Vol.2017-AL-162 No.4
2017/3/13

データ研磨によるバイクラスタマイニング

宇野 毅明[1,a)]　小池 敦[2]　中原 孝信[3]　羽室 行信[4]

概要：バイクラスタは、2 部グラフ $G = (U, V, E)$ に含まれる枝密度の高い部分グラフのことであり、その頂点集合 U', V' の頂点は互いに強く関係している、あるいは相関があると考えられる。たとえばそのグラフが顧客と購入した商品の関係を示したトランザクションデータベースを表しているならば、U' は V' の商品群の商品を多く購入するという性質を持った顧客のグループであると考えることができる。この例を始め、バイクラスタは非常に多くの場面に応用を持つ。比較的小さなバイクラスタをグラフからすべて見つける問題はバイクラスタマイニングとよばれ、データマイニングの中でも重要な問題の一つである。しかし、バイクラスタマイニングは一般に難しく、決定的なアルゴリズムは存在していない。本稿では、著者らが提案したデータ研磨のアプローチをバイクラスタマイニングに適用し、新しいアルゴリズムを提案する。データ研磨は、実行可能仮説に基づいてデータを局所的に変更して揺らぎを除去し、隠された構造を明確化するというアプローチである。本稿では、バイクラスタに含まれるべき枝をその周辺に存在する枝の状態から特徴付けし、それに基づいて 2 部グラフを明確化する手法を提案する。計算実験では、提案手法は非常に高い精度でバイクラスタを見つけることができること、他の手法に比べてはるかに高い精度を持つことが示された。

キーワード：データクリーニング，データ解析，頻出集合，パターンマイニング，バイクラスタ

Bicluster Mining by Data Cleaning like Approach

Abstract: A bicluster is a dense subgraph of a bipartite graph $G = (U, V, E)$, and its vertex sets U' and V' are considered to relatively relate, or corelate to each other. When the graph represents the inclusion relation in a transaction database each whose record is a busket of a customer composed of the items purchased by the customer, U' and V' would imply a group of customers having similar favorites, and V' is a group of items purchased by the customers relatively many. Biclusters have many applications in science and industry. The problem of exhaustly finding the biclusters from a bipartite graph is called bicluster mining. Although the bicluster mining is one of the main problems in data mining, there is no standard algorithm proposed because of its difficulty. In this paper, we propose a new approach based on the data polishing the authors proposed to clustering problems. The concept of data polishing is to clarify the structures in the data by modifying the data locally according to a feasible hypothesis. The approach of this paper is to characterize the edges that should be included in a bicluster with the edges surrounding the edge, and make a new graph according to the characterization. The computational experiments show the efficiency of our method so that our method drastically outperforms the existing methods.

Keywords: data cleaning, data analysis, frequent itemset, pattern mining, bicluster

1. はじめに

バイクラスタリング (biclustering) は、データマイニングの中でも中心的な問題である。2 つの種類のものの集合、たとえば人の集合とものの集合、を考える。このとき、人とものとの関係、例えば「好き」という関係、を考え、各人に対して、その人が好きなものが定まっているとする。こ

[1] 国立情報学研究所
東京都千代田区一ツ橋 2-1-2
[2] 東北大学
宮城県仙台市荒巻青葉 6-3
[3] 専修大学商学部
神奈川県川崎市多摩区東三田 2-1-1
[4] 関西学院大学経営戦略研究科
兵庫県西宮市上ヶ原一番町 1-155
[a)] uno@nii.ac.jp

のとき、この関係は行列や 2 部グラフとして表現できる。具体的には、人を行、ものを列と見なした行列で、(i, j) 要素が 1 であるとき、人 i はもの j が好きなように各セルの値を設定する、あるいは各人とものを頂点だと考え、人 i がもの j を好きなときに頂点 i と j の間に枝をはる、といった形で表現できる。ここでは、データマイニングの用語にあわせ、人の代わりにトランザクション、ものの代わりにアイテムを用いる。トランザクションはアイテム集合の部分集合であり、トランザクション T とアイテム i の間に関係があることを、T が i を含むこととして定義する。このとき、トランザクションの集合はトランザクションデータベースとよばれる。トランザクションとアイテムをグラフの頂点と見なし、トランザクション T がアイテム e を含むときに頂点 T と頂点 e を枝で結ぶ。このようにしてできるグラフをこのトランザクションデータベースのグラフ表現とよぶ。

与えられたトランザクションデータベースに対して、トランザクション集合 X とアイテム集合 Y がバイクラスタであるとは、X の各トランザクションが多くの Y のアイテムを含むこととする。つまり、多くの X のトランザクションと多くの Y のアイテムの間に関係があるということである。多く含む、ということに対する明確な定義の与え方によってバイクラスタの定義は異なってくる。また、他者との関係性をバイクラスタの定義に含むこともある。例えば、X に含まれない任意のトランザクションは、Y のアイテムを少数しか含まない、あるいは、Y に含まれないアイテムは、X のトランザクションに少ししか含まれない、といった条件付けをすることもある。例えば、X の任意のトランザクションが Y の任意のアイテムを含むこと、つまり任意の $T \in X$ に対して $Y \subseteq T$ が成り立つことをバイクラスタの定義とすると、バイクラスタは、グラフ表現の 2 部クリークに対応する。これに、X 以外のトランザクションは、Y のいずれかのアイテムを含まない、という条件をつけると、Y はアイテム集合とその出現となり、Y に含まれないアイテムは X のいずれかのトランザクションに含まれない、という条件をつけると、X は飽和集合 (closed itemset)、あるいはフォーマルコンセプトに対応し、X と Y の組はグラフ表現の極大 2 部クリークに対応する。以後、トランザクションデータはグラフ表現で与えられているとする。

X と Y が誘導する部分グラフの枝密度をバイクラスタの条件付けに使うこともある。密度がある閾値を超えた場合、頂点 v と頂点集合 K に対して、$N_K(x)$ を v が隣接する K の頂点の集合とする。バイクラスタの条件として Y の任意のアイテム v が、$|N_K(v)| \geq \theta|X|$ であるというような条件付けが考えられる。また、$|N_K(v)|$ が大きな頂点が Y に含まれていれば加点、含まれていなければ減点とした評価値を考え、これが大きいかどうかでバイクラスタ性を計る方法も考えられる。これは、グラフクラスタリングにおけるモジュラリティと同じような考え方に基づいている。このモジュラリティににた評価尺度や、バイクラスタに含まれない枝数を最小にするような評価尺度を用いて、トランザクション集合とアイテム集合を分割してバイクラスタの集合を求めるような、最適化型のアプローチも考えられる。

以上のように、トランザクションデータからバイクラスタを求めるには様々なアプローチが考えられるが、現在のところ、決定的な手法は存在しないと言っていいだろう。通常のクラスタリングに対して、Girvan-Newman 法 [2]、グラフカット?、k-means[8]、クリーク列挙 [9]、ビリーフプロパゲーション [3], [4] などの方法が標準的に用いられていることに比べると、差は明確である。多くの研究がある一方で、実装が配布されているものは少なく、現実問題での利用を通じた有効性の確認は行われていないと考えてよいだろう。一方、飽和集合の列挙 [5], [10], [11] に関しては、多くの、非常に効率性の高いアルゴリズムとその実装が提案されている。現実問題において検証がなされており、コンピューターの性能の限界に近いレベルでの最適性が確認されている。しかし、計算的な面での効率が高い一方で、飽和集合を列挙するというアプローチには、非常に大量のバイクラスタ（の候補）が出力され、飽和集合を使ったデータ解析の利便性を大きく損なっているという側面もある。これは、本来バイクラスタであるべきものが飽和集合となっているとは限らず、いくつかのアイテムを含んでいない（欠損）トランザクションが存在するため、その指数的に大量に存在する部分集合が飽和集合となってしまうからである。極大クリークではなく、極大な密部分グラフを出力するという方法も考えられ、効率的なアルゴリズムも構築できるが [12], [13]、かえって解の数は増大してしまう。もし、サンプリングのようなテクニックでこれを解決しようとすると、このような欠損が多いものはよく見つかり、欠損が少ないものは見つかりにくいという、本来求めているものとは逆の事象が発生してしまう。出てきた大量の飽和集合をクラスタリングするアプローチも考えられるが、通常飽和集合の数はトランザクション数より遙かに大きいため、問題が巨大になりすぎてしまう。また、バイクラスタリングの多くが、ソフトクラスタリング的な性質を持つ、つまり 2 つのバイクラスタが、同一のアイテム、または同一のトランザクションを共有することが、現実的に考えて素直にあり得る、ということもある。このため、単にアイテムとトランザクションを分割するだけでは問題が解けるわけではない。このような難しさが解決できていないため、バイクラスタリングには決定的な手法が存在しないと考えられる。

本論文では、バイクラスタリングに対して、データ研磨を用いた新しいアプローチによる手法を提案する。我々

は、上記のような難しさが発生するのは、バイクラスタがデータの中で明確でないことに起因すると考える。上述した欠損が多数あるために、バイクラスタが 2 部クリークのようにモデル化しやすいものではない。そのため、モデル化は直接バイクラスタの本質を表すものではないものを含むこととなる。モジュラリティのような尺度を考案しても、欠損の少ないバイクラスタと欠損が多いバイクラスタが重なり合う場合などに、欠損が少ないクラスタに、欠損が多いバイクラスタの一部が組み込まれてしまったり、両者が一つのバイクラスタに合併されることがありうる。これは、モジュラリティを用いたクラスタリング手法である Girvan-Newman 法が巨大クラスタや多量の微細なクラスタを生成することを通じて観察される。そのため、我々はトランザクションデータベースを直接バイクラスタリングすることはせず、まず含まれるバイクラスタを明確にすることを考える。つまり、入力したトランザクションデータベースに対して、含まれるバイクラスタがすべて 2 部クリークを誘導するようになり、バイクラスタが誘導する部分グラフに含まれない枝は消去されているようなものを作る。このようなグラフであれば、極大 2 部クリークの列挙をしても解の爆発はなく、モジュラリティを用いても望まない分割や合併は起こらないだろうと考えられる。このような発想に基づいてデータを改変する手法を総称してデータ研磨と呼ぶ。データ研磨は一般のグラフにおけるクラスタリングの問題において、非常に高い精度と計算効率性を持つことが示されている 。

我々は、グラフのそれぞれの枝がバイクラスタに含まれるかどうかを周辺のグラフ構造から推定する方法をモデル化し、それを用いたデータ研磨アルゴリズムを設計し、実装した。ランダムに発生させたデータに対する計算実験の結果、ある程度の大きさを持つバイクラスタはデータ研磨によって精度よくクリークにされることが確認された。また、バイクラスタの重なりから小さいバイクラスタがある程度の量生成されることもわかったが、もとのバイクラスタが十分大きければ、両者には明確な大きさの違いがあることも確認された。

2. 記法

$E = \{1, \ldots, n\}$ をアイテムの集合とする。トランザクションはアイテムの部分集合であり、トランザクションデータベースは、トランザクションの集合である。トランザクションデータベースに対して、そのグラフ表現を以下のグラフで定義する。頂点はデータベースのトランザクションとアイテムであり、トランザクション T がアイテム e を含むとき、またそのときに限り、T と e の間に枝があるようなグラフとする。以下、トランザクションデータベースはそのグラフ表現であるとみなす。

グラフの頂点部分集合 U に対して、U の誘導部分グラフを、両端点が U の頂点である枝を集めてできるグラフとする。U の誘導部分グラフが完全グラフになっている、つまり U の任意の 2 頂点の間に枝があるときに、U はクリークであるとする。また、U が 2 つお頂点集合 U_1 と U_2 に分解され、U_1 の頂点と U_2 の頂点の間には必ず枝があり、それ以外の頂点の組には枝がないとき、U は 2 部クリークであるという。クリーク、2 部クリークで、他のクリーク、他の 2 部クリークに含まれないものを極大クリーク、極大 2 部クリークという。

グラフの頂点 v に隣接する頂点を v の近傍とよび、その集合を $N(v)$ と表記する。頂点集合 K に対して、v に隣接する K の頂点の集合を $N_K(v)$ と表記する。つまり、$N_K(v) = K \cap N(v)$ である。バイクラスタは、頂点集合 X と Y の組 (X, Y) で表現する。バイクラスタ (X, Y) の**密度**を、$X \cup Y$ で誘導される部分グラフの枝数の、同じ大きさの完全二部グラフの枝数に対する割合とする。つまり、$(X \cup Y$ で誘導される部分グラフの枝数 $)/(|X| \times |Y|)$ とする。

3. データ研磨

データ研磨の目的は、データに含まれる構造を明確にすることである。構造がバイクラスタであれば、何らかのバイクラスタに含まれる頂点対には必ず枝があり、そうでければ枝がないような 2 部グラフを構築することが目的である。より正確に言えば、与えられた 2 部グラフ $G = (U, V, E)$ がバイクラスタ $(X_1, Y_1), ..., (X_h, Y_h)$ を持つのであれば、$(u, v) \in E$ であれば、またそのときに限り、$u \in X_i, v \in Y_i$ がある i について成り立つ、というグラフを構築することが目的となる。ただし、このようなバイクラスタは明示的に存在するわけではなく、また正解が背景知識として存在するものではなく、あくまで推定するものであることを注意しておく。データ研磨の基本的な考え方は、データのそれぞれの部分に対して、その部分の値を周辺情報から推測して修正するというものである。その部分そのものの値だけでなく、周辺の情報を見ることによって、構造を明確にする意味で値を安定的にする。2 部グラフの場合は、トランザクション T とアイテム e の任意のペアに対して、枝 (T, e) がなんらかのバイクラスタに含まれるかどうかを、周辺情報から推測して判断しなければならない。そこでまず、以下の観察を行う。

観察 1 ： 2 つのトランザクション T_1 と T_2 が同一のバイクラスタ (X, Y) に属するなら、T_1 と T_2 はいくつかのアイテムを共通して含む。つまり $|N(T_1) \cap N(T_2)|$ がある程度の大きさを持つ

T_1 と T_2 が同一のバイクラスタ (X, Y) に属するなら、T_1 も T_2 も Y のアイテムをある程度多く含むであろう。ならば、その中のいくつかは、T_1 と T_2 に共通して含まれるであろう、という考え方である。このような、現実で

は多くの場合に成り立つであろうと考えられる仮説を**実行可能仮説**とよぶ。もちろん、この仮説は常になりたつものではなく、あくまでこのようになっていることが多いだろう、というものである。また、Y の大きさ、バイクラスタ (X,Y) の密度や大きさ、枝の分布の偏りによって、どの程度のアイテムを共有するかは異なってくる。これは、データインスタンスによって異なるもので有り、その設定にはある程度の背景知識が必要となるたぐいのものである。

また、観察 1 から次の観察 2 が導かれる

観察 2 ：トランザクション T がバイクラスタ (X,Y) に属するなら、T は X の多くのトランザクション T' と、いくつかの共通するアイテムを含む。つまり $|N(T)\cap N(T')|$ がある程度の大きさを持つトランザクション T' が X に多く含まれている。

以上の議論は、トランザクションをアイテムに置き換えても成り立つ。つまり、

観察 3 ：2 つのアイテム e_1 と e_2 が同一のバイクラスタ (X,Y) に属するなら、e_1 と e_2 を同時に含むトランザクションがいくつか存在する。つまり $|N(e_1)\cap N(e_2)|$ がある程度の大きさを持つ

が得られる。以後、$G=G_0$、$G_0=(U,V,E_0)$ とし、閾値 θ に対して、$S_G(T)=\{T' \mid T\in U, |T\cap T'|\geq\theta\}$ と定義する。$S_G(T)$ は T を含むことに注意しておく。また、観察 2 から、枝がバイクラスタに含まれる条件付けを以下のように行える。

条件 1 ： 枝 (T,e) がいずれかのバイクラスタに含まれるなら、閾値 k に対して、$|S_G(X)|\geq k$ が成り立つ。

条件 1 は、研磨後の 2 部グラフが枝 (T,e) を含むべきかどうかを推測する上で十分のように感じられるが、実は不十分である。条件 1 が成り立たなければ、枝が存在しない、という方向では、条件 1 は十分と考えられるが、逆方向、つまり条件 1 が成り立つようなペアは枝を持つべきである、という方向に関しては、大きな落とし穴がある。

グラフ G がバイクラスタ (X_1,Y_1) と (X_2,Y_2) を含むとする。ただし、$X_1\cap X_2=\emptyset$ と $Y_1\cap Y_2=\emptyset$ とする。このとき、任意の頂点対 $T\in X_1, e\in Y_2$ に対して、上記の条件が成り立つ。つまり、(X_1,Y_2) はバイクラスタと認識されてしまう。

(X_1,Y_2) は、たとえ 1 つも枝を含んでいなくても、バイクラスタと認識されてしまう。これを回避するためには、(X_1,Y_2) が多くの枝を含む場合にのみバイクラスタとして認識されるようにする必要がある。そのための条件付けを行うため、以下の観察を行う。

観察 4 ：枝 (T,e) がバイクラスタ (X,Y) に含まれるなら、$(T,e'), e'\in Y$ は枝である可能性が高い

観察 4 と、条件 1 を合わせると、下記の条件が導かれる。

条件 2 ： 枝 (T,e) がいずれかのバイクラスタに含まれるなら、閾値 k に対して $(T',e)\in E$ であるトランザクション $T'\in S_G(T)$ が k 個以上存在する。

以上の議論は、アイテムとトランザクションを入れ替えても成り立つことを注意しておく。さらに、現実的な仮定として、「各トランザクションは、少数のバイクラスタにしか含まれない」というものを考える。これは、現実の多くの場面で見られる自然な仮定である。これの上では、$|T\cap T'|\geq\theta$ は、T と T' の類似度 $sim(T,T')$ が閾値 θ 以上である、という条件に置き換えることもできる。T が非常に多くのアイテムを含む場合、$|T\cap T'|\geq\theta$ はほぼすべての T' に対して成り立ってしまう。このようにサイズが大きなトランザクションが存在する場合などは、類似度を用いたほうが精度良くバイクラスタが見つけられると考えられる。以後、類似度を用いた場合は $S_G(T)$ は $S_G(T)=\{T' \mid T'\in U, sim(T,T')\geq\theta\}$ で定義する。

$G_0=(U,V,E_0)$ を $G_0=G$ とし、グラフ G_i に対して、S_{G_i} を簡単に S_i と記述する。条件 2 によって、グラフ G_i から、グラフ $G_{i+1}=(V,E_{i+1})$ を、以下のように定義する。

$$E_{i+1}=\{(T,e) \mid T\in U, e\in V, |N(e)\cap S_i(T)|\geq k\}$$

G_1 は、G_0 の頂点対で、何らかのバイクラスタに含まれると考えられるものが枝で結ばれているようなグラフを、条件 2 を用いて定義したものである。つまり、G_0 上で条件 2 を満たす頂点対に枝が張られているグラフが G_1 である。条件 2 がある程度妥当であれば、G_0 がノイズを多く含んでも、G_1 のノイズは少なくなると考えられる。同様にして、G_1 から G_2 を作れば、さらにノイズは減ると考えられる。最終的に、$G_{i+1}=G_i$ となる i が存在すれば、それはノイズがきれいに除去された状態と考えて良いだろう。このようなグラフ G_i を、G を**研磨した**グラフとよぶ。

4. アルゴリズム

以下に、研磨したグラフを求めるアルゴリズムを記述する。アルゴリズムが収束せず、G_i が存在するとは限らないため、繰り返しは最大 τ 回までとし、その場合はグラフ G_τ を出力する。また、一般性を高めるため、上記の条件 $|N(e)\cap S_i(T)|\geq k$ を $sim'(N(e),S(T))\}|\geq k$ で置き換えている点に注意されたい。

Algorithm 2 部グラフ研磨 ($G=(U,V,E)$: 2 部グラフ)
1. **for** $i:=0$ to τ
2. $\quad E':=\{(T,e)|T\in U, e\in U, v\in V, sim'(N(e),S(T))\geq k\}$
3. $\quad$ **if** $E=E'$ **break**
4. $\quad E:=E'$
4. **end for**
5. **output** $G=(U,V,E)$

このアルゴリズムは、$S(T)$ をすべての T に対して計算するのに最大 $O(|U|^2|V|)$ の時間を要し、ステップ 2 に 最大 $O(|U|^2|V|)$ の時間を要する。そのため、計算量

は $O(\tau|U|^2|V|)$ となる。これは、大きなグラフに対しては絶望的な計算量であるが、一般にグラフが袖ある場合には非常に高速になる。一般のグラフに対するデータ研磨アルゴリズムを用いて $S(T)$ を系算すると、V の頂点の次数の分布がべき乗則に従う場合、計算時間は $O(|E|+\Delta^2)$ となる。ここで Δ は V の頂点の最大次数である。$S^{-1}(T)=\{T'|e\in S(T')\}$ とすると、同様にステップ 2 の計算も $|N(T)|+|S^{-1}(T)|$ がべき乗則に従うのなら、$O(|E|+(\max_{T\in U}\{|N(T)|+|S^{-1}(T)|\})^2)$ の計算時間で終了する。現実的には、多くの実データがべき乗則に従い、巨大で密なデータはほぼ存在しないため、このアルゴリズムは実データに対して多くの場合高速であるとみなして良いだろう。

5. 計算実験

提案したアルゴリズムの評価を行うため、人工的に生成したデータと実データに対して計算実験を行った。本節では、その実験内容と結果を解説する。

まず、人工的に生成したデータの作り方は、以下の通りである。まず、空の 2 部グラフ $=(U,V,E), E=\emptyset$ を用意し、ランダムにバイクラスタを生成する。$|U|=|V|=n$ であるとする。バイクラスタは、U の頂点を x 個、V の頂点を y 個ランダムに選び、それらが 2 部クリークになるように枝を張る。x と y の選び方は、(10,10), (20,20), (10,30), (30,10) の 4 種類を試した。特定の頂点が何回も選ばれることがないように、1 つの頂点がバイクラスタの頂点として選ばれる回数を高々 b 回とし、b を変化させていくつかのグラフを作成した。バイクラスタ数は、各頂点がバイクラスタに含まれる確率が高くなるよう、$n/20*1.7$ とした。頂点集合 V,U はそれぞれ 10000 とした。　実験に用いたグラフは、空のグラフに G、この各バイクラスタに対応する頂点集合が完全 2 部グラフになるように枝を加えた後、ノイズとなる枝を取り除き、あるいは加えて得られたものである。まず得られたグラフの各枝を一定の確率 p で取り除き、次に U の各頂点 u に対して、(u の次数) $\times\, p/(1-p)$ 個の頂点を V からランダムに選び、u とつなぐ。これにより、u の次数は $1+p/(1-p)$ 倍になる。このようにノイズを追加する理由は、多くのノイズ枝に接続し、バイクラスタに含まれる枝に少ししか接続しないような、もともと発見が難しい頂点が生成されるのを防ぎ、かつバイクラスタに含まれる多数の枝に接続するが、あまり多くのノイズ枝に接続しない、発見が容易と考えられる頂点が生成されるのを防ぐためである。本稿では、U,V の大きさがであり、p が であるグラフを個生成し、それぞれに対して実験を行った。実験の結果は、表にまとめた。データ研磨アルゴリズムは、類似度指標として Jaccard 係数を用い、閾値をそれぞれ 0.2, 0.1, 0.07 とした。表の精度は、それぞれの正解クラスタに対して、発見されたクラスタの中で一番 Jaccard 係数が大きくなったものを選び、その平均をとったものであり、平均的にどれくらい精度高く各クラスタが発見されたかを示している。また、その隣に発見されたクラスタ数を記述した。まず、ノイズが大きくなると、発見されるクラスタ数が大きくなる傾向がある。特に多重度が大きな場合に顕著であるが、これはクラスタが重なると、それらの重なりが大きさの異なる極大 2 部クリークを導出してしまうことによるものが大きい。しかし、これらのクラスタは、U あるいは V 側の頂点数が極端に小さく、3 や 4 であることが多いため、実用上は問題にならない。また、閾値が大きめ場合には、ノイズが増えると発見されるクラスタ数が少なくなり、同時に精度が極端に小さくなる。これは、類似する行が少ない行が多くなり、それらは研磨によりデータ上から消えてしまう。研磨は耐えられないほどのノイズがある場合、とたんに精度が落ちてしまうことがわかる。逆にノイズが耐えられるほどの範囲であれば、非常に高い精度となることがわかる。他手法の実験も試みたが、極端に精度が低かったり、あるいは計算時間がかかりすぎ 5 時間以上たっても終わらないことなどあり、今回は十分な結果が示せなかったが、データ研磨は少なくとも 10000 程度の大きさのデータでは、30%ほどのノイズであれば非常に正確にもとのクラスタを抽出できることがわかった。

6. まとめ

本稿では，バイクラスタを網羅的に見つけるマイニング問題に対して、データ研磨アプローチによる、極大 2 部クリークの明確化を行う手法を提案した。同じバイクラスタに含まれる頂点は、ある程度の類似性をそれらの近傍集合に持つことを基本の観察とし、頂点対 (x,y) がバイクラスタに含まれることの実行可能仮説としての特徴付けを、x と近傍集合が似ている頂点の多くが y と隣接することとして行った。計算実験の結果、人工的に生成したデータにおいて、バイクラスタがある程度の大きさがあれば、ノイズを負荷された状態でも非常に高い精度でバイクラスタを極大 2 部クリークとして復元できることを確認した。一方、既存手法ではほとんどバイクラスタが発見できないことも確認した。今後は、小さいバイクラスタの研磨をどのようにすればよいか、小さいバイクラスタがノイズ的に発生してしまう問題をどのように解決すれば良いか、という問題の解決や、違った方向からの研磨手法、一般のグラフに含まれる極大 2 部クリーク構造の研磨などの課題が考えられる。

謝辞

この研究は科学技術振興機構 CREST 「ビッグデータ統合利活用のための次世代基盤技術の創出・体系化」の補助を受けている.

表 1 多重度 $b = 1$ の場合

	研磨	0.1	研磨	0.2
クラスタの大きさ, p	精度	クラスタ数	精度	クラスタ数
10×10, 0.2	0.8235	1565	0.9998	700
20×20, 0.2	0.9981	388	1	350
10×30, 0.2	0.9996	1502	0.9997	233
30×10, 0.2	0.7449	1198	0.9990	233
10×10, 0.3	0.7874	1971	0.9982	700
20×20, 0.3	0.9986	439	1	350
10×30, 0.3	0.9982	2920	0.9988	233
30×10, 0.3	0.7609		0.9967	233
10×10, 0.5	0.8113	1491	0.4428	624
20×20, 0.5	0.9927	350	0.0997	154
10×30, 0.5	0.9891	4860	0.8086	269
30×10, 0.5	0.8283	568	0.2416	200

表 2 多重度 $b = 2$ の場合

	研磨	0.1	研磨	0.2
クラスタの大きさ, p	精度	クラスタ数	精度	クラスタ数
10×10, 0.2	0.9607	7329	0.8244	3988
20×20, 0.2	1		0.8326	2754
10×30, 0.2	0.9927	2955	0.9866	1662
30×10, 0.2	0.9654	6965	0.8468	3747
10×10, 0.3	0.9446	7175	0.5122	1367
20×20, 0.3	0.9980		0.4093	655
10×30, 0.3	0.9977	3131	0.9041	
30×10, 0.3	0.9774	6774	0.5289	631
10×10, 0.5	0.1466	2283	0.0324	631
20×20, 0.5	0.2203	437	0.0246	631
10×30, 0.5	0.8694	1181	0.0578	135
30×10, 0.5	0.4945		0.0185	125

表 3 多重度 $b = 4$ の場合

	研磨	0.1	研磨	0.2
クラスタの大きさ, p	精度	クラスタ数	精度	クラスタ数
10×10, 0.2	0.7442	11256	0.0189	98
20×20, 0.2	0.7746	23330	0.0072	50
10×30, 0.2	0.9883	27140	0.4949	1308
30×10, 0.2	0.8550	16520	0.0811	379
10×10, 0.3	0.2208		0.0002	2
20×20, 0.3	0.3103	3008	0.0004	2
10×30, 0.3	0.9326	13328	0.2399	858
30×10, 0.3	0.6148	6627	0.0010	8
10×10, 0.5	0.0005	3	0.0015	8
20×20, 0.5	0.0007	3	0.0015	8
10×30, 0.5	0.1432		0.0022	8
30×10, 0.5	0.0002	2	0.0011	

参考文献

[1] C. Cortes and V. N. Vapnik, Support-Vector Networks, Machine Learning **20** (1995).
[2] M. Girvan and M. E. J. Newman, Community Structure in Social and Biological Networks, National Academy of Science USA **99**, pp. 7821–7826 (2002)
[3] P. Judea, Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach, 2nd National Conference on Artificial Intelligence, pp. 133–136 (1982).
[4] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, (2008).
[5] B. Goethals, the FIMI repository, `http://fimi.cs.helsinki.fi/` (2003).
[6] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, Trawling the Web for Emerging Cyber-Communities, 8th International World Wide Web Conference (1999)
[7] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/
[8] J. B. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967).
[9] K. Makino and T. Uno, New Algorithms for Enumerating All Maximal Cliques, Lecture Notes in Computer Science **3111** (SWAT 2004), pp. 260–272 (2004).
[10] T. Uno, T. Asai, Y. Uchida, H. Arimura, An Efficient Algorithm for Enumerating Closed Patterns in Transaction Databases, Lecture Notes in Artificial Intelligence **3245**, pp. 16–31 (2004).
[11] T. Uno, M. Kiyomi, H. Arimura, LCM ver. 2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets, IEEE ICDM 2004 Workshop FIMI'04 (2004).
[12] T. Uno, H. Arimura, Ambiguous Frequent Itemset Mining and Polynomial Delay Enumeration, Lecture Notes in Artificial Intelligence **5012** (PAKDD 2008), pp. 357–368 (2008).
[13] T. Uno, H. Arimura, An Efficient Polynomial Delay Algorithm for Pseudo Frequent Itemset Mining, Lecture Notes in Artificial Intelligence **4755** (DS 2007), pp. 219–230 (2007).