DEIM Forum 2012 B7-5

属性をもつグラフのk-匿名化手法

野澤　佳世†　　渡辺知恵美†

† お茶の水女子大学人間文化創成科学研究科　〒 112–0012 東京都文京区大塚 2 丁目 1 番 1 号
E-mail: †{nozawa.kayo,chiemi}@is.ocha.ac.jp

あらまし　ソーシャルネットワーク上に蓄積されたデータ (以下ソーシャルデータ) をマイニングすることによって，提供できるサービスの可能性が広がると考えられる．そのため，近年ではソーシャルデータの有用性に注目が集まっている．その反面，ソーシャルデータにはサービスを利用しているユーザの個人情報が含まれているため，データを公開する際にはユーザのプライバシを考慮しなくてはいけない．本稿では既存のグラフ匿名化手法である k-automorphism [10] を拡張し，個人ノードの属性を考慮したグラフデータの匿名化手法を提案する．具体的にはユーザ属性を k-anonymity を用いて匿名化し，匿名化された属性に従ってノードにラベルを付け，ラベル付きのグラフに対してグラフの匿名化を行った．また，ラベル付きのグラフの匿名化に対し，コスト計算及び頻出サブグラフ抽出方法の拡張を行った．提案手法に関して，匿名化安全性とマイニング精度に対して実験を行った結果，ソーシャルデータの有用性に保ちつつ匿名化を実現していることが示された．

キーワード　ソーシャルネットワーク・セキュリティ・プライバシー保護

A K-Anonymization Scheme for Graph Considering User's Attribute.

Kayo NOZAWA† and Chiemi WATANABE†

† Ochanomizu University Graduate School of Humanities and Scieces
2–1–1 Otsuka, Bunkyoku, Tokyo, 112–0012 Japan
E-mail: †{nozawa.kayo,chiemi}@is.ocha.ac.jp

Abstract　With the growth of social networks service (below SNS), analyzing and mining social network data are gaining attraction to companies and researchers. However, when a data owner of social network data publishes the data, the owner should notice that the social network data includes privacy sensitive information. For publishing social network data without leaking privacy sensitive information, many anonymization techniques are proposed. The fact affects privacy and utility for published social network data. In this paper, we propose an anonymization algorithm which considers both graph structure and node profile data.

Key words　Social Network , Security ,PrivacyPreserving

1. はじめに

近年ソーシャルネットワークサービス (以下 SNS) が流行しており，そのサービス数も増加している．また，ユーザが属するコミュニティや個人の行動パターンの傾向を SNS サイト上で収集し，マイニングを行うことは，マーケティングや社会分析に役に立つと考えられる．サービスを提供している企業は上記のデータに加え，ゲームへの課金額などユーザ本人にしか確認出来ないような情報をソーシャルデータとして蓄積している．このように SNS は利用しているユーザの詳細な情報が含まれていると考えられるため，これらをマイニングすることに注目が集まっている．そこで SNS 会社は蓄積されたソーシャルデータをマイニング業者などの分析者に公開し，新たなサービスの構築に役立てようとしている．しかしソーシャルデータには機密情報が含まれており，プライバシが懸念されるため，データを公開する際には匿名化することが求められる．ソーシャルデータに対する従来の匿名化手法は，ソーシャルデータがグラフ形式であるというデータ構造のみに注目している手法が多かった．本稿では従来の匿名化手法を拡張し，より強固なプライバシ保護の実現を目指す．

一般にソーシャルデータはソーシャルグラフと呼ばれるグラフ構造のデータと，ユーザのプロフィールである属性表で表されることが多い．ソーシャルデータの例を図 1 に示す．図 1(a) は SNS 内で各ユーザがもっている属性値と，それらのユーザがどのノードに割り当てられているかを示している．ノード名とユーザ名に次いで各ユーザのプロフィールである属性値が

記されている．図 1(b) では関係のあるユーザ同士を辺で結び，SNS 内での友人関係を表すソーシャルグラフを形成している．

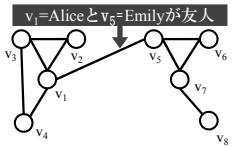
ソーシャルデータを匿名化する最も単純な手法は，個人の識別子をデータから取り除いて，ID など，個人が特定できない情報に置き換えるというものである．しかしこの手法は，匿名化手法として十分でない．図 1(c) では識別子であるユーザ名を ID :u_1, u_2 ... に置き換えているが，攻撃対象ノードの友人関係を背景知識として用いた攻撃は成立してしまう．Alice に友人が 4 人，すなわち 1 つのノードの周りに 4 つのノードをもつ構造のサブグラフを背景知識として持っていた場合，敵はこれと一致するグラフ構造をソーシャルデータから検出することで攻撃を行う．図 1(b) の場合，敵が持っているサブグラフに該当するノードは v_1 のみであり，敵は v_1 がユーザ Alice に対応するノードであると特定することができる．また特定されたユーザの属性表を見ることで，機密属性であるゲームの課金額を知ることができる．

近年提案されているデータ匿名化手法 [10] では，グラフ構造を変化させ同じ構造のサブグラフを複数作り出すことによって，個人のプライバシを保護する．3 ユーザと友人関係にある Cathy をグラフ内から特定する際，グラフ構造から見た候補は $\{u_2, u_3, u_5, u_6\}$ であり，グラフから Cathy を一意に定めることはできない．よって，グラフの匿名性が保持されている．しかし敵が,「Cathy は Ocha Univ. に通っている」という属性に関する背景知識を取得したと仮定する．すると候補のうち該当するノードは u_3 のみであり，そのノードが攻撃対象であるということが攻撃者に特定されてしまう．(図 2)

これより，ソーシャルデータの匿名化を行う際にはグラフ構造だけではなく，属性も匿名化する必要があることが分かる．そこで我々は従来の手法を拡張し，グラフと属性値の両方に対して匿名化を行う手法を提案する．

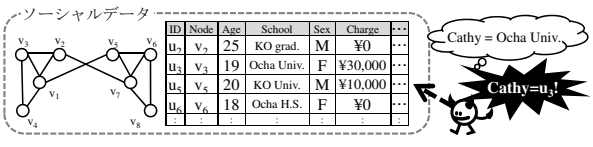
Node	Name	Age	School	Sex	Charge	⋯
v_1	Alice	14	OchaJ.H.S	F	¥200,000	⋯
v_2	Bob	25	KO grad.	M	¥0	⋯
v_3	Cathy	19	OchaUniv.	F	¥30,000	⋯
v_4	David	20	KO Univ.	M	¥1000	⋯
v_5	Emily	20	KO Univ.	M	¥10,000	⋯
v_6	Flora	18	OchaH.S.	F	¥0	⋯
:		:	:	:	:	⋯

(a)ユーザ情報とグラフノードの対応表



(b)友人関係を表すグラフ

図 1　ソーシャルデータの表記方法



ID	Node	Age	School	Sex	Charge	⋯
u_2	v_2	25	KO grad.	M	¥0	⋯
u_3	v_3	19	Ocha Univ.	F	¥30,000	⋯
u_5	v_5	20	KO Univ.	M	¥10,000	⋯
u_6	v_6	18	Ocha H.S.	F	¥0	⋯
:	:	:	:	:	:	:

図 2　属性からユーザ情報が推測されてしまう例

2. ソーシャルデータ匿名化に用いる手法

本研究では匿名化手法として，グラフ構造に対して k-automorphism [10] を，属性には k-anonymity [8] を用いる．

2.1 k-automorphism

グラフ構造からプライバシが侵害されることを防ぐ手法は数多く提案されているが，本稿ではその一つである k-automorphism [10] を用いる．この手法では，単純な匿名化をしたグラフ G' の構造を変化させ，すべてのノードに対して同じ構造のノードが k-1 個以上存在することを保証したグラフ，k-automorphic graph を作成する．k-automorphic graph を作成する k-Match アルゴリズムの概要を図 3 に示す.

まず G' を n 個の部分グラフに分割し，それらの部分グラフのうち同型なものが k 個以上存在するグラフ同士を，それぞれ m 個 (m=1,2,...) のグループ $U_1, U_2, ..., U_m$ に分類する．グループ内のサブグラフをそれぞれブロック $P_{i1}, P_{i2}, ..., P_{ij}$(j=1,2,...k) とする．頻出サブグラフを導き出す際には，各ブロック内に含まれるノードが重複しないように注意する．次にグループ内のブロックを 1hop ずつ拡大して同型なサブグラフを作成する．この操作は，グループ内のブロックすべてに対して並列的に行う．もし拡張するためのノードが足りないブロックがあったら，そのブロックにはダミーノードを追加して一様にブロックを拡張する．拡張する際に，グラフ構造を変化させるために追加される辺の数を匿名化コストとして定義し，ブロックを拡張することでコストが増加したら拡張をやめる．論文 [10] 中では，辺の編集コストは以下のように定義されている．

定義 ブロック P_{ij}(j=1,2,...,k) を持つグループ U_i が与えられていたとき，グループ U_i の匿名化コストは以下のようになる

$$COST(U_i) = AlCost(U_i) + 0.5 * (k-1) * \sum_{j=1}^{k} |CrossEdge(P_{ij})|$$

この式中の $AlCost(U_i)$ とはサブグラフを同型にする際に追加される辺の数であり, $|CrossEdge(P_{ij})|$ はブロック間をまたぐ辺の数である．また, $0.5 * (k-1) * \sum_{j=1}^{k} |CrossEdge(P_{ij})|$ はブロック間をまたぐ辺を追加した数と一致する．グラフ全体の匿名化コストは 1 グループごとの匿名化コスト $COST(U_i)$ を合計したものとなる．図 4(b) のグラフにおける $AlCost$ は辺 (v_6, v_8) であり, $CrossEdge$ は辺 (v_2, v_7) となる.これより，図 4(b) の匿名化コストは 1.5 であるといえる．

最後に，ブロック内の部分グラフが同型になるように各ブロックのグラフに辺または頂点を追加する．ブロック間をまたがっている辺は，対称な辺をコピーすることによって各部分グラフを同型にする．ブロックを拡張し過ぎてしまいグループ内の各サブグラフの構造があまりにも異なってしまうと，同型にする際に大量に追加することになるため，コストの閾値を定めることはとても重要である．

これらの手順を踏んで k=2 で匿名化したグラフは図 4(b) となる．同型なグラフがそれぞれ 2 個以上存在しているため，敵は攻撃対象のノードを一意に定めることが出来ない．またこのグラフは自己相似となるため，敵がどんなに大きなサブグラフを背景知識として持っていても，個人を特定することができない．

2.2 k-anonymity

次に，SNS を利用しているユーザの属性を匿名化する手法に

入力：元のグラフG，パラメタk
出力：匿名化されたグラフG*

1. Gの属性を削除する
2. Gをn個のサブグラフに分解
3. n 個のサブグラフを類似した構造をもつ m 個のグループに分類（グラフ分割・分類アルゴリズム（図8）を参照）
4. for each グループUi (1≦i≦m) do
5. サブグラフPij (1≦j≦k)の構造を同型になるよう編集する
6. end

図 3　k-Match アルゴリズム

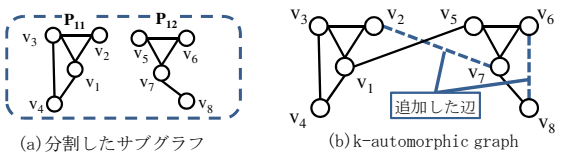


図 4　k-automorphic graph の作成手順

ついて説明する．単体では個人を特定することはできないが，他の情報と組み合わせることで非公開の情報を推測することができるような属性を，準識別子 (quasi-identifier) という．図 2 の例では，敵は Cathy の通う学校名を準識別子として攻撃をしている．識別子を伏せるだけではなく準識別子の存在も考慮にいれてデータの匿名性を保つ手法として，k-anonymity [8] がある．この手法では，公開されているデータと背景知識をどのように組み合わせても，該当ノードの k 個以上の絞り込みができないように準識別子を匿名化する．これを，k-匿名性を保持するという．具体的な手法は，1 つの準識別子に共通する性質を抽象して一般化し，1 つの概念にまとめるというものである．機密属性である課金額は準識別子ではないため，一般化しないように注意する．

具体例を示すため，前述のソーシャルデータに k-anonymity を適用する．図 1(a) の属性表において，グラフ構造との組合せによって識別子を推測できるような準識別子は，Age と School であると考えられる．これらの性質を一般化することにより，ソーシャルデータに k-anonymity を適用した結果を図 5 に示す．図 5 のソーシャルデータでは，同じような属性の組合せを持つノードが少なくとも 2 つ存在しているため，k=2 で匿名性が保たれているといえる．

Node	ID	Age	School	Sex	Charge	⋯
v_1	u_1	14	Ocha J.H.S	F	¥200,000	⋯
v_2	u_2	25	KO grad.	M	¥0	⋯
v_3	u_3	19	Ocha Univ.	F	¥30,000	⋯
v_4	u_4	20	KO Univ.	M	¥1,000	⋯
v_5	u_5	18	Ocha H.S.	F	¥10,000	⋯
v_6	u_6	21	KO Univ.	M	¥0	⋯
v_7	u_7	22	T Univ.	F	¥18,000	⋯
v_8	u_8	28	T grad.	F	¥2,000	⋯

Node	ID	Age	School	Sex	Charge	⋯
v_1	u_1	14	Ocha	F	¥200,000	⋯
v_2	u_2	25	KO	M	¥0	⋯
v_3	u_3	19	Ocha	F	¥30,000	⋯
v_4	u_4	20	KO	M	¥1000	⋯
v_5	u_5	18	Ocha	F	¥10,000	⋯
v_6	u_6	21	KO	M	¥0	⋯
v_7	u_7	22	T	F	¥18,000	⋯
v_8	u_8	28	T	F	¥2,000	⋯

3-anonymity
3-anonymity
2-anonymity

図 5　k-anonymity を適用したソーシャルデータ

2.3　従来のソーシャルデータ匿名化手法の問題点

敵が，4 人と友達であり Ocha に通っている Alice というノードをグラフ内から特定する場合，図 4 のグラフ構造より，Alice の候補ノードは $\{u_1, u_7\}$ である．Ocha に通っているという情報より，図 5 の属性値からの候補は $\{u_1, u_3, u_5\}$ と複数存在することになるが，二つの絞り込み結果に u_1 が重複しているためこのノードが Alice に当たるということが敵に判明してしまう．以上のことより，ユーザの属性値とグラフを別々に匿名化すると，敵が対象ユーザのグラフと属性の背景知識を両方をもっており，それらを組み合わせて攻撃した場合にはユーザが特定されてしまう．

3.　属性を考慮したグラフ匿名化手法

2 節ではユーザの名前を ID で置き換え，更にグラフ構造と属性を匿名化することによりユーザのプライバシを保持しようとした．しかし前述の通り，互いの匿名化手法を考慮しないと十分な匿名化手法とはいえない．

我々は敵の攻撃を，グラフの特徴から該当ユーザの候補ノードを絞り込み，絞り込んだノードの属性を見てユーザを特定する，という 2 つの段階を踏んでいると想定する．これより，グラフ構造と属性の互いを考慮しながら匿名化する手法として，2 種類のアプローチが考えられる．

手法 A)　トポロジが同じまたは類似しているサブグラフ集合を求め，それらのサブグラフ内の各ノードの属性に対して匿名化手法を適用する．

手法 B)　各ノードの属性を見て，類似したノードで k 個以上のサブグラフが構成されるように辺の修正を行う．

手法 A では k-Match アルゴリズムを適用してグラフをグループ分割してから，グループ内サブグラフの属性の匿名化を行う．手法 B ではまず，ノードの属性に k-anonymity を適用して同じ属性を持つノードをグループ化してから，それに基づいてサブグラフのグループを構成していく．この際，k 個のサブグラフの各ノードが同一の属性になるように構成する．グラフ構造，属性の両方を考慮して匿名化を施すという点では同じであるが，どちらを基準として匿名化をしていくか，ということが異なる．

また，どちらの手法においても注意すべき点がある．それは匿名化を重視するあまり，元のデータに対して一般化または変更を過度に行ってしまい，利便性を失わないようにすることである．グラフ構造を基準として匿名化をする手法 A の場合は，作成されたグループ内のサブグラフ間で構造が対応するノードの属性を一般化する．図 4 のグラフにおいて u_1 と u_7 は構造が同じであるため，2 つのノード属性を区別できないように一般化する．すると u_1 と u_7 では各ノードの属性値にあまり共通点が無いため，年齢は 30 歳以下，学校は”通っている (Yes)” か”いない (No)” の 2 値のみになり，図 6 のように値が一般化されすぎてしまう．

ID	Node	Age	School	Sex	Charge	⋯
u_1	v_1	14	Ocha J.H.S	F	¥200,000	⋯
u_7	v_7	22	T Univ.	F	¥18,000	⋯

ID	Node	Age	School	sex	Charge	⋯
u_1	v_1	Under30	Yes	F	¥200,000	⋯
u_7	v_7	Under30	Yes	F	¥18,000	⋯

図 6　一般化した属性の値

また手法 B においても同様の事が言える．k-Match アルゴリズムでは同じグループのサブグラフを同型にすることを目的としている．手法 B を適用するために属性が類似している値を同型にすると，辺を大量に追加しなくてはならなくなることがある．例えば図 5 の属性値表では，属性が似ているノード同士の組合せは $\{u_1, u_3, u_5\}$, $\{u_2, u_4, u_6\}$, $\{u_7, u_8\}$ である．これらのノード周辺の属性を一致させつつブロック内の各サブグラ

フが同型になるように，図 4 のグラフ構造を変化させていく．まず，類似した属性を持つノードの辺の本数を揃えるため，辺 (3,8) と辺 (5,8) を追加する．図 7(a) のグラフでは辺の本数はそろったが，サブグラフが同型でないため，さらに辺を追加して形を揃える．(図 7(b)) すると追加する辺の数が極端に多くなり，本来あるはずのない辺が大量に追加されてしまう．

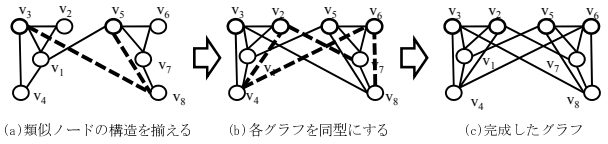


図 7　辺を大量に追加した場合

我々は以上の点を考慮しつつ，ソーシャルデータのグラフ構造とノード属性の両方を考慮した匿名化アルゴリズムについて検討した．

先行研究 [13] ではサブグラフの分割後，各ブロック間でトポロジが類似しているノード同士が同じ値になるように属性値を匿名化する，匿名化手法 A についての考察を行った．先行研究 [13] より，手法 A の問題点が 2 つ発見された．1 つめは，グループ内のサブグラフの中に自己相似があった場合である．図 8 を k-Match アルゴリズムで導き出されたグループのうちの 1 つであるとする．このグループの中で属性を類似させるノードの組合せとして，$\{u_1, u_5\}, \{u_1, u_7\}, \{u_3, u_5\}, \{u_3, u_7\}$ の 4 通りが考えられる．これらの組合せのうち，1 つを選び出して匿名化を行う．このようにどの組合せを採用するか考え出すことが全てのグループ，全てのノードで発生する可能性がある．このため組合せ爆発が起こり，この問題は NP 完全であるといえる．

また，類似した属性値をもつノード同士の組合せが必ず発見される訳ではない．例えば図 8 では，$\{u_1, u_5\}$ の属性値が類似しているためこの組合せを採用すると，残った組合せである $\{u_3, u_7\}$ に対してノードの属性値を類似させることになる．しかし，図 5 より u_3 と u_7 の属性がかけ離れていることが読み取れるため，個の組合せを採用すると値が過度に一般化されてしまう．よって 2 つめの問題点として，全てのノードに対して最適なノードの組合せが発見できるわけではないということが挙げられる．

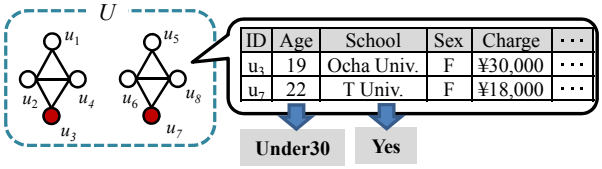


図 8　自己相似のある同型グラフ

以上のことより，ソーシャルデータに手法 A を用いるためには解決すべき問題が多くあるということが分かる．そこで本稿では，上記に述べた二つのアプローチのうち手法 B を採用した．

4. 提 案 手 法

手法 B ではまずユーザの属性に対し k-anonymity を適用し，構造だけでなく属性も類似しているサブグラフを求める．そこで，属性の値も一致させるグラフ分割が行えるように，k-automorphism を拡張することとした．

4.1 匿名化コストの改良

図 9(a) のブロック $\{v_1, v_2, v_3, v_4\}$ と $\{v_5, v_6, v_7, v_8\}$ を拡張する場合，従来のアルゴリズムで考慮すべきコストは (v_5, v_6) 間に追加される辺の本数である AlCost と，(v_2, v_5) 間に追加される辺の数である CrossEdge のコストのみであった．よって図 9(a) における拡張の際の匿名化コストは 2 である．しかし実際に生じるコストはこれらだけではない．それで我々は k-automorphism のコスト計算を見直し改良を行った．

図 9(a) は 2 つのブロックをもち，CrossEdge 以外にもブロック外の v_{10}, v_{11}, v_{12} と繋がる辺が存在する．このグラフに対して，敵がサブグラフ $\{v_7, v_3, v_{12}, v_{13}\}$ を背景知識として攻撃すると該当サブグラフが 1 件となり，k-匿名性を満たさなくなる．よって，ブロック拡張を行わず匿名化をした際には，これらのノードの存在も考慮してサブグラフの構造を類似させなければならない．そのため図 9(a) では v_7 から v_{10}, v_{11}, v_{12} のそれぞれに伸びる辺を 3 本追加することになるため，辺の追加コストがかかると予想される．そこで，ブロック外に伸びる CrossEdge 以外の辺の数を OutEdgeCost として定義し，全体の匿名化コストに含める．

また，拡張の際に追加される辺も存在する．図 9(a) のブロックを拡張した図 9(b) のブロック分割の場合，並列的にブロックを拡張するため v_7 を 3 つのダミーノードとつなぐ必要がある．本手法ではこの際生じるコストを ExpandCost とし，これw 考慮してグラフ拡張を行う．

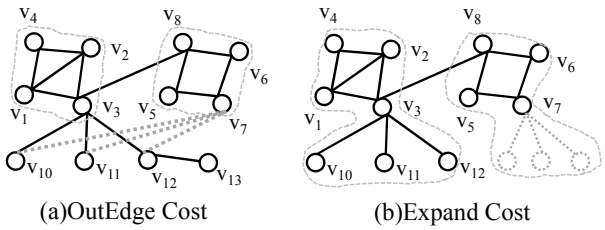


図 9　ブロック拡張の際に生じる辺

以上のことより，拡張時に生じるコストを新たに以下のように定義する．

$$COST(U_i) = AlCost(U_i) + CrossEdgeCost(U_i) + OutEdge(u_i) + ExpandCost(U_i)$$

提案手法では上記の匿名化コストでコスト計算をし，拡張前のコストを上回るまでグラフ拡張を行う．

4.2 抽出方法の改良

k-Match アルゴリズムではサブグラフのグループを作成する際に，グラフ構造を考え同型であるか否かを判別していたが，提案手法では構造だけでなく属性も一致したサブグラフでグ

ループを形成することを目標としている．そこで K-Match アルゴリズムに対し，属性も考慮に入れたグラフ分割が行えるよう拡張した．

拡張したグラフ分割アルゴリズムを図 10 に示す．

アルゴリズム2'：グラフ分割アルゴリズム
入力：元のグラフG 出力：m個にグループ分けされたn個のサブグラフ
1. while |E(G)| > 0 do **2. ソーシャルグラフ内の属性値に対してk-anonymityを適用する** 3. 最小サポート値min_sup>kを満たす最大の頻出サブグラフを求める 4. repeat 5. Gjグループ内のサブグラフを1hopだけ拡張させUi'に入れる 6. until Cost'(Gj) < Cost'(Gi') 7. Uiのサブグラフに含まれるエッジをGから削除する 8. end

図 10　グラフ分割アルゴリズム

まず k-anonymity を用いて属性を匿名化し，匿名化することによって準識別属性が同値となったプロファイルをを持つノード同士に，同じラベルを付与する．例えば図 5 において，u_1, u_3, u_5 の属性値が匿名化によって同一な属性値のプロファイルとなるため，これらのノードには同一のラベルを付ける．図 11 はラベルをノードの色で表したグラフであり，同色のノードには同じラベルが付いているものとする．次に，拡張前の k-Match アルゴリズムと同様頻出サブグラフを発見し，それらをグループ U_i に分類する．k 個以上の頻出同型サブグラフを求めるのに SUBDUE [12] を使用した．SUBDUE はラベルを持つ 1 つの大きなグラフデータに対して，閾値以上登場する部分同型部分グラフを抽出することができる．先行研究である k-automorphism ではラベルを考慮せず部分グラフの構造のみ考慮していたため，例えば図 11 のグラフからは $\{u_1, u_2, u_3\}$,$\{u_4, u_5, u_6\}$,$\{u_7, u_8, u_9\}$ という 3 つのサブグラフが検出される．我々の提案手法ではノードにラベルを付けラベル付きのグラフに対して頻出サブグラフを求めるため，図 11 のグラフから導き出されるサブグラフは $\{u_1, u_2, u_3\}$,$\{u_4, u_5, u_6\}$ となる．このように，ラベルを考慮したグラフ分割を行うと頻出サブグラフの数が少なくなり，匿名化の際に追加する辺の数が多くなることが予想される．そこで本手法では，グループを導出する際にサブグラフ内のノードの重複を許すこととした．重複を許すことで $\{u_1, u_2, u_3\}$,$\{u_4, u_5, u_6\}$ に加え，$\{u_3, u_{11}, u_6\}$ を導き出すことができるようになる．

重複を許すことで発生する問題もある．ソーシャルデータにはスケールフリー性があるため，図 12 ようにハブとなるノードが存在することになる．ノードの重複を許すとハブノードを含む頻出サブグラフが大量に導き出されることになり，それらの構造を全て類似させると匿名化コストが非常に多くなることが予想される．そこで導き出されたサブグラフを全て採用するのではなく，頻出サブグラフ数が k の値の 2 倍以上検出されたらそれ以降はグループに含めないこととした．これより，ハブノードが存在していても，匿名化コストが多くなることを防ぐことができる．

4.3　拡張方法の改良

K-Match アルゴリズムと同様にグラフを 1 ステップずつ拡張し，辺がなるべく多く含まれるようにサブグラフを拡張する．本手法ではノードラベル構成が一致したサブグラフ同士のグループを抽出することを目的としているため，同じラベルが与えられているノードに対して拡張することに注意する．図 13 のグラフでは，1hop 拡張した際に同じラベルを持つノードをサブグラフに含めることができないため，拡張を行うことが出来ない．そこですべてのノードを 1hop 拡張する手法ではなく，それぞれのラベルノードに対して拡張した際のコストを求める．例えば図 13 のサブグラフを拡張する際は，2 つのパターンにおけるそれぞれの拡張コストを計算し，拡張前のコストを超えない限りノードをブロックに含める．これにより，ノードにラベルがついていたとしても，サブグラフが細かくなりすぎずにグラフ分割を行うことができる．

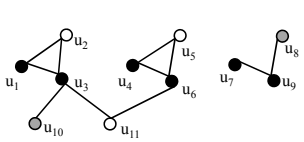


図 11　ラベル付きグラフ

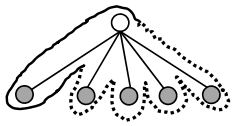


図 12　ハブノードの周辺のグラフ分割

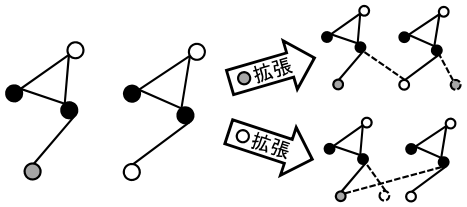


図 13　ラベルを考慮したサブグラフ拡張

グループが一つ導き出されたら，元のソーシャルグラフの複製である作業用のグラフから，検出されたグループ内のサブグラフに含まれる辺を削除する．そして，この操作によって生じた孤立点を削除する．ソーシャルグラフに含まれるすべての辺が頻出サブグラフとして検出されるまで，つまり作業用グラフが空になるまでこれを繰り返す．分割アルゴリズムをこのように拡張することにより，グラフ属性を考慮した k-automorphism を施すことができる．

5. 実　　験

本稿ではソーシャルデータの匿名化手法において，グラフ構造と各属性を組み合わせても個人が特定されないことを目的としている．そこで匿名化手法が十分なものであるか，

- 敵の攻撃に対する防御が適切に行われるか
- 匿名化結果のマイニング精度は適切なものであるか

という点を検証する．本稿で提案した手法を linux マシン上 (Xeon(R) 2.67GHz, RAM 4M byte) に Java 言語で実装し，実験を行った．

5.1　実験に使用したデータ

本実験では実際の SNS データとして，*Prefuse graph (http://prefuse.org/)* を使用した．これは仮想ユーザの SNS データであり，129 個のノードと 161 本のエッジで構成されて

いる．

5.2 安全性の検証

敵はユーザ周辺のグラフ構造であるクエリを持っており，匿名化後のグラフを照らし合わせ個人を特定すると考えられる．そこでまず，匿名化前のソーシャルグラフから任意のクエリ Q を取り出し，匿名化後のグラフとの一致数を調べた．また比較として，匿名化前のラベル付きグラフとクエリの一致数についても実験を行った．頻出サブグラフの最少登場回数 K を 10，ラベル数 L を 10 としグラフの匿名化を行った．実験の結果を図 14 に示す．

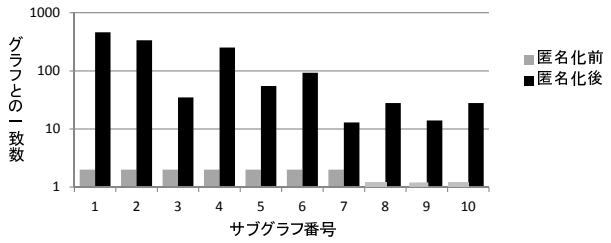


図 14　安全性の検証

匿名化前のグラフではクエリの一致数は高々2 であり，敵の攻撃に対する耐性はあまり期待できないことが分かる．匿名化後のグラフは，多少のばらつきはあるもののクエリ一致数が k の値である 10 を下回ることはなかった．k-automorphism の匿名性を満たしているため，提案手法はグラフ構造からの攻撃に十分対処出来るということが分かる．

5.3 マイニング精度の検証

次に，匿名化後のグラフにおいてマイニングの精度が失われていないか検証した．まず k と l の値をそれぞれ 5,10,15,20 と変化させて匿名化グラフを作成し，追加された辺の本数を調べた．これにより，ノイズとなる辺を過度に追加していないか検証する．追加された辺の本数結果を図 15 に示す．

匿名化後のグラフでは匿名化前のグラフに 100 から 200 本の辺を追加している．また，グループ内に含まれるべき最少ブロック数が増え分割されるサブグラフが小さくなるため，k の値が大きいほど追加される辺の本数が増加した．l=15,20 では k=15,20 の場合において匿名化グラフを作成することができなかった．これはラベル数が多い場合にサブグラフが作成できなかったためと考えられる．

次に，追加された辺がマイニング結果に与える影響について検証する．k-means の k の値を 5 として匿名化前後のグラフをクラスタリングし，相関関係を調べた．結果を図 16 に示す．

図 16 の結果より，グラフの相関関係は 0.5 から 0.8 の間で保たれており，匿名化をしても相関関係は比較的保たれているということが分かる．

以上の実験より，提案した匿名化手法は実際のソーシャルデータ匿名化を行う上で有効なものであるといえる．

5.4 考　　察

本稿ではグラフ抽出の際に，ブロック内でのノードの重複を許してグループを作成した．ノードの重複を許さないグラフ分

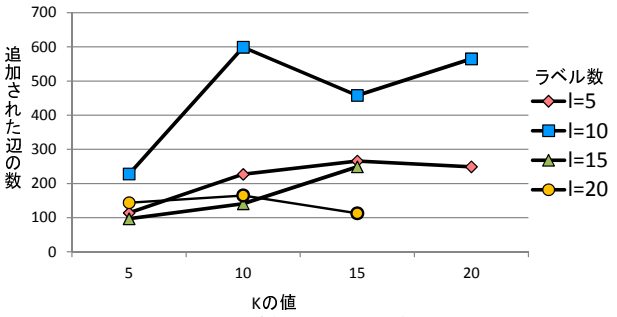


図 15　追加された辺の本数

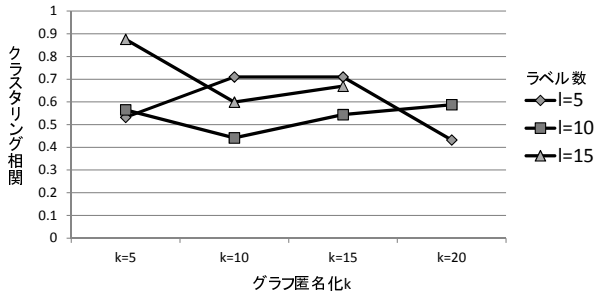


図 16　クラスタリング結果

割手法では，頻出サブグラフが抽出されないこと以外にも問題が生じる．図 17(a) のようなハブノードが存在し，ブロック P_{1j}, P_{2j} がサブグラフ同型であった場合，ブロック同士の構造を合わせる際に OutEdge が大量に追加されることになる．また大量に辺を追加することにより新たなハブノードが作成され，マイニング結果に影響を与える恐れがある．

ノードの重複を許すとハブノードが新たに作成されることはないが，ブロック間で対応するノードの次数が一致していることが保証されない．そのため，ノードの次数を背景知識として用いて攻撃された場合に匿名性を保てない場合がある．今後は利便性と安全性のトレードオフを考え，ノードの重複を許すべきか否かを決定する．

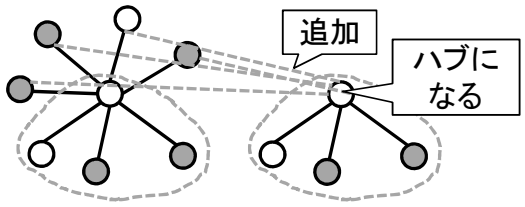


図 17　ブロック間のノードの重複

6. 関 連 研 究

分析のために公開されるデータの匿名化手法についての研究は，これまでにも数多く発表されてきた．本稿で述べた k-anonymity [8] のほかに，l-diversity [2] では，区別がつかなくなった属性値の多様性が維持されるように，k-anonymty を拡張した手法である．属性値を変更するのではなく，データにランダムな値を混入することにより敵の攻撃を撹乱する手法 [4] も提案されている．これらの匿名化手法はリレーショナルデー

タに対して攻撃をモデル化し，匿名化を行うものである．そのため，頂点や辺，近傍グラフ，サブグラフ，またはそれらを結合したものなど，個人を識別するための情報が多いソーシャルデータでは攻撃のモデル化が難しく，従来の匿名化手法をそのまま適用することは難しい．本稿では前節で述べた攻撃対象のうち，個人及び属性に対する攻撃を防ぐ手法について紹介する．ソーシャルデータのプライバシ保護にはグラフ構造とユーザ属性が重要な役割を担っている．そこでソーシャルデータ匿名化のアプローチには，グラフ構造を匿名化するものとユーザ属性とグラフ構造を匿名化するものの二種類が考えられる．

グラフ構造の匿名化として，Liu らは k-degree [7] と呼ばれるグラフ匿名化モデルを提案した．これはソーシャルデータの分析結果に対して影響力が低いと考えられる低い度数のノードに対して匿名化を施し，度数が高いノードそれぞれが最低でも k 個以上存在するように，必要最小限の辺を追加してグラフ構造を変化させることで匿名性を保持する．構造を変化させる手法として他に，本稿で使用した k-automorphism [10] を発展させた手法である，Chemg らの k-Isomorphism [6] があげられる．また，辺の合計数が保持されるように辺の追加と削除をランダムに行うグラフ構造匿名化手法も存在する．Wu ら [9] は低いランクの近似アプローチをとってランダム化を行うことにより，特徴的なグラフ構造を再構築できるような手法を提案した．単純なランダム化をしたグラフよりも元データと類似したグラフが作成されるため，攻撃への脅威は大きくなる可能性がある．ソーシャルグラフをいくつかの部分グラフに分解し，それらが全て類似した構造をもつように辺の編集を行い，構造を一般化する手法である．Hey ら [11] はノードの構造がどのくらい類似しているか計算し，同じような構造をもつグラフの形を変化させることにより，ソーシャルデータを匿名化している．

これまではソーシャルグラフ単体から攻撃される場合の対処法について紹介したが，前述の通り SNS 内におけるユーザ個人の属性情報を背景知識とすることにより，敵に攻撃される場合がある．そこで，グラフ構造だけでなく属性の匿名化も行う手法が提案されている．Zhou ら [3] はどのノードもラベルと呼ばれる属性の値を持っていることに注目し，辺の修正とラベルの一般化を貪欲に行う手法を提案した．まず 1.5 近傍にある特徴的なグラフを抜き出し，DFS 木を使用してそれらをクラスタリングしたものに対して匿名化を行った．これにより，論文 [3] 中で NP 困難であると示された，あるノードから一定のホップ数でいけるサブグラフすべてに対し，同じトポロジをもつグラフが最低でも k 個存在すること保証する匿名化手法，k-neignborhood anonymity を実現した．Zheleva ら [5] はグラフの中に存在するリンクが多様な関係性をもっているという問題について研究した．敵は機密事項であるユーザ同士の関係に攻撃を行うと仮定し，辺の値を匿名化して撹乱することで構造的にも属性的にも匿名性を保持した．具体的な手法としては，まずノードをテーブルとして扱ってそれらの属性を匿名化し，次にネットワーク構造における構造情報の集約結果は保持されるように匿名化を行った．Campan らの手法 [1] では，ノード属性と近隣ノードの構成に注意を払いながらソーシャルデータをクラスタリングし，ソーシャルデータの匿名化を行う．構造上の情報損失と属性の情報損失の双方を考慮し，ソーシャルデータの匿名化を行っている．

7. まとめと今後の課題

本稿では既存の k-Match アルゴリズムを拡張し，敵が攻撃対称ノードの属性値を知っていたとしても攻撃を防ぐことができるように，属性とグラフ構造を匿名化する手法を提案した．

本手法では k-Match アルゴリズムと違い，同型グラフを抽出する際 1 つのノードが複数のグループに含まれることを許している．重複を許すか否か，安全性と利便性のトレードオフを考え決定する．また，構造と属性が一致している k 個のノードの機密属性が全て同じだった場合，ノードが一意に定まらなくても機密属性が敵に判明してしまう恐れがある．そこで準識別子を匿名化したノードの機密属性も多様性を持つものであるように，l-多様性 [2] の概念を属性の匿名化の際に用いることを検討する．また，実験するデータ数を増やし実際のデータに対する影響を求める．

文　献

[1] A.Campan and T.M.Truta: “A Clustering Approach for Data and Structural Anonymity in Social Networks,” In PinKDD, 2008.

[2] A. Machanavgajihara, J. Gehrke, D. Kifer and M.Venkitasubramaniam: “ l-Diversity: Privacy Beyond k-Anonymity” In IEEE ICDE, 2006.

[3] B.Zhou and J.Pei: “Preserving Privacy in Social Networks Against Neighborhood Attacks,” In peoceedings of the 24th International Conference on Data Engineering (ICDE), pp.506-515, 2008.

[4] C.K.Liew, U.J.Choi, C.J.Liew : “A data distortion by probability distribution,” In ACMTODS, 10(3),pp.395-411,1985.

[5] E.Zheleva and L.Getoor: “Preserving in social network,A Survey,” In Social Network Data Analytics, Chapter 10, pp.277-306, 2011.

[6] J.Chemg, A.Wai-Chee Fu and J.Liu: “K-Isomorphism: Privacy Preserving Network Publication against Structural Attacks,” In Proceedings of the 2010 international coference on Management of data (SIGMOD’10), pp.459-470, 2010.

[7] Kun Liu and Evimaria Terzi : “Towards identity anonymization on graphs,” In Proceedings of ACM SIGMOD, Vancouver, Canada, June 2008.

[8] L.Sweeny: “K-anonymity:a model for protecting privacy,” In Uncertainly,Fuziness and Knowledge-based Systems, Vol.10,No.5,pp.557-550,2008.

[9] L.Wu, X.Ying, and X.Wu : “Reconstruction of randomized graph via low rank approximation,” In In proceedings of the VLDB Endowment, In SDM,2010.

[10] L.Zou , L.Chen and M. T. O zsu: “K - Automorphism : A General Framework for Privacy Preserving Network Publication,” In In proceedings of the VLDB Endowment, Vol.2, Issue 1,pp.946-957,2009.

[11] M.Hey, G.Miklau, D.Jensen,D.Towsley, and C.Li: “Resisting Structural Re-identification in Anonymized Social Networks,” In The VLDB Journal, Vol.2010, No.19, pp.797-823, 2010.

[12] N.Ketkar, L.Holder, D.Cook, R. Shah and J.Coble: “Subdue: Compression-based Frequent Pattern Discovery in Graph Data,” In ACM KDD Workshop on Open-Source Data Mining, August 2005.

[13] 野澤佳世，渡辺知恵美: “個人ノードの属性を考慮したソーシャルネットワークデータの k-匿名化手法” 第 153 回データベースシステム研究発表会, C-2-3,2011.