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

理学専攻情報科学コース　野澤 佳世

1　はじめに

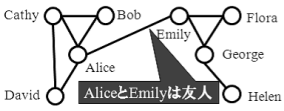
近年ソーシャルネットワークサービス (以下 SNS) が流行しており，そのサービス数も増加している．サービスを提供している企業は，SNS を利用しているユーザが属するコミュニティや個人の行動パターンの傾向を，ソーシャルデータとして収集することができるようになった．SNS は利用しているユーザの趣味や嗜好，普段の生活が反映されていると考えられるため，蓄積されるデータを分析することには多くの有用性が期待される．その反面，ソーシャルデータには個人情報が多く含まれると考えられるため，プライバシの問題が懸念されることになり，公開する際にはデータを匿名化することが求められるようになる．ソーシャルデータに対する従来の匿名化手法は，グラフ形式であるというソーシャルデータの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．

2　従来の匿名化手法と攻撃方法

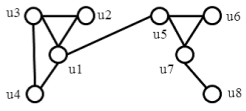
本稿ではソーシャルデータをソーシャルグラフと呼ばれるグラフ構造のデータで表し，その例を図 1 に示す．図 1(a) は SNS 内で各ユーザがもっている属性値と，それらのユーザがどのノードに割り当てられているかを示しており，ノード名とユーザ名に次いで各ノードが持つ属性が表されている．図 1(b) では関係のあるユーザ同士を辺で結び，SNS 内での友人関係を表すソーシャルグラフを形成している．

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

ソーシャルグラフ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ような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 u_1,u_2 といった ID に置き換えているが，攻撃者が Alice に 4 人の友人がいるという背景知識を持っていた場合には，辺の本数より v_1 が Alice であるということが特定されてしまう．そこで，近年提案されているデータ匿名化手法 [2] では，グラフを変化させ同じ構造の部分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している．

それでもまだ個人が敵に攻撃されてしまうことがある．3 ユーザと友人関係にある Bob をグラフ内から特定する際，候補は $\{u_2,u_3,u_5,u_6\}$ の 4 つである．しかし敵が，他の SNS で公開されているプロフィールから,「Bob は学校が KO で趣味が Soccer の 25 歳」という知識を取得すると，候補の中で該当するノードは u_2 のみになる．このように属性の情報を背景知識とすることで，図 2 のように u_2 が Bob であるということが攻撃者に特定されてしまう．

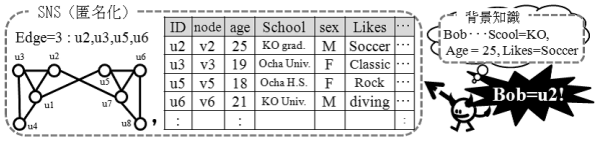


図 2: 属性からユーザ情報が推測されてしまう例

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．そこで我々は従来の手法を拡張し，データ構造と個人の属性の両方に対して匿名化を施すことができるような手法を提案する．

3　提案手法

3.1　グラフ構造と属性値の匿名化手法

本研究では，グラフ構造を匿名化する手法として k-automorphism[2] 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 G*を作成する．G*を作成する k-Match アルゴリズムを以下に示す．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，サブグラフ同型なものが k 個以上存在するグラフ同士をそれぞれ任意のブロックに分類する．次に，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，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．これらの手順を踏んで k=2 で匿名化したグラフは図 3 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．

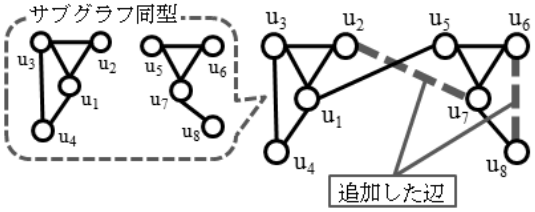


図 3: k-automorphic graph

次に，属性を匿名化することを考える．他の値と組み合わせることで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) といい，このような準識別子の匿名化指標に，k-anonymity[1] があげられる．この手法を適用することによって，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不可能になる．具体的な手法としては，1 つの準識別子に共通

する性質を抽象化し 1 つの概念にまとめて一般化する．前述のソーシャルデータにおける準識別子は Age ,School ,Likes であると考えられ，k-anonymity を適用した結果は図 4 となる．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 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．

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	⋯
u2	v2	20's	KO	M	Sports	⋯
u3	v3	Teen	Ocha	F	Music	⋯
u4	v4	20's	KO	M	Sports	⋯
u5	v5	Teen	Ocha	F	Music	⋯
u6	v6	20's	KO	M	Sports	⋯
u7	v7	20's	T	F	Sports	⋯
u8	v8	20's	T	F	Sports	⋯

3-anonymity
3-anonymity
2-anonymity

図 4: k-anonymity を適用したソーシャルデータ

属性とグラフ構造のそれぞれを匿名化して公開したとしても，十分に匿名性が保たれるとはいえない．敵が，友達が 3 人で 10 代の趣味が Jazz(Music) であるユーザを特定したいとする．図 4 のユーザ属性をのみを見ると候補は $\{u_1,u_3,u_5\}$，図 3 のグラフ構造を見ると候補は $\{u_1,u_7\}$ であり，これらに共通しているノード，つまり敵が特定したいユーザは u_1 である．これより，グラフ構造と属性値を別々に匿名化すると，それらを組み合わせて攻撃した場合には匿名化が意味をなさなくなってしまうということが分かる．そこで我々は，属性値も考慮に入れてグラフ構造の匿名化を行うこととした．

本稿では，ブロック分割を行う際にサブグラフ同型であるだけでなく，各ノードの属性が類似したサブグラフ同士でブロックが構成されるように k-Match アルゴリズムを拡張することとした．また，本手法を適用する上で，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，データ分析をした際の有用性を失ってしまわないよう注意する．

3.2 属性値を考慮したグラフ分割

前節まではグラフ構造のみを利用して同型であるか判別していたが，提案手法では属性も一致したサブグラフのみで同一のグループを形成することを目標とするため，k-Match アルゴリズムを以下のように拡張する．

まず，与えられたソーシャルデータに k-anonymity を適用し，属性を匿名化する．次に，ノードの属性値を考慮した部分グラフのブロック分けを行う．ブロック分けには，SEuS[3] というノードラベルを考慮した頻出サブグラフ導出法を用いる．SEuS を適用するために，属性値に従ってノードにラベルを付与する．例えば，図 4 の u_1,u_3,u_5 は属性値が同じであるため，これらのノードには a という同一のラベルを付与する．全てのノードにラベルを付与したグラフを，図 5 に示す．従来のアルゴリズムでは，図 5 のグラフからは $\{u_1,u_2,u_3,u_4\}$ と $\{u_5,u_6,u_7,u_8\}$ が同型グラフとして導き出されていた．提案手法ではノードの値も考慮に入れて頻出サブグラフを導き出すため，$\{u_1,u_2,u_3\}$ と $\{u_1,u_3,u_4\}$，$\{u_1,u_2,u_5\}$ と $\{u_1,u_5,u_6\}$ のそれぞれが同型サブグラフとして抽出される．

グループが一つ導き出されたら，元のソーシャルグラフを複製することによって作成した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

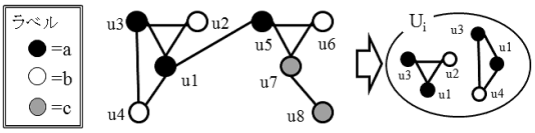


図 5: ラベルつきグラフ G

辺を削除する．そして，これによって生じた孤立点を削除する．こうして作成されたグラフを G" とし，図 6 で表す．G" のままだと頻出サブグラフを導き出すことはできないが，k の値を増やして k-anonymity を施して u_5 と u_8 の属性値を類似させ，ラベルを a' に置き換えることにより $\{u_5,u_6,u_7\}$ と $\{u_8,u_6,u_7\}$ が同型グラフとして検出されるようになる．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．

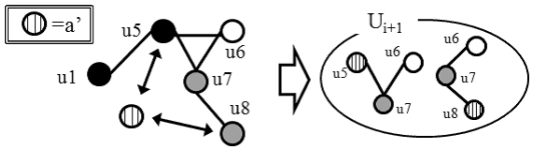


図 6: 2 回目以降の繰り返し結果

提案手法の問題点として，n 回目の繰り返しで導き出された頻出サブグラフ集合に含まれるノードの値が，n+1 回目で違う値になってしまうということが挙げられる．図 6 において 1 回目の繰り返しで導きだされた u_5 のラベルは a であるが，2 回目では u_5 のラベルは a' として検出される．この問題の対策として 2 つの案が考えられる．1 つは頻出サブグラフ集合を導き出した後にエッジだけではなくノードも削除する方法である．もう一つはすでにサブグラフ集合として検出されたノードがその後の繰り返しで異なったラベルを付与された際，すでに検出されているサブグラフ集合内のノードのラベルも付け替えるという方法である．

4 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．本稿では属性を匿名化してから分割するという手法を採ったが，作成されたブロックの中から属性が似ているサブグラフを抜き出し，それらを使って新たにブロック分けをするという手法も考えられる．今後は互いのトレードオフを考えて二つの手法を組み合せ，ソーシャルデータの匿名化に適した手法を発見することが課題である．

参考文献

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

[2] 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.

[3] S.Ghazizadeh, and S. Chawathe: "SEuS: Structure Extraction using Summaries," In Lecture Notes in Computer Science, vol.2534,pp.71-85,2002.