3F7-1

大規模グラフにおける近似部分グラフ検索手法

Searching for Approximate Sub-graphs from a Large Scale Graph

大原 剛三[*1] 鷲尾 隆[*1]
Kouzou Ohara Takashi Washio

[*1]大阪大学産業科学研究所
The Institute of Scientific and Industrial Research, Osaka University

Finding out given sub-graphs from a data graph is one of essential tasks when dealing with graph structured data, although it would be hard if the data graph becomes larger. On the other hand, it might be useful to find out sub-graphs similar to a given query graph, but not ones that exactly match it. Thus, in this paper, we propose a method of searching for sub-graphs similar to a given query graph from a large scale data graph. For that purpose, our method first finds out at most k vertices in the data graph for each vertex in the query graph by means of SASH which realizes an efficient approximate k-NN query, and then constructs approximate sub-graphs from them based on adjacence relation between vertices in the query graph. We also show results of preliminary experiments and discuss the features of the proposed method.

1. はじめに

近年，データマイニングの分野では，単純な表形式では記述できない構造をもつデータを，柔軟に表現できるグラフ構造データへの関心が高まり，グラフとして表現された大量データから知識として有用な部分グラフを見つけ出すグラフマイニングの研究が盛んに行われるようになった [Washio 05, Cook 06]．しかしながら，グラフマイニングにより発見された部分グラフをどう利用するかについては，それほど多くの議論がなされていないのが現状である．発見した部分グラフを実際に利用するためには，新たに与えられたデータにその部分グラフが含まれるか否かを判定することが必要となるが，その判定の困難さにも起因する．この問題は部分グラフ同型判定問題と呼ばれ，近年においても可能な限り効率よくこの問題を解く手法が幾つか提案されている [Messmer 00, 西村 06]．しかしながら，この問題を解く計算量は理論的には NP 完全であることが知られており，それらの手法を用いても対象となるグラフのもつ頂点数に対してその最悪計算量は指数オーダーとなる．

一方，実際の応用では，必ずしも指定した部分グラフと完全に一致する部分グラフを見つけることが有用ではない場合がある．これは，実データにおけるノイズの存在やデータグラフの不完全性などに加え，利用者の興味の対象が部分グラフ検索時点で明確になっていない場合が多分にあるためである．特に，意外な知識の発見という観点からは，検索要求となったグラフに完全に一致する部分グラフのみを見つけるよりも，構造が類似した部分グラフを見つけ出すほうが有用であると考えられる．Yan[Yan 05] らの手法はこのような要請に答えるものであるが，この手法は指定した部分グラフに類似する部分グラフを含み得ないグラフを検索対象から排除するためのものであり，検索対象となるグラフの削減には有用であるが，個々のグラフが大規模化した場合の効率改善には役立たず，最終的な部分グラフの探索は合わせて用いられる類似部分グラフ探索手法に依存する．

以上のような背景の下，本研究では大規模なグラフ（以下，データグラフ）から指定したグラフ（以下，質問グラフ）を厳密に見つけ出すのではなく，質問グラフと構造が類似する部分グラフ（以下，近似部分グラフ）を効率よく検索する手法を提案する．提案手法では，まずグラフの頂点を近傍構造で特徴付け，その差異に基づき頂点間の距離を定義し，質問グラフ中の頂点と近傍構造が類似する頂点をデータグラフから見つけ出す．このプロセスを効率的に実現するために，本研究では SASH (Spatial Approximation Sample Hierarchy) [Houle 05] と呼ばれるデータ構造を用いてデータグラフの頂点を構造化する．SASH は与えられた事例集合を事例間の距離に基づき階層化したもので，事例数が n の場合，その中から質問事例 d に最も近い k 個の事例を求める k-NN 探索の近似解を $O(\log_2 n)$ で求めることができる．提案手法では，このような SASH を通して得られた頂点集合から，質問グラフ中の頂点の接続関係に基づき近似部分グラフを同定する．本稿では，さらに提案手法のプロトタイプによる予備実験の結果を示し，提案手法の特性を解析する．

連絡先: 大原 剛三，大阪大学産業科学研究所
〒 567-0047 大阪府茨木美穂ヶ丘 8 － 1
E-mail: ohara@ar.sanken.osaka-u.ac.jp

2. グラフ構造データ

グラフは頂点と頂点間を結ぶ辺で表現される．本研究では，各頂点と辺がラベルをもつラベル付きグラフを対象にする．いま，V を頂点 v の集合としたとき，ラベル付きグラフ g を $g = (V, E, L, \varphi)$ と表す．ここで，$E \subseteq V \times V$ であり E を辺集合，$e \in E$ を辺と呼ぶ．L はラベルの集合であり，φ は任意の $v \in V$ もしくは $e \in E$ に対して 1 つのラベルを割り付ける写像である．また，グラフ $g = (V, E, L, \varphi)$ に対して，グラフ $g' = (V', E', L, \varphi)$ が $V' \subseteq V$，$E' \subseteq E$ を満たすとき，g' を g の部分グラフと呼ぶ．

一般に，頂点数 n のグラフは $n \times n$ の行列を用いて表現することができる．そのような行列は隣接行列と呼ばれ，第 i 行（第 i 列）を頂点 v_i とし，頂点 v_i から v_j に辺が存在する場合，(i, j) 成分は 1，もしくは辺のラベルとなり，辺が存在しない場合には 0 となる．辺に向きのない無向グラフの場合，(i, j) 成分と (j, i) 成分は同じ値をもつが，辺に向きのある有向グラフの場合，v_i から v_j への辺が (i, j) 成分に対応し，v_j から v_i への辺が (j, i) 成分に対応するものとする．

次に，本研究でグラフ中の頂点の特徴付けに用いる経路 ($path$) を定義する．グラフ g 中の頂点 v に対する長さ h の経路とは，$v_i \neq v (1 \leq i \leq h)$ である重複を含まない h 個の頂点により形成される h 個の辺の系列 $(v, v_1), (v_1, v_2), \ldots, (v_{h-1}, v_h)$

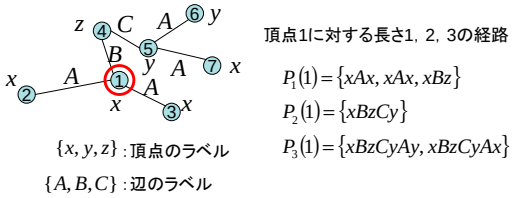

図 1: 経路の例

である．本稿では，経路を構成する各頂点，および各辺のもつラベルの系列で経路を表現する．以下では，頂点 v に対する長さ h のすべての経路からなる集合を $P_h(v)$ と表す．$P_h(v)$ には同一のラベル系列が複数存在し得るため，$P_h(v)$ は多重集合となる．経路の例を図 1 に示す．図 1 では，頂点 1 に対する長さ 1，2，3 の経路の集合を示している．$P_1(1)$ が多重集合となっている点に注意されたい．一方，辺の系列において，頂点の重複を許す場合，その辺の系列は $walk$ と呼ばれる．経路，および $walk$ は，グラフに対するカーネル関数を定義する際に，グラフの特徴量として多用される [Kashima 03, Horváth]．なお，カーネル関数を用いたグラフの判別は一般には 2 つのグラフ全体を比較するものであり，その部分構造の検索には用いることはできない．

3. SASH の概要

本節では，提案手法で用いる SASH の概要について述べる．SASH は高次元データに対して，k-NN 探索を効率的に実現するために考案されたデータ構造である．通常の k-NN 探索のオーダーは事例数 n に対して線形であり，対象データが高次元ベクトルで表現される場合には計算コストの高い距離計算の影響を大きく受けるため，そのようなデータに k-NN 探索を直接用いることは現実的ではない．そのため，SASH では距離関数 $dist$ の下で近傍に存在するデータ同士を事前に階層的に関連付け，検索の効率化を図り，検索時のオーダーを $O(\log_2 n)$ に軽減している．類似手法として，ピボットと呼ばれる複数の基準ベクトルを用いた Vantage Point Tree[Chávez 01, Dehen 87] やボロノイ図を利用した Voronoi Tree[Chávez 01, Yianilos 93] などの検索時計算量が同様に $O(log_2 n)$ である手法が提案されているが，実際の計算速度や検索精度は SASH が非常に優れいてることが明らかになっている．ただし，近傍関係を厳密に再現すると計算コストが高くなるため，SASH 内では近傍関係は完全には再現されない．そのため，検索要求となる事例 d に距離関数 $dist$ の下で最も近い k 個の事例を求める場合，SASH に基づいて得られる事例は近似解となり，通常の k-NN 探索により得られる k 個の事例と必ずしも一致するとは限らない．以下では，SASH による k-NN 探索を近似 k-NN 探索，厳密に d に対する k 個の最近傍データを求める通常の k-NN 探索を完全 k-NN 探索と呼ぶ．

以下，SASH の構築方法と，検索方法を概説する．SASH の構築手順は大きく，(1) 各階層に割り当てる事例を決定する，(2) 上位階層から順に，各階層とその 1 つ上の階層の事例間に，それらの距離に基づき親子関係を決定する（事例の関連付け），という 2 つの手順からなる．手順 (1) では，各階層の事例はランダムに選択し，その数は事例集合 D の事例数を n とした場合，最上位階層は 1，最下位階層は $\lceil n \rceil$，その他の各階層は直下の階層の事例数の半分とする．したがって，直感的には SASH のデータ構造は図 2 に示すようなピラミッド形となる．手順 (2) における事例の関連付けに用いる距離関数には，任意

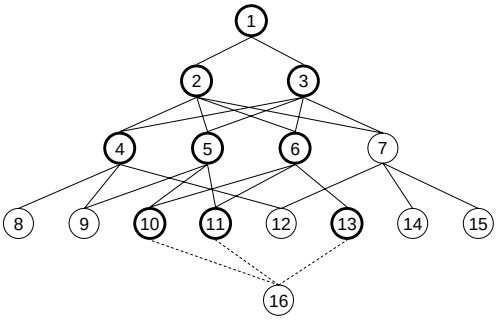

図 2: SASH における親候補選択の例

の $p, q \in D$ について以下の条件を満たすものを用いる．

- $dist(p,p) \geq 0$.
- $p = q$ であれば $dist(p,q) = 0$.
- $dist(p,q) = dist(q,p)$.

階層 l 中の事例 d の親となり得るのは，階層 $l-1$ 中の事例のうち距離関数 $dist$ において d の近傍に位置するものであり，その数は最大 p 個となる．ただし，親候補の決定は単純に階層 $l-1$ 中の事例から求めるのではなく，最上位階層から順に階層的に求める．具体的には，階層 i においては，その上位階層 $i-1$ で求めた最大 p 個の d の近傍事例の子となる事例の中から d に最も近い最大 p 個の事例を求める．これを階層 $l-1$ まで繰り返し，d の親候補を決める．図 2 は，$p=3$ のときに階層 4 まで構築されている SASH に階層 5 の事例 16 を追加する場合の例であり，太線で記述された事例が各階層において事例 16 と最も近い事例である．したがって，事例 16 の親候補は，階層 3 における事例 4，5，6 のすべての子のうち事例 16 に最も近い事例 10，11，13 となる．このように階層 l 中のすべての事例に対して親候補を決定した後，最終的には，階層 $l-1$ の特定の事例に子が集中しないように，それぞれの親を決定する．このような手順で事例数 n の事例集合 D を構造化するための計算量は $O(n \log_2 n)$ となる．

一方，SASH として構造化された事例集合に対して，ある事例 d' と類似する k 個の事例を求める検索手順は，SASH 構築と同様に階層的な手順となる．すなわち，最上位階層から順に各階層において d' と距離関数 $dist$ において最も近い最大 k 個の事例を選択し，次の階層ではそれらの子となる事例からさらに k 個の事例を選択することを繰り返す．そして，そのように選択したすべての事例の中から d' に最も近い k 個の事例を最終的に選択して解とする．

4. 近似部分グラフ検索アルゴリズム

本稿で提案する近似部分グラフ検索アルゴリズムは大きく，以下の 2 つの手順から構成される．

1. 質問グラフ g' の各頂点 v' に対して，v' と近傍構造が類似するデータグラフ g 中の頂点 v を k 個求める．
2. g' の各頂点の接続関係に基づき，上の手順で求めた g 中の頂点から g' と構造が類似する g' の近似部分グラフを求める．

頂点数の多い大規模グラフを想定した場合，上記手順 1 を実現するために完全 k-NN 探索を利用すると頂点数 n のデータ

グラフ g，頂点数 m の質問グラフ g' に対して，$O(mn)$ の計算量が必要となるため効率的ではない．そのため，提案手法ではその計算量を軽減するために，データグラフ中の頂点を事前に SASH により構造化し，それに対して質問グラフ中の各頂点を検索質問として近似 k-NN 探索を実行する．この場合，計算量は $O(m \log_2 n)$ となり，データグラフの規模が大きくなるほど SASH の効果が大きくなる．

以下では，SASH による近似 k-NN 探索を実現するために，グラフ中の各頂点 v をその近傍構造に基づき特徴付け，頂点間の距離を定義する．具体的には，以下のように定義される各頂点 v に対する長さ h 以下の経路の集合 $f_h(v)$ を v の特徴量とする．

$$f_h(v) = \cup_{i=1}^{h} P_i(v) \tag{1}$$

$P_i(v)$ が多重集合であることから，$f_h(v)$ もまた多重集合となる．$f_h(v)$ は頂点 v を起点とした長さ h の範囲の頂点間の接続関係を反映しており，v の近傍構造を表していると考えることができる．

このとき，任意の 2 頂点 v_i と v_j 間の距離 $dist(v_i, v_j)$ を以下のように定義する．

$$dist(v, v') = \begin{cases} 0 & f_h(v) = f_h(v') \\ 2 & f_h(v) \cap f_h(v') = \phi \\ \frac{1}{|f_h(v) \cap f_h(v')|} & \text{otherwise} \end{cases} \tag{2}$$

ここで，$|f_h(v) \cap f_h(v')|$ は 2 つの多重集合の共通部分の元の個数を表す．このように定義した距離関数 $dist$ は前述の SASH が距離関数に求める 3 つの要件を満たす．直感的には，この距離関数は共通する経路を多くもつ頂点間ほどその距離は小さくなり，同一の場合に最小値 0 を取り，まったく共通する頂点がない場合に最大値 2 を取る．

次に，手順 2 において SASH により求めたデータグラフ g 中の頂点から近似部分グラフを求める方法について述べる．いま，質問グラフ $g' = (V', E', L', \varphi')$ の頂点を $v'_1, \ldots, v'_m \in V'$，各頂点 $v'_i (1 \leq i \leq m)$ に対して SASH で求めた最大 k 個の g 中の頂点からなる集合を $A(v'_i)$ とする．このとき求めるのは，第 i 行（第 i 列）に対応する頂点 v が $v \in A(v'_i)$ となる $m \times m$ の隣接行列 M_{g_a} で表現される g の部分グラフの集合 C_m である．ここで，データグラフを $g = (V, E, L, \varphi)$，隣接行列 M の第 i 行（第 i 列）に対応する頂点を $v_i(M)$ とした場合，近似部分グラフ g_a は，任意の i, j について $(v'_i, v'_j) \in E'$ であるならば $(v_i(M_{g_a}), v_j(M_{g_a})) \in E$ とする．

提案手法では，このような近似部分グラフ g_a を g_a の接続行列における第 i 行 $(1 \leq i \leq m)$ に対応する頂点を順次決定して求めるために，以下に示すナイーブな手順を $i = 1$ から $i = m$ まで順に適用する．なお，以下における $C_0, \ldots, C_m$ は近似部分グラフの候補を表す隣接行列の集合であり，C_0 はすべての頂点と辺が未定義である $m \times m$ の 1 つの隣接行列からなるものとする．また，$A(v'_i)$ の元の数を k_i とする．

2a. C_{i-1} から 1 つの隣接行列 M を取り出し，$C_{i-1} = C_{i-1} - \{M\}$ とする．

2b. $v_i(M)$ が未定義の場合，$A(v'_i)$ 中の各頂点を $v_i(M)$ とした k_i 個の隣接行列 $M_1, \ldots, M_{k_i}$ を生成し，$C_{i-1} = C_{i-1} \cup \{M_1, \ldots M_{k_i}\}$ とし，手順 2a へ．

2c. $(v'_i, v'_j) \in E'$ を満たす任意の j について，$v_j(M)$ が定義済みであり，かつ $(v_i(M), v_j(M)) \notin E$ であれば，M を破棄し手順 2f へ．

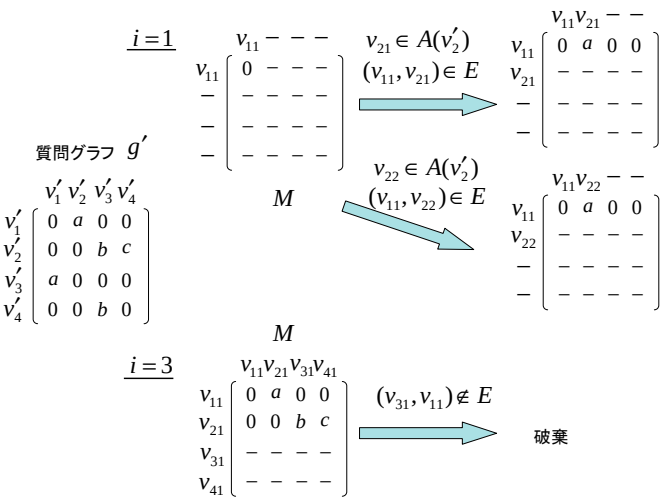

図 3: 隣接行列生成過程の例

2d. $(v'_i, v'_j) \in E'$ である j について，$v_j(M)$ が定義済みであり，かつ $(v_i(M), v_j(M)) \in E$ であれば M の (i, j) 成分にラベル $\varphi((v_i(M), v_j(M)))$ を割り当てる．

2e. $(v'_i, v'_j) \in E'$ である j について，$v_j(M)$ が未定義の場合，$A(v'_j)$ 中の頂点 v のうち $(v_i(M), v) \in E$ を満たすものを $v_j(M)$ とし，M の (i, j) 成分にラベル $\varphi((v_i(M), v_j(M)))$ を割り当て，$C_i = c_i \cup \{M\}$ とする．各 j に対して条件 $(v_i(M), v) \in E$ を満たす頂点 $v \in A(v'_j)$ が複数存在する場合，すべての組み合わせについて M' を生成し，C_i に加えるものとする．

2f. C_{i-1} が空集合であれば終了．そうでなければ，手順 2a へ．

直感的には，上記の手法は，v'_i に対応する頂点 $v_i(M)$ を決定後，手順 2c〜2e において v'_i との間に辺をもつ頂点 v'_j に基づき，M 中の j 番目の頂点 $v_j(M)$ と v'_i に接続する辺を順次決定していくものである．上記手順 2c では，質問グラフ g' における接続関係が保存されないため，候補グラフ M が破棄される．たとえば，図 3 における $i = 3$ の場合には，M 中の頂点 v_{31} が頂点 v_{21} との接続関係により定義済みとなっているが，質問グラフ g' 中の辺 (v'_3, v'_1) に対応する辺 (v_{31}, v_{11}) がデータグラフ中に存在しないため，手順 2c において M は破棄される．一方，図 3 の $i = 1$ の場合は，g' 中の辺 (v'_1, v'_2) に対応する辺がデータグラフ中に 2 つ存在するため，それぞれについての候補が手順 2e にて次の候補が生成される．

上記の手法の計算量は，質問グラフ中の任意の頂点 v'_i に対して，$(v'_i, v'_j) \notin E' (i < j)$ であるときに最悪となる．この場合，v'_i に対応する頂点を決定しても i 番目以降のいかなる頂点も決定されないため，$A(v'_j)$ 中の最大 k 個の頂点すべてが v'_j $(i < j)$ に対応する頂点の候補となる．したがって，上記の手順の計算量は $O(k^m)$ となる．ただし，辺の始点となる頂点を先に決定することで，この最悪計算量は容易に回避可能である．

5. 予備実験

本節では，頂点数 3,000，2,500，2,000，1,500 の 3 つの大きさの異なる有向連結グラフを人工的に生成し，それらをデータグラフとして用いて行った予備実験の結果を示す．以下では，それぞれを T3000，T2500，T2000，T1500 と呼ぶ．各デー

表 1: 予備実験結果

	検索時間 (ms)		問合せ時間 (ms)		回答数	
探索手法	近似	完全	近似	完全	近似	完全
T1500	870	958	56	61	37.9	27
T2000	1,013	950	133	150	21.4	27
T2500	1,425	2,083	280	367	34.9	59
T3000	1,792	2,144	548	887	19	21

タグラフは 10 種類の頂点ラベルと 10 種類の辺ラベルを用い，頂点間の辺の存在確率を 0.2%としてランダムに作成した．この場合の各グラフにおける 1 つの頂点に接続する辺の数の平均は 10〜18 程度である．

本実験では，これらのデータグラフに頂点数 5，辺ラベル数 10，頂点ラベル数 10，頂点間の辺の存在確率を 1%として同様に作成した部分グラフを 50 個埋め込み，それを質問グラフとした．辺の存在確率がデータグラフと異なるが，相対的に頂点数が少ないため，埋め込んだグラフにおける 1 つの頂点に接続する辺の数は平均で 1 となっている．なお，頂点を特徴付ける経路の長さ h は 3 とし，$k = 200$ とした場合の近似 K-NN 探索と完全 k-NN 探索による検索時間と問合せ時間，および回答数について調べた．検索時間は質問グラフを読み込んでから結果を出力するまでの時間であり，問合せ時間は近似 k-NN 探索，および完全 k-NN 探索が質問グラフすべての頂点に対して k 個のデータグラフ中の頂点を求めるために要した時間である．なお，SASH の設定値はデフォルト値を用い，実験は CPU：Pentium 4 2.8GHz, メモリ：3GB を有する計算機上（OS：WindowsXP）で C++を用いて実装して行った．

表 1 に結果を示す．探索手法の行における「完全」は完全 k-NN 探索を,「近似」は SASH による近似 k-NN 探索を意味する．なお，いずれの値も 10 回試行の平均値である．近似 k-NN 探索に関しては，SASH 内で各階層に割り当てられる頂点が試行ごとに変化するため，試行ごとに回答も変化する．表 1 から，近似 k-NN 探索の方がほとんどの場合において，短い検索時間，および問合せ時間で回答していることがわかり，その差は，データグラフの頂点数が多くなるほど大きくなる傾向にある．回答数に関しては，若干，近似 k-NN 探索のほうが少なくなる傾向があるが，これは探索精度によるものと思われる．

一方，k の値を 300 にした場合，いずれの探索方法においても問合せ時間は大きく変化しなかったが，検索時間が大幅に増加する結果となった．これは，提案手法で用いているナイーブな近似部分グラフ検索アルゴリズムに起因するものと思われる．また，T3000 では問合せ時間については近似 k-NN 探索の方が短いが，検索時間では完全 k-NN 探索の方が短くなった．これは，k を大きくし過ぎたために近似 k-NN 探索の精度が下がり，最終的には破棄される無駄な近似部分グラフが多く生成されたためと考えられる．

以上の結果から，提案手法において近似 k-NN 探索が有効に機能することが確認できたが，その一方で，近似部分グラフ検索アルゴリズムに改善の余地があることがわかった．

6. おわりに

本稿では，大規模なデータグラフから質問グラフに構造が類似する近似部分グラフを効率的に検索する手法を提案した．提案手法は，データグラフから質問グラフと一致する部分グラフを網羅的に探すのではなく，質問グラフに対する近似部分グラフを近傍構造が類似した頂点から求める．近傍構造が類似した頂点の検索は，SASH を用いることでデータグラフの頂点数 n に対して $O(\log_2 n)$ で実現されており，データグラフが大規模化した場合でも効率的に機能する．一方，近傍構造が類似した頂点から近似部分グラフを同定する手法は，現状ではナイーブな方法を採用しているため，予備実験の結果からも分かるように，今後，より効率的な方法を検討する必要がある．

謝辞

SASH のプログラムを提供していただくとともに，本研究について貴重なご意見を頂いた国立情報学研究所 客員教授 Micahael E. Houle 氏に感謝致します．

参考文献

[Chávez 01] Chavez, E., Navarro, G., Baeza-Yates, R., and Marroquin, J.L.: “Proximity Searching in Metric Spaces”, ACM Computing Surveys (CSUR), Vol.33, Issue 3, pp.273-321(2001).

[Cook 06] Cook, D.J. and Holder, L.B. (Ed.): Mining Graph Data, Weley-Interscience (2006).

[Dehen 87] Dehen, F. and Nolteimer, H.: “Voronoi trees and Clustering Problems”, Information Systems, Vol.12, No.2, pp.171–175(1987).

[Horváth] Horváth, T., Gärtner, T., and Wrobel, S.: “Cyclic Pattern Kernels for Predictive Graph Mining”, Proc. of SIGKDD 04, pp.158-167(2004).

[Houle 05] Houle, M. E. and Sakuma, J.: “Fast Approximate Similarity Search in Extremely High-Dimensional Data Sets”, Proc. of ICDE 2005, pp.619–630(2005).

[Kashima 03] Kashima, H., Tsuda, K., and Inokuchi, A.: “Marginalized Kernels Between Labeled Graphs”, Proc. of ICML-2003, pp.321–328(2003).

[Messmer 00] Messmer, B.T., and Bunke, H.: “Efficient Subgraph Isomorphism Detection: A Decomposition Approach”, IEEE trans. on Knowledge and Data Engineering, Vol.12, No.2, pp.307–323(2000).

[西村 06] 西村，片山，太田，石川：グラフの連結性に基づく Messmer らの部分同型判定手法の改良，電子情報通信学会第 17 回データ工学ワークショップ (ISSN 1347-4413)(2006).

[Washio 05] Washio, T., Kok, J. N., and De Raedt, L. (Ed.): Special issue on Advances in Mining Graphs, Trees and Sequences, Fundamenta Informaticae, IOS Press Vol.66, No.1-2 (2005).

[Yianilos 93] Yianilos, P.: “Excluding Middle Vantage Point Forests for Nearest Neighbor Search in General Metric Spaces”, Proc. of 4th ACM-SIAM Symposium on Discrete Algorithms (SODA’93), pp.311-321(1993).

[Yan 05] Yan, X, Yu, P. S., and Han, J.: “Substructure Similarity Search in Graph Database”, Proc. of SIGMOD 2005, pp.766-777(2005).