距離索引を利用したMSTの効率的発見手法

4P－4

石川 雅弘　劉 奕　大保 信夫
筑波大学 工学研究科　(株) 常磐システムエンジニアリング　筑波大学 電子・情報工学系

1999年8月6日

1 はじめに

クラスタリングはデータ発掘や知識発見における主要な問題の一つであるが、グラフの分割問題として捉えられる事も多い。例えばあるデータベースに対しデータ対間の非類似度を与える距離関数が定義されている時、各データを頂点、頂点間には距離を重みとする辺が存在すると考えるとデータベースは辺重み付き完全グラフとして捉える事ができる。この時、このグラフ上のMST(最小全域木, Minimal Spanning Tree)の構築過程は単リンク方式のクラスタリング過程とみなす事ができ、MSTの木構造表現である樹状図はクラスタの階層構造の表現と見る事ができる。

しかし画像や文字列(ゲノムシーケンス等)など多くのアプリケーションにおいて距離関数の計算コストは高くなる傾向にあり、またこのようなグラフは完全グラフである事から、PrimやKruskalの方法などの古典的なMST構築手法をそのまま適用した場合の計算コストは膨大となる。

そこで辺の重みが距離公理を満している事を前提として、距離索引を利用する事でMST構築の際に必要となる距離計算回数を削減し、効率的にMSTを発見する手法を提案する。またそのための距離索引である距離行列を導入する。

2 MSTと貪欲解法

$G = \langle V, E, \phi \rangle$ を辺重み付き無向連結グラフとする。ここで V は頂点の集合、E は辺の集合、ϕ は以下のような辺重み関数である:

$$E \subseteq V \times V$$
$$\phi : E \to \mathbb{R}$$

$|E| = \frac{|V|(|V|-1)}{2}$ が成り立つ時、すなわち全ての頂点間に辺が存在する時 G は**完全グラフ**と呼ばれる。G の**全域木** (*spanning tree*) とは次の条件を満す連結グラフ $G_{st} = \langle V, E_{st}, \phi \rangle$ である:

$$E_{st} \subseteq E \ \wedge \ |E_{st}| = |V| - 1.$$

全域木は複数存在し得るが、辺の重みの和 $(W = \sum_{e \in E_{st}} \phi(e))$ が極小となるものを G の**最小全域木** (*MST*) と呼ぶ。

MSTの構築法として良く知られているPrimやKruskalの方法はいずれも貪欲法と呼ばれる枠組で捉えられる [2]。貪欲法では空の辺集合から始まり、辺を一つづつ追加(確定)していく事でMSTを構築する。$|V| - 1$ 個の辺が確定した時点で処理は終了する。図1に貪欲法の枠組を示す。貪欲法にあって最も重要で頻繁に行なわれるのは、辺の集合(前線集合)から最小の重みを持つもの(最小辺)を選択する処理である。最小辺を選択するためには前線集合に含まれる全ての辺の重みを参照する(距離計算を行なう)必要がある。そのため対象が完全グラフでしかも距離計算コストが大きい場合には、全体の処理コストが膨大となるという問題がある。

C	–	クラスタ (頂点集合 C_V と辺集合 C_F の対)
C_V	–	クラスタに含まれる頂点の集合
C_F	–	端点が C_V と $V \backslash C_V$ にある辺の集合 (C_V の前線集合)
P	–	クラスタの集合 (G の分割)
e	–	辺 (二つの頂点 u, v と重み $\phi(u,v)$ の三つ組)
MST	–	MSTを構成する辺の集合

```
1: // クラスタ集合の初期化
2: P ← ∅
3: for all u ∈ V do
4:   C_V ← {u}
5:   C_F ← ∅
6:   for all v ∈ V do
7:     C_F.append(⟨u, v, φ(u, v)⟩)
8:   end for
9:   P.append(⟨C_V, C_F⟩)
10: end for
11: // 辺の確定とクラスタの合併による MST の構築
12: MST ← ∅
13: while |MST| < |V| − 1 do
14:   C_V, C_F ← C ← P.choose()
15:   repeat
16:     u, v, dist ← e ← C_F.findMin() // 最小辺の選択
17:   until v ∉ C_V
18:   MST.append(e)
19:   P.joinCluster(v, C) // v を含むクラスタと合併
20: end while
21: return MST
```

図1: MST構築のための貪欲法

3 アプローチ

我々のアプローチは次のような仮定に基づいている:

> MST構築に必要なのは各頂点の近傍辺のみである。

ここで**近傍辺** (*close edge*) とは近傍頂点間の辺のことである。

前線集合からは最小辺が選択されるが、これはクラスタ内の各頂点の近傍辺の一つである。したがって前線には各頂点の近傍辺のみが含まれていれば充分である。そこで始めから全ての辺を格納するのではなく、必要に応じて各頂点の近傍辺を検索し前線集合に格納する事にする。こうする事で最小辺を選択するために必要となる距離計算回数が削減できる。近傍辺の検索には距離索引を利用する。距離索引は距離空間中のデータに対し中心データ q と半径 r が与えられた時、q から距離 r 以内にある全てのデータを検索するための索引である [3]。

Efficient MST construction for Complete Metric Graphs
Masahiro Ishikawa †,
Yi Liu*,
Nobuo Ohbo‡
† Doctoral Program in Eng., University of Tsukuba
* Tokiwa System Engineering Inc.
‡ Institute of info. Sci. and Elec., University of Tsukuba

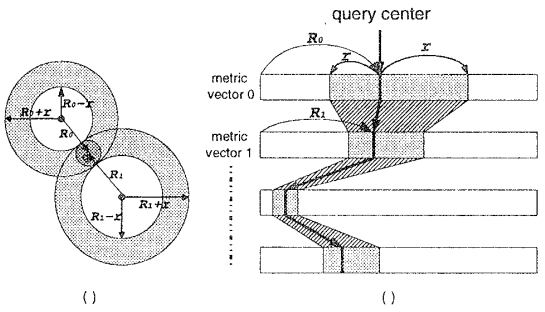

図 2: 解の候補範囲の重なりと距離行列

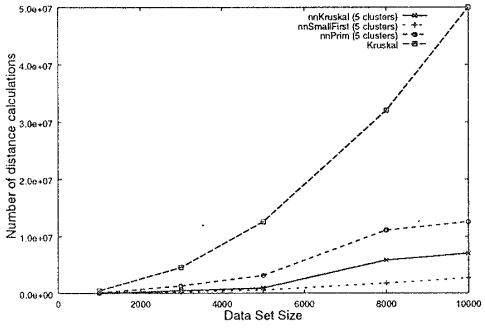

図 3: MST 発見に要した距離計算回数

4 距離行列

3 で述べたようなアプローチでは、同一頂点を中心として半径を漸時増加させた複数回の近傍検索を行なう必要が生じる。そのため既存の距離索引では同一辺のための距離計算を複数回行なってしまうため、距離を計算する辺の数は減少しても述べ数としての距離計算回数は増加してしまう可能性が高い。そこで近傍検索の仕様を変更した上で、半径を漸時増加させた時にはその差分のみを検索するような索引である**距離行列** (*metric matrix*) を考案した。これにより、重複した距離計算を抑制できる。

距離行列における近傍検索では、中心頂点 q と半径 r_k が与えられた時、q から距離 r_k 以内にある全ての頂点が検索されるが、実際には r_k 以内にはない頂点も検索される。r_k に続いて半径を $r_{k+1}(> r_k)$ として検索を行うと、q から距離 r_{k+1} 以内にある全ての頂点を含むような頂点の集合が返される。ただし半径を r_k として検索した際に返された頂点は含まない。
このように実際には近傍辺でない頂点も含むものの、検索半径の増加に応じた差分検索を行なう事で、重複した距離計算を排除している。

距離行列による検索の原理は一点からの距離を基準とした三角不等式による絞り込みである。基準頂点から他の全ての頂点への距離を求め、その配列を昇順にソートしたものを**距離配列** (*metric vector*) と呼ぶ。基準頂点から距離 R にある検索中心頂点 q について、q から半径 r 以内にある頂点の基準からの距離 r' は $R-r \leq r' \leq R+r$ であるため、配列上では連続した範囲となる。距離行列は複数の距離配列から構成され、それぞれの候補範囲に含まれる頂点集合の共通集合を取る事で、候補集合を絞り込む。このようにして得られた頂点集合は、q から距離 r 以内にある全ての頂点を含んではいるが、r 以内にない頂点も含む。
それらを含めたまま検索結果としてしまう事で、検索半径が増加した時、それまでに検索した頂点を無視して処理する事が可能となる。

図 2 に距離行列における検索 (絞り込み) のイメージを示す。図 2(a) は二つの基準点による近傍候補範囲の重なりを示しており、(b) は複数の距離配列における検索中心頂点とその近傍候補範囲を示す。

5 実験

合成データによる実験を行ない、MST 発見に要した距離計算回数を測定した。データは文献 [1] の Appendix H の方法により生成したクラスタデータであり、生成パラメータの分散を $\sigma^2 = 0.1$ とし、クラスタ数は 5 で均等サイズとした。

Kruskal の方法そのままの場合と、Prim, Kruskal の方法に距離索引を利用した場合における距離計算回数を図 3 に示す。距離索引を利用した方法では、ほぼデータ数に比例して増加しており、Kruskal の方法をそのまま適用した場合に比べてデータ数が増加した時の増加比率が小さい事が分かる。

6 まとめ

辺の重みが高コストの距離関数として与えられているような完全距離グラフにおいて、距離計算回数を削減する事で効率的に MST を構築する手法を提案した。またこのために必要な性質を備えた距離索引である距離行列を考案した。合成データによる実験を行ない、提案した手法により距離計算回数を減少できる事を確認した。距離計算回数の削減以外にも、本手法では各頂点を中心とした近傍検索を行なう事によって密度情報が収集されるというメリットがある。この情報は MST により得られるクラスタを改善するために利用できる。

今後は実データによる実験を行なう予定である。

参考文献

[1] A.K.Jain and R.C.Dubes. *Algorithm for Clustering Data.* Printice Hall, 1988.

[2] R.E.Tarjan. *Data Structure and Network Algorithm.* Society for Industrial and Applied Mathematics, 1983.

[3] 石川雅弘 能登谷淳一 陳漢雄 大保信夫. 距離索引 MI-tree. **情報処理学会 論文誌:データベース** *(印刷中)*, 1999.