0	1.30

表 1: あるサイズのクエリについての true positive と false positive 数の 20 回平均

true positive と false positive を比較した．

$q = 4$ の場合は $u_i = \max(\mathbf{u_q})$, $\pi_i = 1$ ならば true，$q = 64$ の場合は $u_i \leq \max(\mathbf{u_q}) - 5$, $\pi_i \leq 5$ ならば true とした．ランダム化したものについても同様である．$q = 4, 64$ に関する出力の true positive, false positive の 20 回平均は表 1 のようになる．この結果では q が小さい方が true positive が小さく，false positive が大きくなることが示されている．また，$\mathbf{u_q}$ よりも $\pi_\mathbf{q}$ の方が ture positive を多く，false positive を少なく持ちランダム化前後で結果が保存されている．

また，あるクエリの結果について結果を直接出力する場合と $\epsilon = \log 2$ としたランダム化を行った場合の $\{u_i\}$, $\{\pi_i\}$ の分布の変化は図 2 のようになった．$\{u_i\}$ の場合，スコアが大きくなるにつれレコード数が少なくなること，ランダム化を行った場合スコアの大きなレコード数が増加する傾向にあることが見て取れる．$\{\pi_i\}$ についても同様の傾向がみられる．よって，ランダム化は分布の片よりの影響を受け，もともと頻度の少ない値が相対的に大きな変化を受ける傾向があると考察される．

7 終わりに

本稿では，データベースが検索者からの類似検索クエリに応答して, 各レコードとクエリとの類似度順の順位を出力するモデルを定義し，そのプライバシについて定式化および, ランダム化方法の導入，実データを用いた実験を行った．また我々の提案したプライバシ保護基準において，類似度ベクトルおよび順位ベクトルを出力するケースともに, 完全に漏えいする可能性が常に存在することが示された．

これを考慮し, 類似度ベクトルおよび順位ベクトルの出力における差分プライバシを達成す

るランダム化メカニズムを導入した．実データを用いた漏えいリスクの検証及びランダム化の出力の変化からは，漏えいが起きるケースはクエリのサイズ (クエリに含まれる 1 の数) が大きいほど少なくなることが考察された．また，順位ベクトルを出力する方が類似度ベクトルの出力に比べ漏えいリスクおよびランダム化の影響が少ないことが考察された．今後の課題として，漏えいリスクの定量化から得られた知見を活かし，出力の有用性をより保ったランダム化の工夫があげられる．また，今回は検索者は単一のクエリを発行することを想定していたが、現実的には検索者はインタラクティブに複数クエリを発行することが想定され、そのリスクの評価とこれにたいする安全性の向上が課題である．

謝辞

本研究は JST さきがけ「知の創成と情報社会」および部分的に最先端研究開発プログラム (FIRST)「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」の助成を受けたものである．

参考文献

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 563–574. ACM, 2004.

[2] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210. ACM, 2003.

[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] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222. ACM, 2003.

[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] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.

A Proof

Proof of Theorem 1. $u_i = \sum_{h \in H_{\mathbf{q}}} a_{ih}$ より，

$$|S_{\mathbf{u_q}}| = {}_q\mathrm{C}_{u_i} \cdot 2^{\ell-q} \cdot \prod_{j \neq i} \left({}_q\mathrm{C}_{u_j} \cdot 2^{\ell-q}\right) \quad (4)$$

$$|S_{ih,\mathbf{u_q}}| = {}_{q-1}\mathrm{C}_{u_i-1} \cdot 2^{\ell-q} \cdot \prod_{j \neq i} \left({}_q\mathrm{C}_{u_j} \cdot 2^{\ell-q}\right). \quad (5)$$

式 (1) より全ての a_{ih} について，

$$\mathrm{Conf}(a_{ih} = 1|\mathbf{u_q}) = \frac{{}_{q-1}\mathrm{C}_{u_i-1}}{{}_q\mathrm{C}_{u_i}} = \frac{u_i}{q} \quad (6)$$

となる．レコード $\mathbf{a}_i$ と $\mathbf{q}$ の類似度 u_i について，$u_i = \mathbf{a}_i \cdot \mathbf{q}$ より $u_i \leq \min(q, a_i)$ であるので $a_{ih}(i \in I, h \in H_{\mathbf{q}})$ は $\min(1, a_i/q)$-private となる． □

Proof of Theorem 2. クエリ $\mathbf{q}$ が $q_h = 1$, $q_k = 0 (k \neq h)$ とすると $u_i = 1$ である．式 (6) より $\mathrm{Conf}(a_{ih} = 1|\mathbf{u_q}) = \frac{u_i}{q}$ なので $a_{ih}(i \in I, h \in H, a_{ih} = 1)$ が 1-private となるクエリが必ず存在する． □

Proof of Theorem 3. 定義より $U_{\pi_{\mathbf{q}}}$ と $V_{\pi_{\mathbf{q}}}$ は一意に対応する．式 (2)，(4)，(5) より

$$\begin{aligned}\mathrm{Conf}(a_{ih} = 1|\mathbf{q}, \pi_{\mathbf{q}}) &= \frac{\sum_{\mathbf{u_q} \in U_{\pi_{\mathbf{q}}}} |S_{ih,\mathbf{u_q}}|}{\sum_{\mathbf{u_q} \in U_{\pi_{\mathbf{q}}}} |S_{\mathbf{u_q}}|} \\ &= \frac{\sum_{\mathbf{u_q} \in U_{\pi_{\mathbf{q}}}} |S_{\mathbf{u_q}}| \cdot \tilde{v}_{\pi_i}/q}{\sum_{\mathbf{u_q} \in U_{\pi_{\mathbf{q}}}} |S_{\mathbf{u_q}}|}.\end{aligned}$$

$$g(\pi_i, \mathbf{q}, \pi_{\mathbf{q}}) = \frac{1}{q} \cdot \frac{\sum_{\mathbf{u_q} \in U_{\pi_{\mathbf{q}}}} |S_{\mathbf{u_q}}| \cdot v_{\pi_i}}{\sum_{\mathbf{u_q} \in U_{\pi_{\mathbf{q}}}} |S_{\mathbf{u_q}}|}$$

であったので，$a_{ih}(i \in I, h \in H, a_{ih} = 1)$ は $g(\pi_i, \mathbf{q}, \pi_{\mathbf{q}})$-private である． □

Proof of Theorem 4. $h \in H_A$, $q_h = 1, q_j = 0$ $(h \in H_A,\ j \neq h)$ を満たす $\mathbf{q}$ を考える．$v_{\mathbf{q}} = (1, 0)$，$\pi_i \in \{1, 2\}$，$u_i = 2 - \pi_i$ が成り立つので式 (6) から $\pi_i = 1$ ならば

$$g(1, \mathbf{q}, \pi_{\mathbf{q}}) = u_i/q = 1. \quad (7)$$

となる．よって $a_{ih}(i \in I, h \in H_A, a_{ih} = 1)$ には 1-private となるクエリが必ず存在する． □