DEIM2022 J31-4(day3 p36)

属性間依存度を考慮したデータベースの安全なフラグメント化

磯田　飛鳥†　　戸田　貴久†

† 電気通信大学　大学院情報理工学研究科　情報・ネットワーク工学専攻
〒 182–8585 東京都調布市調布ヶ丘 1-5-1
E-mail: †{isoda,toda}@disc.lab.uec.ac.jp

あらまし　PPDP とは，パーソナルデータを匿名化し第三者に公開する技術である．その 1 つのフラグメント化は，属性間の関係を秘匿するために，データベースを属性で複数に分離してフラグメントを作る．先行研究では，属性間の依存関係に関する背景知識を用いて，そのような依存関係からフラグメントの連結を回避するフラグメント化の方法を与えている．しかし，属性間の依存関係は所与のものとして扱われており，これを求める方法は与えられていない．さらに，依存関係の有無を扱えるだけであり，依存の強度を扱うことができない．そこで本研究では，統計的な尺度で依存度を評価し，その依存度に応じてフラグメント連結を阻止する制約にコストを与え，フラグメント化を最適化問題として定式化した．実験では，提案手法と依存を考慮しないものとの比較で，フラグメントの連結を回避できたことを確認した．

キーワード　PPDP，データベース匿名化，プライバシー，データマイニング，制約最適化問題，T-score

1　はじめに

PPDP(Privacy Preserving Data Publishing) とは，パーソナルデータをプライバシー侵害が起きないよう処理し第三者に公開する技術である [1]．PPDP の 1 つに，フラグメント化がある．これは，属性間の秘密にしたい関係を分離させる，つまりデータベースを属性によっていくつかのフラグメントに分離する技法である [2]．

先行研究 [3] では異なる属性間に依存があることによる問題が指摘されている．この問題とは，属性間の依存によって一方の属性値からもう一方の属性値を推測されてしまい，そこから異なるフラグメント同士が連結されてしまうことである．よって，属性間の秘密にしたい関係を結びつけられてしまう．解決手法として，属性間の依存を制約として与えることで，フラグメントの連結が起きないようなフラグメント化を求める問題を制約充足問題（CSP）として解いた．

この手法では，属性間の依存関係は所与のものとして扱われており，依存関係を求める方法については与えられていない．そのため，問題点として，1) 属性間の依存があるか判断するために扱うデータに関する知識が必要であり，また，2) 依存性の有無を扱えるだけであり，依存の強度を扱うことが出来なかった．

本論文は，統計的な尺度で依存度を評価し，依存を考慮したフラグメント化を行う手法を提案する．この手法では，属性間の依存の強さを統計的に解析し，その程度に応じてフラグメントの連結が起きないようにする制約にコストを与える．そして，フラグメント化をコストの最小化問題として捉えることで，連結の可能性が最小のフラグメントを求める．実験では，依存を考慮しないものと提案手法とを比較することで，依存による連結を回避するフラグメントが求められることを確認した．

本論分は次のように構成される．2 節では，属性間依存を考慮したフラグメント化の説明とこの手法の問題点を述べる．3 節では，提案手法で使用する基本概念について説明する．4 節では，属性依存度を考慮したフラグメント化の枠組みを与える．5 節では，属性依存度を考慮したフラグメント化を求めるための手法を与える．6 節では，提案手法の実験とその分析を行う．7 節では，本論文のまとめと今後の研究課題を述べる．

2　先行研究

本節では，提案手法に直接関係のある 2 つの先行研究を紹介する．また，本稿を通して使う用語や記法も導入する．

2.1　データベースのフラグメント化

表 1　病　　院

氏名	性別	郵便番号	誕生日	病気	治療法
A	男	182-8585	11/9	風邪	投薬
B	男	182-8585	2/27	癌	手術
C	女	450-0002	2/27	骨折	ギブス
D	女	182-8585	11/9	風邪	投薬
E	女	450-0002	11/9	風邪	投薬
F	男	450-0002	11/9	骨折	ギブス

表 2　フラグメント化

性別	病気
男	風邪
男	癌
女	骨折
女	風邪
女	風邪
男	骨折

郵便番号	誕生日	治療法
182-8585	11/9	投薬
182-8585	2/27	手術
450-0002	2/27	ギブス
182-8585	11/9	投薬
450-0002	11/9	投薬
450-0002	11/9	ギブス

表 3　機密性制約と可視性制約の一覧

機密性制約：　$c_1 =$ { 氏名 }
　　$c_2 =$ { 郵便番号,誕生日,病気 }

可視性制約：　$v_1 =$ 治療法
　　$v_2 =$ 病気 $\wedge$ 性別
　　$v_3 =$ 郵便番号 $\wedge$ (性別 $\vee$ 誕生日)

データベースを公開する際には，プライバシー保護のために秘密にしなければならない情報と，公開情報の利活用のために開示されることが求められる情報がある．データベースの**フラグメント化**とは，秘密にしたい情報に関する制約（**機密性制約**）と開示したい情報に関する制約（**可視性制約**）が指定されるとき，どちらの制約も満たすように元のデータベースを複数の部分（**フラグメント**）に分割することを意味する．以降では，リレーショナルデータベースの用語を使って，このフラグメント化の一般的な定義を与える．詳細は [2] を参照されたい．

本稿を通して，匿名化の対象となるリレーションスキーマを任意に 1 つ固定して，$R(a_1, \ldots, a_n)$ と表すことにする．ここで R はリレーションスキーマの名称であり，$a_1, \ldots, a_n$ は属性である．各属性 a_i $(i = 1, \ldots, n)$ のドメインを $D(a_i)$ と表す．このときフラグメントは次のように定義される．

定義 1. リレーションスキーマ R の**フラグメント**（*fragment*）とは，R の属性集合 $\{a_1, \ldots, a_n\}$ の部分集合である．

例えば，表 1 の 6 個の属性に対して，{ 性別,病気 } は 1 つのフラグメントである．表 2 の左は，表 1 のリレーションをこのフラグメントにしたがって制限したものである．同様に，別のフラグメント { 郵便番号,誕生日,治療法 } に対しては，表 2 の右のリレーションが得られる．

このように，フラグメントの集まりを求めることを**フラグメント化**（*fragmentation*）と呼ぶ．後で述べるように，一般には所与の制約の下でフラグメント化を行う．フラグメント化の結果として得られるフラグメントの集まりもフラグメント化と呼ぶことがある．

機密性制約は，秘密にしたい属性の組合せを表現するものとして次のように定義する．

定義 2. リレーションスキーマ R 上の**機密性制約**（*confidentiality constraint*）とは，R の属性集合 $\{a_1, \ldots, a_n\}$ の部分集合である．

機密性制約が唯１つの属性からなるときは，その属性を公開したくないということを意味し，どのフラグメントにも含まれないようにする．機密性制約が複数の属性からなるときは，それらの属性の組合せが同時に現れるフラグメントが公開されたくないということを意味し，どのフラグメントにもその属性の組合せが含まれないようにする．

例えば，表 3 の機密性制約 $c_1 =$ { 氏名 } を見てみよう．氏名は単体でも個人を特定できる可能性が高いため，どのフラグメントにも氏名が含まれないことが要請されていると考えられる．一方，$c_2 =$ { 郵便番号,誕生日,病気 } については，どの属性も単体で個人を特定できそうにないが，組合せにより個人が推測される可能性が高いため，この組合せが排除されていると考えられる．

可視性制約は，公開したい属性の組合せを表現するものとして次のように定義する．

定義 3. リレーションスキーマ R 上の**可視性制約**（*visivility constraint*）は，R の属性集合 $\{a_1, \ldots, a_n\}$ 上の単調な命題論理式 [1]である．

ここで，各属性 a_i はその属性がフラグメントに出現するか否かを表す論理変数としてみなされている．フラグメント F が与えられるとき，各属性 a_i について $a_i \in F$ のとき a_i に真を割り当て，そうでないとき偽を割り当てることで，論理変数への真偽値割り当て π_F が定まり，可視性制約として指定された論理式の真偽値が定まることに注意されたい．以降では，真偽値割り当て π_F の下で論理式 ϕ が真になることを $\pi_F \models \phi$ と表す．

命題論理式の単調性は以下のように，フラグメントの包含関係に関する単調性として解釈することができる．$\pi_F \models \phi$ ならば，F を包含する任意の F' に関して $\pi_{F'} \models \phi$ となる．直感的には，F に含まれる属性の組合せによって可視性制約が満たされるならば，他の属性を F に加えても可視性制約が満たされる．

機密性制約と可視性制約を満たすフラグメント化を求めることを目的とする場合には，正しいフラグメント化は次のように定義される．

定義 4. リレーションスキーマ $R(a_1, \ldots, a_n)$，機密性制約の集合 $\mathcal{C}$，可視性制約の集合 $\mathcal{V}$ が与えられるとき，R のフラグメント化 $\mathcal{F}$ が $\mathcal{C}$ と $\mathcal{V}$ に関して**正しい**（*correct*）とは次のすべての条件が満たされるときである．

（1）　$\forall c \in \mathcal{C}, \forall F \in \mathcal{F}.\ c \nsubseteq F$（機密性）

（2）　$\forall v \in \mathcal{V}, \exists F \in \mathcal{F}.\ \pi_F \models v$（可視性）

（3）　$\forall F_i, F_j \in \mathcal{F}.\ F_i \neq F_j \implies F_i \cap F_j = \emptyset$（連結不可能性）

最小サイズの正しいフラグメント化を求める問題は Min-CF と呼ばれており，一般に NP 困難であることが知られている [2].

2.2　データ依存関係

データ依存関係は，ある属性 a の値から別の属性 b の値が推測されてしまうような属性間の関係 $a \rightsquigarrow b$ を意味する．値を正確に決定できるか，あるいは，何らかの不確実性のモデルの下で高確率で値を推定できるような状況を表している．この関係は属性集合 X, Y の間の依存関係 $X \rightsquigarrow Y$ に拡張することもできる．形式的な定義は次の通りである．

定義 5. リレーションスキーマ R 上の**データ依存関係**（*data dependency*）とは，R の互いに素な部分集合 X, Y に対して $X \rightsquigarrow Y$ として表現される式である．$X = \{a\}, Y = \{b\}$ のときは $a \rightsquigarrow b$ と書くこともある．

1：否定記号なしで論理積と論理和だけで組み立てられる命題論理式に相当する．

例として，表 2 を観察してみよう．患者に施された治療法は，その患者が患っている病気の情報までも伝えてしまう．実際，治療法としてギブスを付けたら，その人は骨折をしたのだとわかり，手術をうけていたら，癌などの重い病気を患っていることがわかる．このような時，データ依存関係 治療法 $\rightsquigarrow$ 病気 があると考える．

このような属性依存があると，フラグメントが連結されてしまう危険性が生じる．表 2 では，治療法属性の値で手術は 1 個のレコードだけであり，病気属性の値で手術を受けるほど重い病気は癌だけである．よって，異なるフラグメントに存在するレコードが連結されてしまう．

そこで，データ依存関係も考慮に入れたフラグメント化を考える枠組みが与えられている [3]．以下の定義は，データ依存関係 $X \rightsquigarrow Y$ が $|X| = |Y| = 1$ の場合で，かつ反射律と対称律を満たす場合に特殊化した形で与えている [2]．

定義 6. リレーションスキーマ $R(a_1, \ldots, a_n)$，機密性制約の集合 $\mathcal{C}$，可視性制約の集合 $\mathcal{V}$，データ依存関係の集合 $\mathcal{D}$ が与えられるとき，R のフラグメント化 $\mathcal{F}$ が $\mathcal{C}, \mathcal{V}, \mathcal{D}$ に関して**正しい**（*correct*）とは次のすべての条件が満たされるときである．

（1） $\forall c \in \mathcal{C}, \forall F \in \mathcal{F}.\ c \nsubseteq F$（機密性）

（2） $\forall v \in \mathcal{V}, \exists F \in \mathcal{F}.\ \pi_F \models v$（可視性）

（3） $\forall F_i, F_j \in \mathcal{F}, \forall a \rightsquigarrow b \in \mathcal{D}.\ F_i \neq F_j \implies (a \in F_i \implies b \notin F_j)$（連結不可能性）

依存を考慮したフラグメント化の例を表 4 に挙げる．ここで，フラグメント前の元のデータベースは表 1 で与えられるものとし，$\mathcal{C}$ と $\mathcal{V}$ は表 3 で与えられるものとする．データ依存関係は次に挙げるものからなる：病気 $\rightsquigarrow$ 治療法，治療法 $\rightsquigarrow$ 病気，各属性 a_i に対して $a_i \rightsquigarrow a_i$．データ依存関係が反射律を満たすことは，相異なるフラグメントが互いに素になることに等しいので，データ依存関係を考慮しないフラグメント化の定義の拡張になっている．

表 4　依存を考慮したフラグメント化

性別	病気	治療法
男	風邪	投薬
男	癌	手術
女	骨折	ギブス
女	風邪	投薬
女	風邪	投薬
男	骨折	ギブス

郵便番号	誕生日
182-8585	11/9
182-8585	2/27
450-0002	2/27
182-8585	11/9
450-0002	11/9
450-0002	11/9

2.3 問 題 点

データ依存関係を扱う枠組み [3] では，属性の依存関係は所与のものとして扱われており，依存関係を求める方法については与えられていなかった．実際，計算機実験では，ランダムに生成されたデータ依存関係（および機密性制約，可視性制約）が用いられた．属性の個数が多い場合やデータに関する背景知識に精通していない場合，人手で依存関係を与えることには限界がある．このことに関して 2 つの問題点を以下で順に述べる．

1 つ目の問題点は，属性の依存関係を求めるために，扱われるデータに関する背景知識が必要なことである．例えば，郵便番号と病気は一見あまり依存関係が見られないかもしれないが，風土病が顕著に現れるデータベースなどにおいては強い依存関係が現れるだろう．このことは，先行研究でなされたようにリレーションスキーマに対してフラグメント化を考えるのでは不十分であり，個別のリレーションに関する背景知識が必要であると考えられる．しかし，そのような背景知識を人手で与えるのは限界がある．ある業界ではよく知られていることであっても，業界に精通していないデータ加工者には思いもよらない属性依存もありうる．

2 つ目の問題点は，先行研究では依存関係の有無を扱うだけであり，依存の強度を扱えないことである．例えば，性別によって喫煙率には差があることが知られており，男性が 27．1%，女性が 7．6%というデータがある [4]．よって，性別が男性であるならば，喫煙の習慣があることが相対的に高く期待されるので，性別と喫煙習慣の間には一定程度の依存関係が見られるといえる．ただ，他の属性の間により強い（あるいはより弱い）別の依存関係がありうるので，それらを依存関係の有無だけで捉えるのでは，現実をうまく反映したものにならないだろう．

3 準　　備

本節では，提案手法の中で使われる基本概念を説明する．

3.1 T　　値

T 値（*T-score*）とは，コーパス研究でつながりのある単語 (例: according to， bad taste など) を見つける際によく使われる指標である [5]．T 値は t 検定をもとに設計されており，式 (1) で算出される統計値を使って単語 x， y の依存関係を判定する．すなわち，有意水準 p において t 値が t 分布の数値表から求めた値以上であれば，帰無仮説 $P(x, y) = P(x) \cdot P(y)$ が棄却され，x と y の出現には依存関係があると判定する．

$$\text{T-score}(x, y) = \frac{n_{xy} - \frac{n_x \times n_y}{n}}{\sqrt{n_{xy}}} \tag{1}$$

ここで，n_{xy} は x と y が同時に出現した回数，n_x は x が出現した回数，n_y は y が出現した回数，n はコーパスの総語数を表す．

3.2 制約最適化問題

制約充足問題（*CSP*）とは与えられたすべての制約を満たす解を求める問題であり，形式的には組 (X, D, C) として定義される．ここで，

（1） X は変数の集合である．

（2） D は各変数 $v \in X$ に対してその取りうる値の有限集合（v の**ドメイン**）を割り当てる関数である．

（3） C は X 上の関係（**制約**）の集まりである．

CSP の**解**とは，各変数への値の割り当てのうち，すべての制約を満たすものを意味する．

2：本稿で取り組む研究では，最初のステップとして先行研究のデータ依存関係の特別な場合に関して枠組みを拡張した．一般の場合は今後の課題である．

制約最適化問題（*COP*）は，目的関数 f を伴う制約充足問題 (X, D, C) である．ここで，f は (X, D, C) の解に対して数値を割り当てる関数である．制約最適化問題の目的は，f を最小化（あるいは最大化）する (X, D, C) の解を求めることである．

制約充足問題や制約最適化問題には各種のソルバーが一般に公開されている．そのようなソルバーを利用するためには，制約充足問題（や制約最適化問題）としての定式化を与え，ソルバーの入力として与えて，ソルバーを動作させることにより，問題の解を求めることができる．

4 属性依存度を考慮したフラグメント化

本節では，先行研究の問題点を解消するために，属性依存度を考慮したフラグメント化の枠組みを与える．

属性依存の強度は，形式的には次のように定義する．

定義 7. リレーションの**属性依存度**とは，各属性 a, b の間に対して実数を割り当てる関数であり，δ で表記する．

$\delta(a, b)$ は a から b への依存関係の強度を表す．属性依存度はリレーションスキーマ R のインスタンスごとに定まるものと考える．2 節で述べたように本稿では問題を単純にするために，対称的な依存関係だけを扱うので，強度も方向を持たないと考え，$\delta(a, b) = \delta(b, a)$ と仮定する．また，次節で与える属性依存度の計算方法ではリレーションスキーマ R の属性はすべてカテゴリ属性と仮定している．

属性依存度を考慮したフラグメント化は，先行研究の定義 6 の正しいフラグメント化のうち，指定されたフラグメントの個数をもち，最小のコストを持つものを求める問題である．

定義 8. リレーションスキーマ R，機密性制約の集合 $\mathcal{C}$，可視性制約の集合 $\mathcal{V}$，依存関係の集合 $\mathcal{D}$，フラグメントの個数 m，R のフラグメント化に対してコストを割り当てる関数 f が与えられるとする．ただし，フラグメント化が $\mathcal{C}, \mathcal{V}, \mathcal{D}$ に関して正しくないか，あるいは，フラグメントの個数が m でないとき，f は無限大のコストを持つ．このとき，**属性依存度を考慮したフラグメント化**とは，f を最小化する R のフラグメント化を求める問題である．

この問題は形式的にはリレーションスキーマに対して定義されているが，本稿では特定のリレーションから定まる属性依存度を用いて，問題の目的関数を設計する．

5 提 案 手 法

本節では，属性依存度を考慮したフラグメント化を求めるための手法を与える．各小節において提案手法の詳細を述べる前に，以下に概要をまとめる．

まず，提案手法では，属性依存の存在とその強度とを公開したいデータベースから統計的な指標に基づいて算出する．本稿では，属性依存度が一定の値 α 以上であれば，明白な依存関係が認められると考え，先行研究で与えられた枠組みと同様にその依存関係が同じフラグメントの中に現れないようする．したがって，そのような場合には依存度の数値の違いによって結果が変わることがない．一方，α 未満の属性依存度を持つ場合には，依存度の値が大きければ大きいほど，(他の属性ペアに比べて）より優先的に依存度を解消するようなフラグメント化を求めるようにしたい．

そこで，属性依存によるフラグメントの連結を阻止する制約を導入して，その制約が満たされないときに属性依存度がコストとして加算されるように目的関数を設計する．この種の制約は必ず満たされることが要求される制約（**ハード制約**）ではなく，制約が満たされない場合に割り当てられたコストが発生する制約（**ソフト制約**）であることに注意されたい．このようにして定式化される制約最適化問題をソルバーで解くことで，依存によるフラグメントの連結の危険性が最小になるフラグメントの集合を求めることができる．

以降の小節では，最初に，与えられたリレーションから属性依存度を算出する方法を与える．そして，属性依存度の値に応じて，属性間の依存関係を分類する方法を与える．次に，これらの情報に基づいて，目的関数を設計する．最後に，属性依存度を考慮したフラグメント化を制約最適化問題として定式化する方法を与える．

5.1 属性依存度の算出と分類

属性依存度は，与えられたリレーションから T 値を求めることで算出する．

具体的には，各属性 a_i, $a_j (1 \leqq i < j \leqq n)$ と，これらの属性のとるすべての値の対 $(x, y) \in D(a_i) \times D(a_j)$ に対して T-score(x, y) を計算する．ここで，式 (1) において n_{xy} は属性値 x, y が同時に含まれるレコードの個数，n_x と n_y はそれぞれ x と y が含まれるレコードの個数，n はレコードの総数を表す．そして，属性 a_i と a_j の間の依存度は

$$\delta(a_i, a_j) := \max_{x \in D(a_i), y \in D(a_j)} \text{T-score}(x, y) \tag{2}$$

により定める．

次に，属性の対 (a_i, a_j) $(1 \leqq i < j \leqq n)$ を次の基準で分類する（α,β は定数）．

強い依存: $\alpha \leqq \delta(a_i, a_j)$
弱い依存: $\beta \leqq \delta(a_i, a_j) < \alpha$
依存なし: $\delta(a_i, a_j) < \beta$

ここで，強い依存の場合は必ず依存関係を解消するようなフラグメント化を行うことにする．弱い依存の場合は依存度の値 $\delta(a_i, a_j)$ に応じて依存関係の解消を行うようにする．依存なしの場合は依存関係の解消の対象に含めない．この目的に合致するように定数 α, β は適切な値にあらかじめ設定されているものとする．

以降では，a_i と a_j が強い依存に分類されるとき**強い依存関係がある**と呼び，2 節と同じ記法 $a_i \rightsquigarrow a_j$ で表す．一方，a_i と a_j が弱い依存に分類されるとき**弱い依存関係がある**と呼び，$a_i \dashrightarrow a_j$ と表して強い依存と区別することにする．

ちなみに，各属性 a_i に関して $\delta(a_i, a_i) = \alpha$ と定めることにする．これにより反射律 $a_i \rightsquigarrow a_i$ が満たされる．式 (2) より，

対象律も満たされることは明らかである．

5.2 目的関数の設計

属性依存度を考慮したフラグメント化における目的関数 f を定める．

本稿では，依存関係の場合と同様，**強い可視性制約**と**弱い可視性制約**の 2 種類を考える．強い可視性制約の集合と強い依存関係の集合はそれぞれ $\mathcal{V}$ と $\mathcal{D}$ で表し，弱い可視性制約の集合と弱い依存関係の集合はそれぞれ $\mathcal{V}^*$ と $\mathcal{D}^*$ で表す．

フラグメント化 $\mathcal{F}$ のコスト $f(\mathcal{F})$ は，弱い可視性制約 $v \in \mathcal{V}^*$ と弱い依存関係 $a \dashrightarrow b \in \mathcal{D}^*$ のうち，それぞれ式 (3) と (4) を満たすもののコストの総和である．

$$\forall F \in \mathcal{F}.\ \pi_F \not\models v \tag{3}$$

$$\exists F_i, F_j \in \mathcal{F}.\ (F_i \neq F_j) \wedge (a \in F_i) \wedge (b \in F_j) \tag{4}$$

ここで，弱い依存関係 $a \dashrightarrow b$ のコストは $\delta(a,b)$ である．一方，弱い可視性制約に対するコストは人手で与えるものとする．

5.3 制約最適化問題としての定式化

属性依存度を考慮したフラグメント化を制約最適化問題 (X, Dom, C, f) として定式化する．

変数集合とドメイン：X の変数は 3 種類ある．1 つ目の変数は f_i の形で表され，$i \in \{1,\ldots,n\}$ とする．ここで n は属性の個数である．f_i のドメインは $Dom(f_i) = \{0,1,\ldots,m\}$ となる．ここで m はフラグメントの個数である．$f_i = 0$ のとき属性 a_i はどのフラグメントにも含まれないことを表し，$f_i > 0$ のとき a_i の属するフラグメントの番号を表す．

2 つ目の変数は x_i の形で表され，$i = \{1,\ldots,k\}$ とする．$Dom(x_i) = \{0,1\}$ である．ここで，$k = |\mathcal{V}^*|$ とし，$\mathcal{V}^* = \{v_1,\ldots,v_k\}$ で表されるとする．$x_i = 1$ のとき，そしてそのときに限り，v_i に関して式 (3) が成立することを意味する．この意味を正しく反映するように後で適切な制約を与える．

3 つ目の変数は y_i の形で表され，$i = \{1,\ldots,l\}$ とする．$Dom(y_i) = \{0,1\}$ である．ここで，$l = |\mathcal{D}^*|$ とし，$\mathcal{D}^* = \{d_1,\ldots,d_l\}$ で表されるとする．$y_i = 1$ のとき，そしてそのときに限り，d_i に関して式 (4) が成立することを意味する．この場合も，後で適切な制約を与える．

目的関数： 5.2 節で設計した目的関数 f は次のように表される．

$$f(X) = c_1x_1 + \cdots + c_kx_k + e_1y_1 + \cdots + e_ly_l$$

ここで，c_i は v_i のコストを表す定数，e_i は d_i のコストを表す定数である．

制約集合：機密性制約は属性番号の集合として表されているとするとき，各機密性制約 $C \in \mathcal{C}$ に対して次の制約を定める．

$$\left(\bigvee_{i,\ j\in C,\ i\neq j} f_i \neq f_j\right) \vee \left(\bigvee_{i\in C} f_i = 0\right)$$

この制約は，C に現れる属性の組合せは同じフラグメントに含まれないことを表す（定義 6 の 1 番目の条件）．

強い可視性制約 $v \in \mathcal{V}$ は属性集合 $\{a_1,\ldots,a_n\}$ 上の単調な論理式として与えられるのであった．よって，i 番目のフラグメント F_i に対して $\pi_{F_i} \models v$ が満たされることは，v における各論理変数 a_j を $(f_j = i)$ で置き換えることで，v から得られる論理式 $v[(f_1 = i)/a_1,\ldots,(f_n = i)/a_n]$ により表現される．したがって，$v \in \mathcal{V}$ に関する定義 6 の 2 番目の条件は次で与えられる．

$$\bigvee_{i=1,\ldots,m} v[(f_1 = i)/a_1,\ldots,(f_n = i)/a_n]$$

弱い可視性制約も同様の考え方で制約を導入する．ここで，$\mathcal{V}^* = \{v_1,\ldots,v_k\}$ と表されることを思い出されたい．各 $v_i \in \mathcal{V}^*$ に対して，v_i に関する式 (3) が成り立つとき，そしてそのときに限り，$x_i = 1$ となるのであった．このことを表す制約は次で与えられる．

$$x_i = 1 \iff \bigwedge_{i=1,\ldots,m} \neg v[(f_1 = i)/a_1,\ldots,(f_n = i)/a_n]$$

強い依存関係 $a_i \rightsquigarrow a_j \in \mathcal{D}$ に対しては，次の制約を導入する．

$$\bigwedge_{1\leqq s<t\leqq m} (\ (f_i = s) \implies (f_j \neq t)\ )$$

これは定義 6 の 3 番目の条件に対応する．

弱い依存関係に対しては以下のように定める．まず，$\mathcal{D}^* = \{d_1,\ldots,d_l\}$ と表されることを思い出されたい．各 $d_i = a_u \dashrightarrow a_v \in \mathcal{D}^*$ に対して，d_i に関する式 (4) が成り立つとき，そしてそのときに限り，$y_i = 1$ となるのであった．このことを表す制約は次で与えられる．

$$y_i = 1 \iff \bigvee_{1\leqq s<t\leqq m} (\ (f_u = s) \wedge (f_v = t)\ )$$

6 実　　験

現実のデータベースを用いて，提案手法でフラグメント化を制約最適化問題に変換し，汎用ソルバーを適用して最適解を求めた．また，この解がフラグメント連結リスクが回避できるているかを分析した．

使用したソルバーは，Sugar-2.3.4 [3]である．Sugar では目的関数の係数は整数しか扱えないので，コストは 10 倍して小数点以下は四捨五入した．目的関数を正の定数を乗じても最適性に代わりはないが，四捨五入により小さい誤差が生じている．厳密な意味での最適化になっていないが，比較的良い近似を与えていると考えられる．

実験環境を次に挙げる．

OS: Ubuntu Server 20.04.2 LTS
CPU: Intel Xeon CPU E5-2609 v4 @1.70GHz
コア数: 8
メモリ: 32GB
COP solver: sugar-2.3.4
SAT solver: cadical-sc2020[4](COP solver 内部で動作)

3：https://cspsat.gitlab.io/sugar/
4：https://satcompetition.github.io/2020/

実験に使用したデータベースは，UCL Machine Learning Repository [5]の Adult データである．このデータベースの属性とそのドメインは表 5 で示している．このデータベースでは値に整数をとるような数値属性があるが，カテゴリ属性とみなして属性依存度を求めている．

表 5　Adult データベースの属性とそのドメイン

属性	ドメイン
age	整数
workclass	{Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, ...}
fnlwgt	整数
education	{Bachelors, Some-college, 11th, HS-grad, ...}
education-num	整数
marital-status	{Married-civ-spouse, Divorced, Never-married, Separated, ...}
occupation	{Tech-support, Prof-specialty, Sales, Farming-fishing, ...}
relationship	{Wife, Own-child, Husband, Not-in-family, ...}
race	{White, Asian-Pac-Islander, Amer-Indian-Eskimo, Black, Other}
sex	{Female, Male}
capital-gain	整数
capital-loss	整数
hours-per-week	整数
native-country	{United-States, Cambodia, England, Puerto-Rico, ...}
salary-class	{>50K, <=50K}

機密性制約と可視性制約は表 6 のように与えた．入力の想定では，センシティブな属性を {occupation, salary-class} として，それ以外の属性を 3 つ以上組合せたものを個人が特定できうるものとして扱っている．そのため機密性制約では，これらの組合せが同じフラグメントに含まれないように制約を与えている．

各属性値間の T 値の算出と分類，およびこれらを基にした制約生成プログラムは C++で実装した．依存度の分類で使う定数は $\alpha = 1.6$ とした．β は設定せず，属性依存の分類はすべて弱い依存か依存が存在しないかのどちらかにした．

6.1 結　　果

実験の結果得られたフラグメントの集合が表 7 になる．結果の分析のため，表 8 に属性依存を考慮しないときのフラグメント化の結果を挙げる．表 7 と表 8 のそれぞれの計算時間を表 9 に挙げる．

表 7 と表 8 とを比較する．依存を考慮しないときは，3 つの属性 workclass，edunum，relation が 1 番目のフラグメントに属している．一方で，依存度を考慮したとき，これら 3 つの属性がどのフラグメントにも含まれず，公開されないことになる．

5：https://archive.ics.uci.edu/ml/index.php

表 6　機密性制約・可視性制約の一覧：ソフト制約のコストは丸括弧で囲まれた数値

機密性制約	$\{a_i, a_j, a_k, a_s\}$
	$(a_i, a_j, a_k \in$ {Adult の属性 } - {occupation, salary-class}, $a_i \neq a_j \neq a_k$, $a_s \in$ {occupation, salary-class})
可視性制約（ハード）	native-country
	occuaptin
	fnlwgt ∧ salary-class
	education
	marital ∨ relation
	age ∧ sex
	hours-per-week ∧ salary-class
	race ∧ salary-class
	race ∧ salary-class
	capital-gain ∧ capital-loss
可視性制約（ソフト）	race ∧ salary-class (50)
	sex ∧ salary-class (50)
	age ∧ salary-class (20)
	hours-per-week ∧ salary-class (20)
	marital-status ∧ salary-class (30)
	(sex ∧ education) ∨ (sex ∧ workclass) (40)
	(race ∧ education) ∨ (race ∧ workclass) (40)

これら 3 つの属性が 1 番目のフラグメントに含まれるときは，2 番目のフラグメントに含まれる属性との属性ペアに対する依存度がコストとして発生する．このコストを表 10 に挙げる．

表 7　依存度を考慮したフラグメント化（$m \leqq 2$）

非公開	fragment 1	fragment 2
workclass	age	fnlwgt
edunum	education	occupation
relation	marital	hour-per-week
	race	salary class
	sex	
	capital-gain	
	capital-loss	
	native	

表 8　依存を考慮しないフラグメント化（$m \leqq 2$）

非公開	fragment 1	fragment 2
	age	fnlwgt
	education	occupation
	marital	hour-per-week
	race	salary class
	sex	
	capital-gain	
	capital-loss	
	native	
	workclass	
	edunum	
	relation	

表 9　フラグメント化の計算時間

属性依存度を考慮する	属性依存を考慮しない
17m42.55s	0m0.88s

表 10　属性 work class，edunum，relation がフラグメント 1 に属するときのコスト

	work-class	edunum	relation
fnlwgt	19	20	23
occupation	168	210	162
hour-per-week	95	89	148
salary class	136	189	342

6.2 考　　察

表 7 と表 8 とを比較して，提案手法で属性依存によるフラグメント連結リスクを回避できるているか分析する．

依存度を考慮する提案手法では，表 7 のように 3 つの属性 workclass，edunum，relation がどのフラグメントにも含まれないようになっている．これらの属性は，依存を考慮しないときの表 8 ではフラグメント 1 に含まれる．これらの属性がフラグメント 1 に含まれるときに受けるコストが表 10 である．つまり，依存度を考慮することで表 10 のコストを回避できる．

もう少し詳しく分析するため，workclass で一番コストが大きいものに着目する．これは occupation とのコストで 168 である．このコストは workclass と occupation との属性値ペアで最大の T 値である T-score($Self\text{-}emp\text{-}not\text{-}inc$, $Farming\text{-}fishing$) を 10 倍して小数点以下を四捨五入したものである．workclass と occuaption とは，どちらも職業に関するデータであり，直感的にはほぼ同じことを指している．その中でも (Self-emp-not-inc, Farming-fishing) という 2 つの属性値は繋がりが強い．ある人が自営業かつ会社でないという workclass 属性を持つなら，その人の occupation 属性が林業，漁業ではないかという推測は導きやすい (表 5 では一部だが各属性のドメインを挙げている)．

もし，workclass がフラグメント 1 に含まれた時を考える．先に述べたように workclass と occupation は強い依存がある．そのためフラグメント 1 に含まれる workclass の属性値からそれに対応するフラグメント 2 に含まれる occupation の属性値を推測できる．よって，これらの関係からフラグメント 1，2 は連結されてしまう．すると表 6 で入力として与えた機密性制約にすべて違反することになる．つまり，提案手法の依存度を考慮したフラグメント化はこのフラグメント連結リスクを回避できたことになる．

7　まとめと今後の課題

先行研究で提案された属性依存関係を考慮したフラグメント化に対して，1) 属性依存度を統計的な尺度に基づいて評価し，2) その依存度に関する最適化問題として定式化することで，拡張した．属性依存度は各属性値間の T 値を計算し，その値を指標とした．フラグメント化は制約最適化問題に変換した．制約最適化問題の制約として，機密性制約，強い可視性制約（ハード制約）と弱い可視性制約（ソフト制約），強い依存関係（ハード制約）と弱い依存関係（ソフト制約）に関する制約を加えた．制約最適化問題の目的関数は，ソフト制約にかかるコストの総和として設計し，目的関数が最小になるように定式化した．

現実のデータベースを用いて，属性依存度を考慮したフラグメント化を行った結果，属性依存によるフラグメント連結リスクを回避するような解が得られることが確認された．

今後の課題としては，提案手法の精度を定量的に評価したい．

謝　　辞

本研究は JSPS 科研費 JP17K17725 の助成を受けたものである．

文　　献

[1] 小栗秀暢. プライバシー保護データ流通のための匿名化手法. システム／制御／情報, Vol. 63, No. 2, pp. 51–57, feb 2019.

[2] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, and Pierangela Samarati. Fragments and loose associations: Respecting privacy in data publishing. *Proceedings of the VLDB Endowment*, Vol. 3, No. 1, pp. 1370–1381, sep 2010.

[3] Sabrina De Capitani Di Vimercati, Sara Foresti, Sushil Jajodia, Giovanni Livraga, Stefano Paraboschi, and Pierangela Samarati. Fragmentation in presence of data dependencies. *IEEE Transactions on Dependable and Secure Computing*, Vol. 11, No. 6, pp. 510–523, nov 2014.

[4] (C)JAPAN HEALTH PROMOTION and FITNESS FOUNDATION. 成人喫煙率（厚生労働省国民健康・栄養調査）. `https://www.health-net.or.jp/tobacco/product/pd100000.html`, 2020. [Online; accessed 10-January-2022].

[5] Christopher D. Manning and Hinrich Schűtze. *FOUNDATIONS OF STATISTICAL NATURAL LANGUAGE PROCESSING*. The MIT Press, 1999.