

疑似独立集合制約と正規化カットを用いた グラフの構造比較

間澤 直寛^{1,a)} 原口 誠^{1,b)} 富田 悦次^{2,c)}

概要: ソーシャルネットワーク等において時間の経過や地域による差を調べるために、ノーマライズカットを用いたグラフの比較法を提案する。グラフを比較する際、モジュラリティや k -Plex など差異を示す頂点集合を検出する方法が提案されているが、サイズの大きな頂点集合の検出には適していない。本稿では、比較対象の2つのグラフのうち、一方のグラフに対しては疑似独立集合（補グラフにおける k -Plex）制約を課す。他方、もう一つのグラフに対しては、正規化ラプラシアンによって得られる距離空間において局所密度を考慮した形でグラフを形成し、そのグラフにおいてクリークをなすことを要請する。 k -Plex もクリークも共に単調性を持つことから、2つの制約を同時に満たす解頂点集合を高速に列挙できるエンジンをクリーク全列挙エンジンに習って設計できることを示す。

1. はじめに

目まぐるしく変化する情報の世界においては、変化および変化の兆しや地域により注目されるトピックの違いを検出することは重要なタスクであると考えられる。顕著な差異を検出するシステムとしては、emerging pattern, contrast set, バースト解析など様々なアプローチがあるが、顕著ではない差異をも検出できる研究は未だ十分ではない。偶発的な差異も差異としては極めて多数存在するからである。偶発的なものを排除し、ある程度の必然性を持つものに絞りこむことが重要である。

本稿では、ソーシャルネットワークに代表されるグラフ構造 [1] における差異の検出を具体的なターゲットとし、地域、イベント発生の前後、トピック、時間、カテゴリー等の文脈変化の前後における2つのグラフを比較し、潜在的だが決して偶発的ではない差異の候補を検出するタスクについて考察する。

無向グラフに対する頂点結合の必然性を図る尺度として疑似クリークが提案されている。[2] グラフの構造上の特徴として、一方のグラフに対しては疑似独立集合になり、もう一つのグラフに対しては疑似クリークとなっているこ

とを要請した。発散的な疑似独立集合から内向きに閉じた疑似クリークへの変化を必然性と解する。疑似クリークに関しては様々な定義が可能だが、ソーシャルネットワーク解析において一定の使われ方をしている k -plex [3] を用いる。 k -plex を求める際、 k の値がある程度大きくなければ大きな頂点集合を得にくい。しかし、 k の値を大きくすると組み合わせのパターンが膨大に増え、それに伴い計算量も膨大に増加する。

本稿では、疑似クリークではなくノーマライズカットによる制約を設ける。ノーマライズカットの値は頂点集合内の辺の多さと、その外部へ向かう辺の少なさで決まるので内向きに閉じた密な頂点集合を取り出せる。そのため従来の組み合わせによる差異の検出より大きなクラスタを取り出しやすい。ノーマライズカットを求めるためにスペクトル解析を用い、得られた距離空間においてグラフを作りクリーク枚挙エンジンを使用することで計算時間を短縮できる。さらに、計算時間に影響するパラメータがなく、しかもそれらは一般の統計的な観点から設定できる。本稿では、時間差によって得られる2つのグラフを対象として差異の抽出実験を示すが、これに加えて地域・場所の差異の抽出も予定している。一部が徐々に変化していく傾向の強い時間の変化とは違い、地域ごとに取り上げられる話題などをグラフとすると、二つのグラフに大規模な差異が生じる可能性がある。本手法ならば、大きな集合からの組み合わせ探索を避けられるため、その大規模な変化をとらえられる可能性がある。その実験は発表時に報告したい。

¹ 北海道大学
Hokkaido University, Sapporo, Hokkaido 060-0814, Japan

² 電気通信大学
UEC Tokyo

a) mazawa@kb.ist.hokudai.ac.jp

b) mh@ist.hokudai.ac.jp

c) tomita@ice.uec.ac.jp

2. スペクトル解析とノーマライズカット

本章ではスペクトル解析とノーマライズカットの関係について記述する。クラスタリングの目的はデータをいくつかのグループに分けることである。似たデータを同じグループ、似ていないデータを別のグループとすることで似たデータ同士の集合を集める。各データを頂点集合 $V = \{v_1, \dots, v_n\}$, 頂点 v_i と v_j の共起を w_{ij} としたグラフからスペクトル解析をする。

2.1 グラフの定義

頂点集合 $V = \{v_1, \dots, v_n\}$ を持つ無向グラフ $G = (V, E)$ を定義する。 G は重み付きの辺を持ち、頂点 v_i と v_j 間の非負の重みを $w_{ij} \geq 0$ と表す。頂点 v_i と v_j の全組み合わせに対する w_{ij} の行列 $W = (w_{ij})_{i,j=1,\dots,n}$ を V の重み付き隣接行列とする。 $w_{ij} = 0$ ならば頂点 v_i と v_j 間に辺はつながっていない。 W は対称行列であり $w_{ij} = w_{ji}$ となる。頂点 $v_i \in V$ に対して次数

$$d_i = \sum_{j=1}^n w_{ij}.$$

を定義する。次数行列 D は、次数 d_1, \dots, d_n を対角成分に持つ対角行列である。

部分集合 $A \subset V$, 部分集合に含まれない集合 $\bar{A} = V \setminus A$ と部分集合に含まれる頂点を示すインディケーターベクトル $\mathbb{1}_A = (f_1, \dots, f_n)' \in \mathbb{R}^n$ を定義する。

$$f_i = \begin{cases} 1 & \text{if } v_i \in A \\ 0 & \text{otherwise.} \end{cases}$$

索引 $i \in A$ を $\{i | v_i \in A\}$ とすると、部分集合 A の隣接行列を $\sum_{i \in A} w_{ij}$ として扱える。素集合である必要のない二つの部分集合 $A, B \subset V$ に対して

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}.$$

が定義される。部分集合 A に対して以下の大きさを定義する。

$$\text{vol}(A) := \sum_{i \in A} d_i.$$

実際にスペクトル解析で考えるのは A と \bar{A} の二つの部分集合ではなく非空集合の k 個の頂点集合 A_1, \dots, A_k ($A_i \cap A_j = \phi$ and $A_1 \cup \dots \cup A_k = V$) に分割する。

2.2 正規化グラフラプラシアンとスペクトル解析

ノーマライズカットを最大化するためにはグラフラプラシアン L を正規化した正規化グラフラプラシアン L_{norm} が必要である。グラフラプラシアンは

$$L = D - W$$

と定義される。この行列は以下の性質を持つ

(1) あらゆるベクトル $f \in \mathbb{R}^n$ に対して

$$f' L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2.$$

が成り立つ。

(2) L は半正定値の対称行列。

(3) L の最小固有値は必ず 0, それに対する固有ベクトルは $\mathbb{1}$ 。

(4) L は非負実数の固有値を n 個持つ。

$$(0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n).$$

正規化グラフラプラシアンは以下のように定義される。

$$L_{\text{norm}} = D^{-1/2} L D^{-1/2}.$$

この行列は以下の性質を持つ

(1) あらゆるベクトル $f \in \mathbb{R}^n$ に対して

$$f' L_{\text{norm}} f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2.$$

が成り立つ。

(2) L_{norm} の固有ベクトル u には次数で補正が掛かっており $D^{1/2} u$ が実際の値となる。

(3) L_{norm} の最小固有値は必ず 0, 共起する固有ベクトルにも補正が掛かっており, $D^{1/2} \mathbb{1}$ となる。

(4) L_{norm} は非負実数の固有値を n 個持つ。

$$(0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n).$$

2.3 ノーマライズカット

カットとはグラフを分割した際に切断される辺の重みの総和であり、以下の様に定義される。

$$\text{cut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k W(A_i, \bar{A}_i).$$

この値が小さいと部分集合から外部へ向かう辺が少なくなり閉じた集合であるといえる。ノーマライズカットはカットに部分集合内の辺の総和で補正をかけたもので、

$$\begin{aligned} \text{Ncut}(A_1, \dots, A_k) &:= \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, \bar{A}_i)}{\text{vol}(A_i)} \\ &= \sum_{i=1}^k \frac{\text{cut}(A_i, \bar{A}_i)}{\text{vol}(A_i)}. \end{aligned}$$

と定義される。部分集合内部の辺の数が多い密な結合ほどこの値は小さくなりやすい。

二つの部分集合に分ける場合、Ncut のインディケーターベクトルを

$$f_i = \begin{cases} \sqrt{\frac{\text{vol}(\bar{A})}{\text{vol}(A)}} & \text{if } v_i \in A \\ -\sqrt{\frac{\text{vol}(A)}{\text{vol}(\bar{A})}} & \text{if } v_i \in \bar{A} \end{cases}$$

$$(i = 1, \dots, n; j = 1, \dots, k).$$

と定義すると,

$$\begin{aligned} f'Lf &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2 \\ &= \frac{1}{2} \sum_{i \in A, j \in \bar{A}} w_{ij} \left(\sqrt{\frac{\text{vol}(\bar{A})}{\text{vol}(A)}} + \sqrt{\frac{\text{vol}(A)}{\text{vol}(\bar{A})}} \right)^2 \\ &\quad + \frac{1}{2} \sum_{i \in \bar{A}, j \in A} w_{ij} \left(-\sqrt{\frac{\text{vol}(\bar{A})}{\text{vol}(A)}} - \sqrt{\frac{\text{vol}(A)}{\text{vol}(\bar{A})}} \right)^2 \\ &= \text{cut}(A, \bar{A}) \left(\frac{\text{vol}(\bar{A})}{\text{vol}(A)} + \frac{\text{vol}(A)}{\text{vol}(\bar{A})} + 2 \right) \\ &= \text{cut}(A, \bar{A}) \left(\frac{\text{vol}(A) + \text{vol}(\bar{A})}{\text{vol}(A)} \right. \\ &\quad \left. + \frac{\text{vol}(A) + \text{vol}(\bar{A})}{\text{vol}(\bar{A})} \right) \\ &= \text{vol}(V) \text{Ncut}(A, \bar{A}) \end{aligned}$$

となり, $g := D^{1/2}f$ (L_{norm} の性質 (2) より) とすると,

$$g'D^{-1/2}LD^{-1/2}g = g'L_{\text{norm}}g$$

なので, L_{norm} の分割は, ノーマライズカットの値に対応する. $k > 2$ のときも同様である.

グラフを k 個に分割する際, インディケータ行列 $H \in \mathbb{R}^{n \times k}$ は

$$h_{i,j} = \begin{cases} 1/\sqrt{\text{vol}(A_j)} & \text{if } v_i \in A_j \\ 0 & \text{otherwise} \end{cases}$$

$$(i = 1, \dots, n; j = 1, \dots, k).$$

と表せる. よって,

$$\begin{aligned} \text{Ncut}(A_1, \dots, A_k) &= \sum_{i=1}^k h_i' L_{\text{norm}} h_i \\ &= \sum_{i=1}^k (H' L_{\text{norm}} H)_{ii} \\ &= \text{Tr}(H' L H). \end{aligned}$$

Ncut 最小化の式は,

$$\min_{H \in \mathbb{R}^{n \times k}} \text{Tr}(H' L H).$$

ただし, $H'H = I$.

3. クラスタリング

本章では, 前章のインディケータ行列 H の値を決定するためのクラスタリング手法について記述する. L_{norm} の 0 以外の N 番目までの最小固有値に対応する固有ベクトル $D^{1/2}u$ を距離空間として, DBSCAN により近い点をクラスタとしてまとめる. 得られたクラスタから単位円グラフ

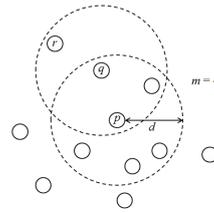


図 1 DBSCAN の例

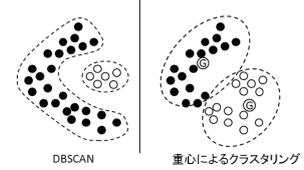


図 2 クラスタリングの比較

を作り, 極大クリークを枚挙する. それが内部の辺の多く, 外部へ向かう辺の少ない頂点集合となる. クラスタから単位円グラフを作る前に, クラスタの集合にもう一方のグラフの疑似独立集合による制約を掛けるが, それについては次章で説明する.

3.1 DBSCAN

DBSCAN とは, ある点 $p_i (i = 1, \dots, n)$ から一定の距離 d 内にある点の数を数え, 規定値 m 個以上の点が d 内に含まれていれば p_i をクラスタの core, core から距離 d 内にあり core の条件を満たさない点をクラスタの boarder として, core と boarder を合わせた集合を一つのクラスタとするクラスタリング手法. 図 1 のように, 規定値 $m = 4$ とすると, 点 p から距離 d 以内に五つの点が含まれるので点 p は core となる. 点 q は距離 d 以内に三つの点しかもっていないため core にはなれない. しかし, core である点 p から距離 d 以内に含まれているため boarder としてクラスタに含まれる. 点 r のような core ではなく, 別の core にも含まれない点ほどのクラスタにも含まれないノイズとして破棄される.

DBSCAN は一定距離内の点を追うようにクラスタリングできるので複雑な形のクラスタを取り出すことができる. また, 計算量は $O(n \log n)$ と非常に高速である. 一般的な重心を用いたクラスタリングでは円または楕円形のクラスタとなるが, DBSCAN では複雑な形のクラスタを取り出せる. 図 2 のようなデータセットを二つに分ける場合, 左側の細長く折れ曲がったデータの集まり (以降クラスタ A) と右側の小さな集まり (以降クラスタ B) で分けたい. 重心を用いたクラスタリングではクラスタ A に囲まれるようにクラスタ B があるため, 図 2 右側のようにクラスタ B とクラスタ A の一部を結合させることになり, 二つ以上に分けられるとしてもクラスタ A を分割しなければならない. しかし, DBSCAN は近くにある点を追うようにクラスタリングするためクラスタ A のような複雑な形でも一つのクラスタとして取り出せる.

3.2 単位円グラフ

単位円グラフ (Unit Disk Graph) とは, 平面上に配置された頂点から半径 r の単位円を展開し円が交差, または接していれば辺を張ることで作られるグラフである. 単位円

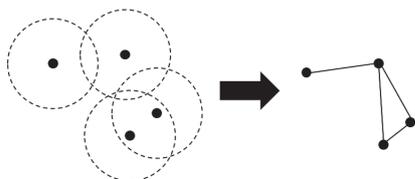


図 3 単位円グラフ

	1	2	3	4	5	6
1	0	0	0	0	1	0
2	0	0	0	0	1	0
3	0	0	0	0	1	0
4	0	0	0	0	1	0
5	1	1	1	1	0	0
6	0	0	0	0	0	0

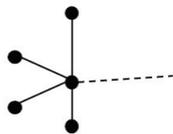


図 4 スター構造グラフの例

グラフは主にアドホックネットワークのモデルとして知られている。[7] アドホックネットワークとは基地局やアクセスポイントなどを用いずに端末のみで構成されるネットワークある。各端末の通信半径が全て等しいと仮定したときに得られるネットワークが単位円グラフとなる。

本手法は、DBSCAN で得られた頂点集合を基に単位円グラフを作る。DBSCAN によりクラスタの core となる点は一定の距離 d 以内に必ず m 個以上の頂点を持っており、単位円グラフの半径 $r \geq d/2$ ならば少なくとも m 本以上の辺を張ることができ、クリークを枚挙する際にある程度大きな頂点集合に絞り込むことができる。

固有値分解の場合、単純な距離パラメータでは取り出しにくいクラスタがある。それが、図 4 のようなスターグラフに近い構造の場合である。この場合、1~4 の固有ベクトルは非常に近い値となるが、スター構造の中心の固有ベクトルとは多少離れる。この距離は中心につながる数や固有ベクトルの大きさに左右され様な距離で近さを計れない。しかし、他の頂点と比べて 1~4 に近い距離にある可能性が高いため以下の手段を取る。

- (1) DBSCAN でクラスタを作り、得られたクラスタの各頂点からクラスタ外の最近傍の頂点集合 (以降 Cand) を取り出す。
- (2) $Cand_i (i = 1, \dots, n)$ を最近傍を持つクラスタ内の頂点集合の内、最も遠いものも出の距離を $Cand_i$ の r とする。
- (3) クラスタ内の各頂点を持つ r と交差または接した場合は辺を張る。クラスタ外の頂点とは辺を張らない。これにより、クラスタごとに動的な近さの基準を作る。

4. 疑似独立集合

本稿では、グラフの比較を行うため一方のグラフ (以降 Target) には 2 章のノーマライズカットによる制約をかけ、

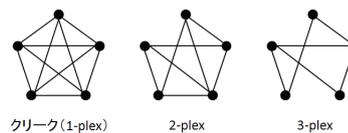


図 5 k-plex



図 6 補グラフの k-plex 制約の例

そして、もう一方のグラフ (以降 Base) には疑似独立集合による制約をかける。独立集合とは集合内部に一切辺を持たない集合であり、疑似独立集合は集合内の辺がほとんどつながっていない集合を指す。部分集合が独立集合、疑似独立集合という場合、ともに部分集合の外へのつながりは制限されない。つまり、主に部分集合の外部へつながっている発散的な集合といえる。補グラフに k-plex 制約をかけることで疑似独立集合を検出する。

4.1 k-plex

k-plex とはクリークの緩和モデルの一つで、各頂点ごとに非隣接頂点数を制限している。k-plex は自信をのぞいた非隣接頂点数は高々 $k-1$ までの条件を満たす頂点集合。独立集合とクリークは逆の概念であり、クリークの補グラフが独立集合である。よって、本稿ではクリークの緩和モデル、疑似クリークである k-plex の補グラフを疑似独立集合として扱う。

4.2 グラフ比較のための制約

3 章の Target から得られたクリークに対し、base で同じ頂点集合の部分グラフを作る。その補グラフが k-plex 制約を満たすか確認する。k-plex 制約を満たすか確認する方法は非常に簡単であり、部分集合内の頂点の次数を数えるだけで組み合わせなどは一切使わない。図 6 のように、補グラフの次数が $k-1$ 以下の頂点を除けば、残りのグラフは k-plex となる。つまり、部分集合内の頂点の次数が $k-1$ 以下の頂点だけを残せば、Target からクラスタリングで得た集合、および、その最近傍から、Base において疑似独立集合の条件を満たす頂点集合を取り除ける。その単位円グラフから枚挙された極大クリークが、一方では発散的でもう一方では内側に密で閉じた集合となり、最終的な出力となる。

5. 実験

本稿での提案手法を Java および Mathematica で実装し、CPU : Intel Core(TM)2 Quad CPU Q6600(2.40GHz)、主記憶 : 2GB の PC 上で、Twitter ユーザを頂点とする無向

グラフを対象とした構造の差異の検出を試みた。

5.1 Twitter ユーザの関係グラフ

Twitter 検索 API を用いて、キーワード民主党・自民党・公明党・社民党・共産党・みんなの党・国民新党・たちあがれ日本・新党大地・新党きづな・大阪維新の会の何れかを含むツイートを集め、それらツイートに関わったユーザ間の関係をグラフで表現する。具体的には、ユーザ A がユーザ B に対してリツイートを行った場合に、辺 (A;B) を張ることでユーザ間の関係グラフを作成する。ここでは、平成 24 年 4 月 4 日～6 日の 3 日間に収集したツイート群から抽出したユーザ間の関係グラフをグラフ G_{base} 、平成 24 年 4 月 13 日～15 日の 3 日間のツイート群からのそれをグラフ G_{target} とし、そこで観測される構造差異の検出を試みる。

5.2 パラメータ設定

本節では、実験で設定するパラメータについて説明する。実験データとして、頂点数 1999 (G_{base} , G_{target} 共通)、辺数 4137 (G_{base})、4187 (G_{target}) のグラフを用いる。DBSCAN のパラメータは $d = 1.0 \times 10^{-14}$ 、 $m = 5$ とする。 d は全固有ベクトルの最近傍までの距離を昇順に並べ、最初にあられた閾値を参考に決める。 m は特に大きすぎる、もしくは小さすぎない限り問題はない。 $(6 \geq m \geq 3)$ 程度が現実的) DBSCAN で得たクラスタ内の単位円グラフの半径 $r = d$ とする。 G_{base} における k-plex の補グラフによる疑似独立集合の制約は $k=4$ とする。 G_{target} において極大クリークを列挙し、そのサイズ s の度数分布表を作る。サイズ s 以上の極大クリークの度数と全体の度数との割合がある程度小さくなる s の値を k とする。 G_{target} において部分集合内で度数 s の頂点は有意に密とみなし、 G_{base} において、部分集合内で度数 s 以上の頂点を破棄する。

5.3 解集合

解集合の一例として、平成 24 年 4 月 13 日～15 日の 3 日の間に某政党の原子力発電所再稼働に対する見解や、党員の汚職疑惑に対して議論する 6 人のユーザが見つかった。このユーザ間のやりとりは 6 割以上がこの 6 人の間で行われてた。この 6 人のユーザは平成 24 年 4 月 4 日～6 日の 3 日間には一切やり取りがなく、外部のユーザとのやりとりのみである。そこでのツイート内容は、某政党のマニフェスト違反や消費税増税に対する意見がある。

実際、4 月 13 日になされた大飯原発運転再開が妥当であるとの判断を巡って、社会的な議論が巻き起こっており、某政党に対する批判が高まったものと考えられる。このように、本構造差異検出手法によりある種のコミュニティ形成が始まる様子を捕えることが可能である。

この例では最も差異が顕著だった例を挙げたが、本手法

では、ある程度大きな集合を取り出すことに向いている。解集合が大きくなると、集合内部の密度が下がりやすく、外とのつながりが多くなる傾向があるが、同様にある政党に対して批判、あるいは支持する集合が得られた。

ユーザ間の関係グラフを作成する際、今回はリツイートの有無のみに着目した無向辺を考えたが、これを有向辺として扱うことで現実の関係をよりの確に反映したものになると思われる。さらに、やりとりの頻度を辺の重みとして反映させることも考えられる。これらの点を考慮した制約や評価尺度を導入することで、抽出対象となる集合により明確な意味を持たせることができると思われる。

5.4 計算時間と解集合のサイズ

本稿において、いくつか設定するパラメータがあるが、そのどれも計算時間には大きく関わらず、単純に関係グラフの大きさの 2 乗に比例する。計算過程で固有値分解をしており、計算時間のほとんどはそれに費やされる。計算のうち固有値分解は Java から Mathematica を呼び出して行われ、それ以外は Java プログラムにより計算される。 G_{target} においてグラフのサイズを変え、Ncut による制約をかける場合 (以降 Ncut 制約) と、k-plex による制約をかける場合 (以降 k-plex 制約) で計算時間と解集合のサイズを比較し、本機構の有用性を確認する。

実験に使用したグラフのサイズは、表 1 参照。得られた解集合の最大サイズは Ncut 制約では 29、k-plex 制約では 4 だった。k-plex 制約は G_{target} と G_{base} の統合グラフの k-plex の全列挙により発散的な疑似独立集合から内側に密な閉じた集合を検出している。

得られる解の最大サイズはどのデータセットでも大差はなく、Ncut 制約では 20～30、k-plex 制約では 4 か 5 である。この結果は、twitter 上でリツイート関係から作られたグラフは密度等の条件がほぼ同様であるためと思われる。

計算時間に関しては、頂点数を増やしていくと図 7 のように k-plex 制約は組み合わせの数が多くなり、計算時間は指数関数的に増える。一方、Ncut 制約の場合、図 8 のように多項式時間で高速に計算でき、比較的大きな頂点集合も容易に検出できる。k-plex の全列挙において k の値を増やすと計算量は指数関数的に増える。また、グラフの密度が上がると組み合わせ数も増えやすくなり、計算時間が増える原因となる。Ncut 制約は計算量に関わるパラメータもなく、密度の影響も受けないので、何らかの条件により計算時間が極端に大きくなることはないため容易に扱える。

6. まとめと今後の展望

構造の差異を検出するために、Ncut や k-plex を用いる本研究は、一方のグラフにおいて頂点集合から外部への辺が少なく内部で多いこと、および、もう一方のグラフにおいて内部での辺が少ないことを制約として用いた。この

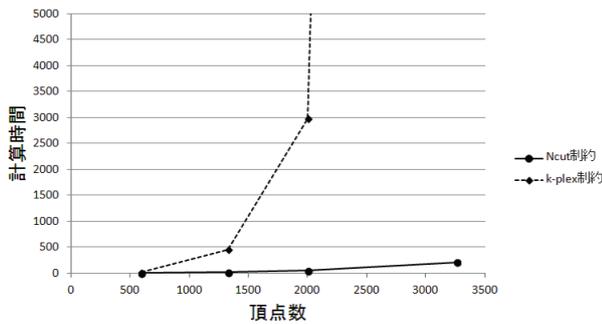


図 7 計算時間

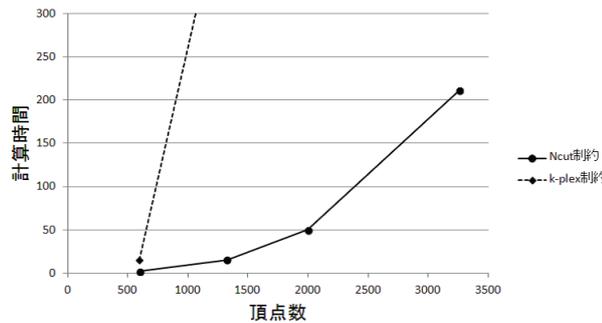


図 8 計算時間 (拡大)

表 1 頂点と辺

頂点数	辺 (G_{base})	辺 (G_{target})
594	964	914
1325	2451	2436
1999	4137	4187
3259	8134	8155

条件の場合、内部が変わらず疎のまま、外部への辺が極端に減った集合を検出してしまうことがある。Ncut 後のチェックを Base のみから、Base・Target の両方を見ることで改善が期待できる。また、単位円グラフからクリークを枚挙しているが、それを k-plex にすることでさらに大きな頂点集合を得られる可能性がある。DBSCAN で検出されるクラスは最大でも 100 程度のため組み合わせの計算にはそれほど時間はかからないと考えられる。

参考文献

[1] B. Furt (ed.), Handbook of Social Network Technologies and Applications, Springer, 2010.
 [2] 大久保好章 and 原口誠, 擬似クリークと負の制約を用いたグラフの構造変化検出, The 26th Annual Conference of the Japanese Society for Artificial Intelligence, 2012
 [3] S. B. Seidman and B. L. Foster, A Graph Theoretic Generalization of the Clique Concept, Journal of Mathematical Sociology, 6, pp. 139 - 154, 1978.
 [4] Ulrike von Luxburg, A tutorial on spectral clustering, Statistics and Computing, 17 (4), 2007.
 [5] Martin Ester, Hans-Peter Kriegel, Jorg Sander and Xiaowei Xu, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, Knowledge Discovery and Data Mining (KDD-96), 1996.

[6] Brent N. CLARK and Charles J. COLBURN, Unit Disk Graphs, December 1990, Pages 165-177, 1990.
 [7] Fabian Kuhn, Roger Wattenhofer and Aaron Zollinger, Ad-Hoc Networks Beyond Unit Disk Graphs, DIAlM-POMC '03 Proceedings of the 2003 joint workshop on Foundations of mobile computing Pages 69-78, 2003.
 [8] Bin Wu and Xin Pei, A Parallel Algorithm for Enumerating All the Maximal k-Plexes, PAKDD 2007 Workshops, LNAI 4819, pp. 476-483, 2007.
 [9] M. Haraguchi, Things That Change, Things That Do Not Change, Proc. of GCOE-NGIT 2012, pp. 19 - 21, 2012.
 [10] エラウィンディサラ・原口誠・大久保好章・富田悦次, クリーク全列挙に基づく構造変化検出アルゴリズム, 情報処理学会研究報告, Vol. 2011-MPS-087 No. 32, 2012.