Vol.2011-DBS-153 No.22
2011/11/3

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プライバシ保護の一手法

野澤佳世† 渡辺知恵美†

ソーシャルネットワークサービス（以下 SNS）の発達に伴い，SNS のデータ（以下ソーシャルデータ）の管理やマイニングに対しての研究に注目が集まっている．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．ユーザのプライバシ保護を実現するための最も簡単な方法は，ユーザの氏名などの個人を特定することができるような情報を，元のデータから取り除くことである．しかし，これだけではユーザのプライバシを十分に保護したとはいえない．k-automorphism という手法では，ソーシャルデータがグラフ形式であることを利用して，グラフの構造を変化させることによってデータの匿名化を行った．我々はこの手法を拡張し，ユーザが持つ属性の値を考慮した匿名化手法を提案し，より強固なプライバシ保護手法の実現を目指す．

A Privacy preserving method for social networks considering the attribute value of nodes.

Kayo Nozawa† and Chiemi Watanabe†

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. However, these techniques focus only on the graph structure of the social network data, they don't enough attention to the profile data of each node. 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. We apply k-automorphism as basic algorithm, which generates an anonymized graph that includes more than k isomorphic subgraphs. We extend graph partitioning and clustering algorithm of k-automorphism to apply anonymization to profile data of all nodes of partitioned graphs.

1. はじめに

近年，mixi や Facebook，Gree といったようなソーシャルネットワークサービス(以下 SNS）が流行し，サービス数が増加している．人々は SNS を利用することにより，知人間でのコミュニケーションをより円滑に進めることができるようになった. また，趣味や地域などのつながりから新しい友人を見つけることができるようになった．また，サービスを提供している企業は SNS を利用している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，データとして収集することができるようになった．SNS 上で収集したユーザデータのことをソーシャルデータと呼ぶ．ソーシャルデータを分析することにより，企業はユーザの行動に則した，より細やかなサービスを提供することができるようになる．ソーシャルデータの有用性に注目が集まる一方で，プライバシ問題が懸念されるという問題もある．ソーシャルデータには個人の趣味，嗜好，属するコミュニティなどの個人情報が数多く含まれるため，個人のプライバシを保護する観点から，データを公開する際には匿名化したデータの公開が求められる．

データを匿名化する最も単純な手法は，ユーザの氏名などの個人の識別子をデータから取り除くという方法である．しかし，この方法は匿名化手法としては十分ではない．ソーシャルデータは人と人との繋がりをグラフで表現しているため，SNS 内におけるユーザの友人関係が分かればグラフの構造から個人を特定することができるようになってしまう．そこで，近年提案されているデータ匿名化手法**[5][7][8][9]**では，グラフを変形させることによって，グラフ構造から個人のプライバシが侵害されることを防いでいる．しかし，これらの手法ではユーザの所属や趣味など個人の属性値を考慮していないため，攻撃者が属性値から匿名化されたデータ上の正確な値を推測することが可能となる．

そこで本稿では，従来のグラフデータ匿名化手法の 1 つである “k-automorphism” **[7]** を拡張し，ユーザ個人の属性値から個人が特定されることのないような匿名化手法を提案する．ソーシャルデータに本手法を適用することで，匿名化したデータから個人の情報が特定される確率がより低くなる．プライバシを侵害される確率が低くなることにより，ユーザはソーシャルデータを公開されることへの抵抗が減り，企業はソーシャルデータをマイニングすることができるようになる．本稿の構成は以下のようになっている．まず，2 節でソーシャルデータに対する従来の匿名化手法と，攻撃者の攻撃手法について述べる．3 節では k-automorphism の紹介と，属性値を考慮した匿名化手法に拡張するためにはどのようにしたらよいのかということについて述べる．4 節で関連研究について述べ，最後に 4 節でまとめと今後の課題を提示する．

† お茶の水女子大学大学院　人間文化創成科学研究科　理学専攻情報科学コース
Graduate School of Humanities and Sciences, Ochanomizu University, Tokyo, JPN

2. 従来の匿名化手法と攻撃方法

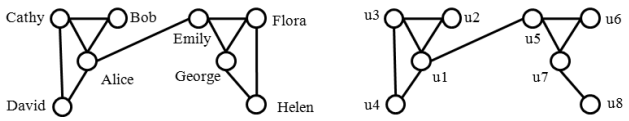
ソーシャルデータは多くの場合グラフで表され，それらはソーシャルグラフと呼ばれる．本稿ではデータ形式としてこのソーシャルグラフを利用する．ソーシャルデータからグラフを作成する手順の例を示すため，SNS に参加しているユーザをソーシャルグラフで表していく．SNS 内のユーザは，以下のように名前や年齢，学校，性別，趣味などの属性値を持っているとする．

（Alice，14，Ocha，Female，Jazz，…）

ソーシャルデータの表記例を図 1 に示す. ユーザの友人関係を表すグラフを図 1(b)に，各ユーザのプロファイルを図 1（a）に示す．この例では上記のユーザ Alice はソーシャルグラフ内のノード v1 にユーザ Alice を割り当てられている．SNS 内の他のユーザも同様に，グラフ内のノードに 1 対 1 の対応で割り当てている．ユーザとノードの対応は図 1（a）の表で表すことができ，ノード名とユーザ名に次いで各ノードが持つ属性が表されている．SNS におけるユーザ同士のつながりは，友人関係にあるユーザのノード間を辺で結ぶことで表す．例えば図 1(b）では辺（Alice，Bob）によって Alice と Bob の両者が結ばれているため，二人は友人関係にあるといえる．

node	name	age	School	Sex	Likes	･･･
v1	Alice	14	Ocha	Female	Jazz	･･･
v2	Bob	25	K.O.	Male	Soccer	･･･
v3	Cathy	20	K.O.	Female	BaseBall	･･･
:	:	:				:

(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

(c)匿名化したグラフ

図 1　ソーシャルデータの表記方法

以上の手順で作成したグラフにはユーザの個人前が含まれているため，このグラフをそのまま公開すると，悪意を持ったユーザによって特定の個人の情報が読み取られてしまう危険性がある．そこで，ユーザのプライバシを保護するために，ソーシャルグラフを匿名化してからデータを公開する．ソーシャルグラフの匿名性を保つ最も簡単な手法は，名前など個人を識別できるような情報を伏せることである．この情報のことを識別属性値（Identifier Attribute）と呼ぶ．図 1 のソーシャルグラフにおいての識別子は，個人名であると考えられるので，これを ID に置き換えてデータを公開する．Alice を u1，Bob を u2 というように ID で置き換えて元のデータを匿名化したソーシャルグラフは，図 1（c）となる．SNS 内のどのユーザがソーシャルグラフ内のどのノードに当たるかということは，図 1（c）から読み取ることができない．よって，図 1（c）のグラフはユーザのプライバシを保持することができているといえる．

2.1 グラフ構造からのユーザ特定

しかし，ユーザ名を伏せても u1 が Alice であると判明してしまうことがある．それは，敵が攻撃対象となるユーザの周辺情報をすでに背景知識として持っていた場合である．例えば，敵が「Alice に 4 人の友達がいる」という情報を背景知識として持っており，匿名化された図 1（c）のグラフから Alice のノードを特定しようとしているとする．ソーシャルグラフでは友人関係を辺で表しているため，友人の数はノードに接続している辺の本数で表される．図 1（c）のグラフにおいて，4 人と友人関係を持っているノード，つまり辺を 4 本もっているノードは u1 のみであるため，敵はグラフ構造から u1 が Alice であると特定することが可能である．このように，識別子を取り除いただけではユーザのプライバシに対して十分に配慮したデータであるとはいえない．

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が数多く提案されているが，本稿ではその 1 つとして，k-automorphism**[7]**という手法を紹介する．この手法では単純な匿名化手法を施したグラフを変形し，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 k-automorphic graph を作成，グラフ構造によるノードの特定を防いでいる．k-automorphic graph である G 内には同型グラフが k 個存在するため，敵がノードを特定する確率は 1/k となる．k-automorphic graph を作成する手法を k-Match アルゴリズムといい，その手順を図 1（c）のグラフを例に，k=2 として以下に示す．まず，ユーザ名を取り除いて単純な匿名化をしたネットワークグラフ G’を n 個のサブグラフに分割し，それらのサブグラフのうちサブグラフ同型であるものをそれぞれグループ Ui(i=1,2,...m)に分類する．ここで注意することは，グループ内に属するサブグラフの数は k 個以上でなくてはいけないということである．図 2 では，点線で G’が 2 つのサブグラフへ分割されることを表しており，2 つのサブグラフはサブグラフ同型であるため同じグループ Ui に属することになる．k は 2 であるので，グループ内のサブグラフ数については条件を満たしている．次に，このグループに属するサブグラフを k-automorphic graph に変換していく．グループ内のサブグラフをそれぞれ P11，P12 とし，P11 と P12 が同型グラフになるように各サブグラフのグラフに辺または頂点を追加する．図 2 のグラフではノード 6 と 8 の間に辺を追加し，P11 と P12 の構造を揃えている．サブグラフ間をまたがっている辺に関しては，辺をコピーすることによって P11 と P12 を同型にする．これらの手順を踏んで匿名化したグラフは図 2 となり，k-automorphic graph 上には同型なグラフがそれぞれ k=2 個以上存在している．もし攻撃者が，4 人の友人を持っているユーザが Alice であるという背景知識を持っていたとしても，図 2 の k-automorpic graph から導き出される Alice の候補ノ

ードが{u1,u7}であるため，ユーザを一意に定めることができない．

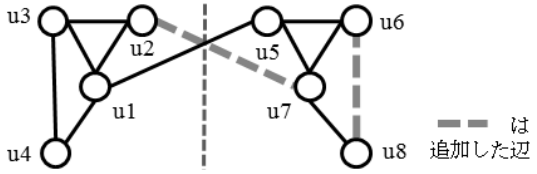


図 2　k-automorphic graph

2.1 従来のソーシャルデータ匿名化手法の問題点

前節まではソーシャルグラフ単体から個人を特定する場合について議論して生きた．2.1 節で述べたように，公開されているソーシャルグラフのノードはすべて SNS 内のユーザに対応しており，それぞれが個々の属性を持っている．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，敵がユーザ個人を攻撃する場合もある．属性情報によってソーシャルデータの匿名性が脅かされる例を，3 人と友人関係にあるユーザ，Bob を特定する手順を示すことによって説明する．

前節の図 2 において，3 人との友人関係を持つノードは{u2,u3,u5,u6}の 4 つである．そのため，属性に関しての背景知識がない状態では，これら 4 つのノードのどれかが Bob である確率は同様に確かである．このため，現段階では正しいノードが特定される確率は 1/4 である．ここで敵が SNS から，Bob は School が KO であり Likes が Soccer で age が 25 である，という知識を取得したとする．4 つのノードのうち，“School=KO”という属性を持つのは{u2,u3,u6}であり，Bob のノードこのどれかであるという絞り込みを行うことができる．さらに 3 つのノードのうち“Likes=Soccer”かつ“age=25”という条件で絞り込んでいくと，u2 のみが残る．すると図 3 で示されているように，u2 が Bob であるということが攻撃者に特定されてしまう．つまり，敵は k 個以上のノードが絞り込めができない，という k-automorphism の匿名性が保てなくなってしまう．

よって，公開されているデータの属性値から元データの識別子を推測されないようにするためには，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．次の章では属性を匿名化する手法について述べる．

3. 属性値を考慮した k-automorphism

3.1 k-anonymity

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て考えると非公開の属性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．

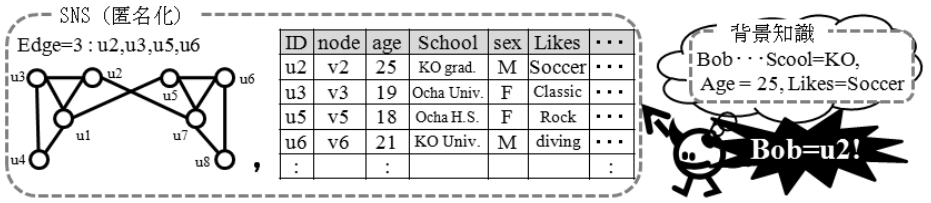


図 3　属性からユーザ情報が推測されてしまう例

グラフ構造から個人を特定するという例では，グラフ構造という準識別子を利用し敵が攻撃を行っている．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 1 つの手法として，k-anonymity[6]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．準識別子の匿名化の具体的な手法は，表現を一般化するということである．表現を一般化するということは，1 つの準識別子に共通する性質を抽象し，1 つの概念にまとめるということである．

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 4 の例において，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, School，Likes である．属性 Age，School，Likes の表現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 4 に示す．図 4 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 3 つ存在しているため，k=3 で匿名性が保たれているといえる．k-anonymity を適用して匿名化したデータを公開することにより，敵が Bob の属性を知っていたとしても，Bob が u2 であるとは特定できず，3 つ以下のノードの絞り込みもできない．

ID	node	age	School	sex	Likes	・・・
u1	v1	14	Ocha J.H.S	F	Jazz	・・・
u2	v2	25	KO grad.	M	Soccer	・・・
u3	v3	19	Ocha Univ.	F	Classic	・・・
u4	v4	20	KO Univ.	M	Base boll	・・・
u5	v5	18	Ocha H.S.	F	Rock	・・・
u6	v6	21	KO Univ.	M	diving	・・・
u7	v7	22	T Univ.	F	running	・・・
u8	v8	28	T grad.	F	Soccer	・・・

⇒

ID	node	age	School	sex	Likes	・・・	
u1	v1	Teen	Ocha	F	Music	・・・	3-anonymity
u2	v2	20's	KO	M	Sports	・・・	
u3	v3	Teen	Ocha	F	Music	・・・	3-anonymity
u4	v4	20's	KO	M	Sports	・・・	
u5	v5	Teen	Ocha	F	Music	・・・	
u6	v6	20's	KO	M	Sports	・・・	2-anonymity
u7	v7	20's	T	F	Sports	・・・	
u8	v8	20's	Ocha	F	Sports	・・・	

図 4　属性値の粒度を下げて表現したソーシャルデータ

3.2 ソーシャルデータに k-anonymity を適用する際の問題点

しかし，属性のみを考えて k-anonymity を適用した場合でも，必ず匿名性が保たれるというわけではない．図 4 は age，School，Likes の 3 つの属性を一般化することにより，最低でも 2-anonymity が保たれている．k-Match アルゴリズムを適用した際の k が 2 だったため，このユーザ属性は k 匿名性を満たしているといえる．しかし敵が，友達が 3 人で 10 代，趣味が Rock（Music），である Emily というノードをグラフ内から特定したい場合を考える．図 4 のユーザ属性をのみを見ると Emily の候補であるユーザは{u1,u4,u5}の 3 つである．図 3 のグラフ構造を見ると，友達が 3 人であるノードは{u2,u3,u5,u6}であり，二つの絞り込み結果に共通しているノードは u5 のみである．よって，グラフ構造と属性から絞り込まれたノード中に共通しているノードである u5 が Emily であることが，敵に特定されてしまう．以上のことより，グラフ構造または属性をそれぞれ単体で匿名化するとデータの匿名性を保てない場合があることが分かる．つまり，ユーザプロファイルデータとグラフデータと別々に匿名化を適用しても，敵が対象ユーザのグラフ上の特徴とプロファイル上の特徴の両方をもっており，それらを組み合わせて攻撃した場合にはそれらの匿名化は意味をなさなくなってしまう．

3.3 解決法

2 節ではユーザの名前を ID で置き換え，更にグラフの構造を変えることにより匿名化をした．前節では属性に k-anonymity を適用することを検討したが，ソーシャルグラフの十分な匿名化手法とはいえなかった．前節より，我々が想定する敵の攻撃は

1. グラフの特徴を見て，該当ユーザの候補ノードを絞り込む
2. 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 つの段階を踏んで行われること考えることができる．また前節の例から，グラフ構造の匿名化と属性の匿名化を別々に議論すると十分な匿名性が保てない場合があるということが分かる．

グラフ構造と属性の 2 つを併せて匿名化する手法としては，2 種類のアプローチが考えられる．

手法 A） まず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．

k-automorphism を例に取った場合，手法 A では，k-Match アルゴリズムを適用して同じ構造をもつサブグラフがある一定の数以上になるようにグラフを編集してから，サブグラフ内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，サブグラフを構成していく．この際，例えば k 個のサブグラフの各ノードが同一の属性グループに属するように構成する．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属性の一般化またはグラフ構造の変更を過度に行ってしまい，ソーシャルデータの有用性を失ってしまうことである．

例えば，手法 A であるグラフ構造を基準として匿名化をする手法の場合，同型グラフを作成した後に類似した構造をもつノードの属性を一般化することを考える．例えば図 2 のグラフにおいて，u1 と u7 のノードは構造が同じである．図 2 は k=2 の k-automorphic graph であるため，u1 と u7 の二つのノード区別がつかなくなるように属性の値を一般化する．二人は 14 歳と 22 歳であるため年齢は 30 歳以下（= under 30），学校へは通っている（= Yes），のように値を一般化した結果を図 5 に示す．u1 と u7 では各ノードの属性値にあまり共通点が無いため，図 5 のように値が抽象的になりすぎてしまう．この場合，マイニングをした際に一般的な結果しか出ず，ソーシャルデータをマイニングする意味が無くなってしまう．

ID	node	age	School	sex	Likes	・・・
u1	v1	14	Ocha J.H.S	F	Jazz	・・・
u7	v7	22	T Univ.	F	running	・・・

⇨

ID	node	age	School	sex	Likes	・・・
u1	v1	under 30	Yes	F	Yes	・・・
u7	v7	under 30	Yes	F	Yes	・・・

図 5　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．例えば k-automorphism で使用されている k-Match アルゴリズムでは，同じグループに属するサブグラフのグラフ構造を同型にすることを目的としている．構造のみに着目し属性の値は考慮されていないため，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場合がある．このような場合について，図 2 のグラフを例に説明する．図 4 の属性値の表より，属性が似ているノード同士の組合せは{u1,u3,u5}，{u2,u4,u6}，{u7,u8}であることが分かる．これらのノード周辺のグラフ構造を一致させつつ，P11 と P12 のグラフが同型になるようにグラフ構造を変化させていく．まず，組合せ内の辺の本数を揃えるため，辺（u3,u8）と辺（u5,u8）を追加する．図 5（a）のグラフでは辺の本数はそろったが P11 と P12 が同型でないため，さらに辺を追加して形を揃える．（図 5（b））すると，追加する辺の数が極端に多くなってしまい，本来あるはずのない辺が大量に追加されたため，マイニングの結果にも誤差が生じてしまうことになる．

我々は，以上の点を考慮しつつ，ソーシャルグラフの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．本稿では上記に述べた二つのアプローチのうち手法 A を元にし，まずは複数の類似したサブグラフを求めたうえでそれらの属性の k-anonymity を適用していく．また，匿名化手法として k-automorphism を採用

Vol.2011-DBS-153 No.22
2011/11/3

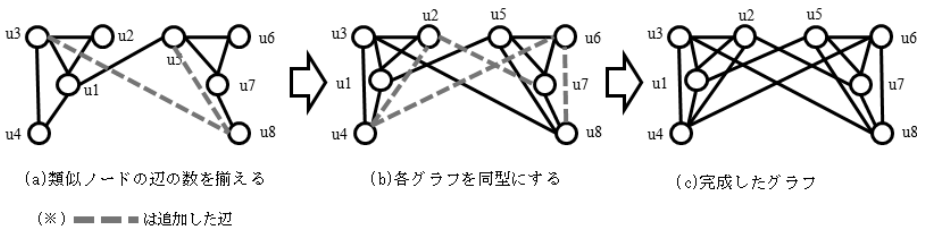


図 6　属性が似ているノードを同じ構造にする

し，これを拡張することでソーシャルグラフの匿名化を行う．

k-automorphism では同型のサブグラフを全て同じグループに振り分けているが，本稿では属性値を考慮してグループ分けをすることによって，より強固なソーシャルグラフ匿名化を実現する．

3.4 k-automorphism におけるグラフの分割と分類

まず，本項で k-automorphism のアルゴリズムについて説明したうえで，このアルゴリズムの拡張方針について次項で述べる．

k-automorphism における k-Match アルゴリズムの概要を図 7 に示す．

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 の識別子を削除し G'とする
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 7　k-Match アルゴリズム

k-Match アルゴリズムでは元のグラフを n 個のサブグラフに分解し，そ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する．サブグラフに対しては 2.1 節で説明したように，グループ内の各サブグラフが同型になるように辺を追加して編集を行う．また，グラフ分割・分類を行うアルゴリズムを図 8 に示す．

アルゴリズム 2：グラフ分割・分解アルゴリズム	
入力：識別子を削除したグラフ 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:	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
3:	求めたサブグラフを 1 つのグループ Ui に入れる
4:	repeat
5:	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(Ui) < Cost(Ui')
7:	Ui のサブグラフを G'から削除する
8:	end

図 8　グラフ分割・分解アルゴリズム

最適なグラフ分割方法を見つけることは NP 完全であるため，グラフの匿名化コストを用いて経験的に導き出されている．まず同型のサブグラフでグループを作成するために，min_sup=k を満たす最大の頻出サブグラフを 1 種類求め，同型のサブグラフを 1 つのグループとする．この時点では，グループ内のサブグラフは全て同型である．この後，サブグラフを 1 ホップずつ拡大する．拡大する理由は，グラフを細かく分割し過ぎてしまうとサブグラフの数も多くなり匿名化コストが高くなるため，できる限り大きなサブグラフに分割したほうが良いからである．ただし，全く同じ構造にならない場合は，この後のグラフの編集を踏まえ，できるだけ辺の追加が多くならないようにする．その方法として 1 ホップ拡張させた場合の辺の編集コストについて定義しており，新しいコストが拡張前のコストを上回った場合には拡張を止めるというアルゴリズムになっている．辺の編集コスト COST(Ui)は以下のように定義されている．

定義 1　サブグラフ Pij(j=1,2,...,k)を持つグループ Ui が与えられていたとき，グループ Ui の匿名化コストは以下のようになる

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

この式中の AlCost(Ui)とはグループ内の各サブグラフの構造を一致させるために追加される辺の数であり，|CrossEdge(Pij)|はサブグラフ間をまたぐ辺の数である．また，$0.5*(k-1)*\sum_{j=1}^{k}|CrossEdge(Pij)|$はコピーして作成された辺の数と一致する．グラフ全体の匿名化コストは，1 グループの匿名化コスト COST(Ui)を合計したものとなる．

G'をサブグラフに分割した後，サブグラフ同型であるグラフはすべて同じグループに分類される．さらにグループに含まれるサブグラフに対して，以下の定義が成り立つ．

定義 2　グラフ G と最小サポート値 min_sup が与えられていたとき，グラフ gf は G の頻出サブグラフという．必要十分条件は，gf と同型なサブグラフが G 内に min_sup 以上存在していることである．

3.5 提案手法

k-Match アルゴリズムでは，同じグループに k 個以上のサブグラフが分類されれば，そのグループはアルゴリズム内で正しいものとされてきた．しかしこのアルゴリズムは，グラフの分類においてサブグラフの構造のみを考慮したものである．そこで我々は，アルゴリズムのこの箇所において各ノード属性の類似度を考慮するよう拡張することとした．

拡張したグラフ分割・分類アルゴリズムを図 9 に示す．

アルゴリズム 2'：グラフ分割・分解アルゴリズム（ノード属性を考慮した場合）

入力：識別子を削除したグラフ 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:      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
3:      求めたサブグラフを 1 つのグループ Ui に入れる
4:      グループ Ui 内の属性値に対して k-anonymity を適用する
5:      グループ Ui のサブグラフを同じ属性値をもつグループ Gj に分類する
6:      for Gj
7:          repeat
8:              Gj グループ内のサブグラフを 1hop だけ拡張させ Gj'に入れる
9:          until Cost'(Gj) < Cost'(Gi')
10:         Gi のサブグラフを G から削除する
11:     end
12: end
```

図 9　グラフ分割・分解アルゴリズム

まず，グループ内に含まれるサブグラフの数を表す min_sup を大きめに設定する．グラフ G'がどのようなサブグラフに分割されるかは，min_sup の値によって決まる．例えば図 10 のようにノード数 5 のグラフが 3 つ含まれているグループ Uh を考える．この場合，図で示されるようなノード数 5 のサブグラフが G'の中に 3 つ存在していると考えられるので，min_sup の値は多くても 3 であるといえる．ここで，min_sup を 5 に変更したとすると，グループ Uh は定義 3.2 を満たさなくなってしまう．そこでグラフ G'をさらに細かく分割し，より細かいサブグラフを 5 つ作成することによって，条件を満たすグループ分けをする．

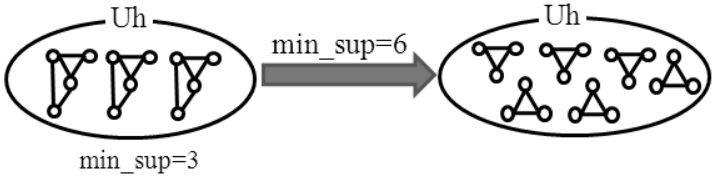


図 10　min_sup の値を大きくした場合

min_sup を大きく設定し，同型のサブグラフが min_sup 以上含まれるグループを作成したら，次に 1 つのグループ内に対する属性リストを作成する．属性リストでは，グループ内に含まれるノードの属性値を 1 行に並べて表示し，1 タプルで 1 つのサブグラフに含まれるノードの属性を表している．属性を 3 個持つノードで構成されるサブグラフは，図 11 のように 3 列で 1 つのノードの属性値を表すことになる．グループ内の各サブグラフにおいて構造的に対応しているノードの属性が同じ列で表されているため，図 11 の表に k-anonymity を適用するということは，同じ構造をもつノードの属性値に k-anonymity を適用していることと同義になる．以上の方法を k-Match アルゴリズムに組み込むことで，同じ構造をもつノードの属性類似度を高めることができる．よって，グラフ構造だけでなく属性値からのノードの絞り込みも防ぐことができたといえる．

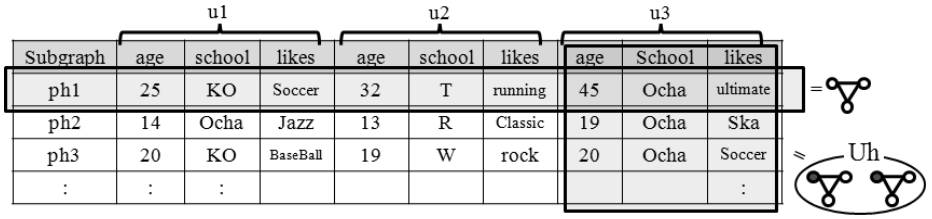


図 11　グループ内サブグラフに対する属性リスト

サブグラフが細かすぎると匿名化コストがかかってしまうため，ここでグラフを拡張していく．従来の手法と同様，コストを考慮に入れてグラフを 1 ホップずつ拡張し

ていくのだが，ここではコスト計算の定義を変更し，属性リストの匿名化コストを考慮に入れるようにする．ノードを 1 つ増やした際に，k-automorphism ではコストが拡張前を上回った場合は拡張を停止するようにしていた．しかし属性値を考慮に入れた場合，それでは拡張自体が難しくなると考えられる．このため閾値を設定し，k-anonymity の匿名化コストがあらかじめ定めておいた閾値以下ならそのノードを採用し，グラフを拡張することとした．匿名化コストは以下のように定義する．

定義 3　ノードの属性数が a, サブグラフ内のノード数が n であるサブグラフの匿名化コスト COST’(s)は以下のように表すことができる．

$$COST'(s) = AlCost(Gj)+0.5*(k-1)*\sum_{j=1}^{k}|CrossEdge(Pij)| + \sum_{l=1}^{an} Attr(l)$$

Attr(l)はリスト中のセル 1 つを匿名化するコストであり，それらをセル数 an 個加算することにより 1 タプルの匿名化コストを表す．Attr(l)は k-anonymity による匿名化で用いられる一般化階層において，一般化した階層の数をコストとして数えている．例えば属性が Likes である場合，このドメインの一般化階層において根のノードは all，子ノードは Sports や Music，Study など大まかなジャンル，そのうちの Sports の子ノードは Soccer，BaseBall といった詳細な競技名というように，階層があらかじめ定義されているとする（図 12）．k-anonymity によって属性を匿名化する際には，元の値に対して匿名化後の値がこの一般化階層においてどれだけ階層が上がったかということをコストとして表示する．Soccer を一般化して Sports にした場合のコストは 1，all にした場合のコストは 2 である．

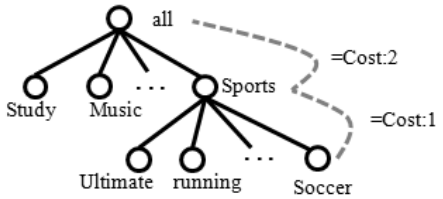


図 12　属性の一般化とそのコスト

定義 3 で示されたサブグラフの匿名化コストを用いて，グループ Gi の匿名化コスト COST’(Gi)を表す．

定義 4　サブグラフが k 個含まれる集合 Gi に対して k-anonymity を適用した際のコスト COST’(Gi)は以下のようになる．

$$COST'(Gi) = \sum_{j=1}^{k} COST'(s)$$

なお匿名化によるコストは同型サブグラフのグループ化を行った後（アルゴリズムの 7 行目）と拡張時（10 行目）にチェックする．まず同型サブグラフのグループ Gj を生成した後にコストを計算し，それが既に閾値を超えていた場合は属性を一般化しすぎているということになるため，そのグループ Gj は使用しない．またサブグラフを 1 ホップずつ拡張する場合も追加したノードに関するコストを計算し，閾値を超えた場合拡張を止める．

本提案手法においては，元にしている k-automorphism に比べて分割されるサブグラフが細かくなることが予測される．例えば，全サブグラフの全属性に対して k-anonymity を適用すると，共通の値を一般化し過ぎてしまい，図 5 のように属性匿名化のコストがかさむ可能性がある．その場合の対応としては，公開する属性の値を制限することによって属性匿名化のコストを抑えることを検討している．マイニングをする側の人間がどの属性値を使用して分析を行うかということを調べ，必要最低限の値のみを公開することによって，k-anonymity を適用すべき属性の数が減り，属性の匿名化コストも低くなる．

4. 関連研究

SNS のプライバシ保護に関する論文は数多く発表されており，それらの多くは既存のデータ匿名化手法を使ってグラフデータの匿名化を行っている．具体的な手法として, Sweeny の提唱したテーブルデータのプライバシ保護手法である k-anonymity**[6]**や，これを発展させた l-diversity**[2]**，t-closeness**[10]**などが挙げられる．l-diversity は機密属性以外の値を一般化し，データの中に少なくとも l 種類の機密属性を存在させる手法である．t-closeness は l-diversity を拡張したものであり，l-diversity において生じる多様性の偏りをなくす方法である．これらの手法は機密属性を守るための非常に強固な手法であるが，リレーショナルデータに対して適用されるものであり，ソーシャルデータの匿名化として使用するには限界がある．

そこでソーシャルデータに対して匿名化を行う際には，グラフ構造を匿名化することに焦点を当てている. Hay ら**[9]**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．Zhou ら**[3]**は k-neighborhood anonymity を提案し，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の存在を最低でも k 個保証するものである．グラフ構造の匿名化に関する論文はユーザ個人の属性値を無視している場合が多く, このことは k-isomorphism**[5]**や k-candidate anonymity**[8]**，本稿で使用した k-automorphism**[7]**にも同様のことが言える．

本稿ではグラフ構造の匿名化に加えて，SNS に参加しているユーザ属性の匿名化も

行った．Zheleva ら**[4]**はソーシャルデータの辺にも意味があることに着目し，機密属性である辺を匿名化，構造的にもリレーショナル的にもソーシャルデータの匿名性を保持した．本手法との相違点は，ノードではなく辺に対して匿名化を行うことで，グラフの構造的にもノード同士の関係的にもグラフの匿名化を行っているという点である．Campan らの手法**[1]**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．匿名化をする際には，構造に関する情報損失や属性を一般化した際に生じる情報損失に注意を払っている．属性値とグラフ構造の両方に対して匿名化を行う手法は提案されているが，本論文で取り扱ったような手法は今まで提案されていない．

5. まとめと今後の課題

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プライバシが侵害される可能性が低くなるような匿名化手法を提案した．本論文ではサブグラフを細かく設定し，サブグラフ全体の属性値を考慮してグラフを成長させていくという手法 A の案を基に拡張を行ったが，今後は手法 B を基にした匿名化手法についても議論していきたい．そして両方の手法について考え，評価実験を行い，本稿で提案されている手法をよりよいものにしていくことを課題とする．

参考文献

1) A.Campan and T.M.Truta: A Clustering Approach for Data and Structural Anonymity in Social Networks, 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) E.Zheleva and L.Getoor: Preserving in social network,A Survey, In Social Network Data Analytics, Chapter 10, pp.277-306, 2011.
5) J.Chemg, A.W.Fu and J.Liu: K-Isomorphism: Privacy Preserving Network Publication against Structural Attacks, Proceedings of the 2010 international coference on Management of data (SIGMOD'10), pp.459-470, 2010.
6) L.Sweeny, :K-anonymity:a model for protecting privacy,Uncertainly,Fuziness and Knowledge-based Systems,Vol.10,No.5,pp.557-550,2008.
7) L.Zou , L.Chen and M. T. O zsu :K - Automorphism : A General Framework for Privacy Preserving Network Publication,In proceedings of the VLDB Endowment, Vol.2, Issue 1,pp.946-957,2009.
8) M.hey, G.Miklau, D.Jensen, P.Weis and S.Srivastava: Anonymizing social networks, Technical report, University of Massachusetts, 2007.
9) M.Hey, G.Miklau, D.Jensen,D.Towsley, and C.Li: Resisting Structural Re-identification in Anonymized Social Networks, The VLDB Journal, Vol.2010, No.19, pp.797-823, 2010.
10) N.Li, T.Li, and S. Venkatasubramanian: t-Closeness: Privacy Beyond k-Anonymity and l-Diversity,InIEEE ICDE ,2006.