2B3-3

Connectivity カーネルを利用した制約付きクラスタリング

Constrained Clustering with the Connectivity Kernel

岡部正幸[*1] 山田誠二[*2]
Masayuki Okabe Seiji Yamada

[*1]豊橋技術科学大学 Toyohashi University of Technology
[*2]国立情報学研究所 National Institute of Infomatics

This paper proposes a method of constrained clustering enhanced by the Connectivity Kernel. This method is unique because it uses constraints not only for the process of clustering but also for the calculation of distance between data. We also propose an approach to select constraints actively. Experimental results show that our method improves the clustering performance slightly higher.

1. はじめに

データ間に存在する制約を利用することでクラスタリングの精度を向上させる方法は，制約付きクラスタリング，半教師ありクラスタリングなどと呼ばれ，近年精力的に研究が行われている．制約付きクラスタリングには主に 2 つのタイプが存在する [Basu04]．一つは制約ベースのタイプで制約を満たしながら目的関数を最適化していくアプローチである．もう一つは距離ベースのタイプで制約を満たすクラスタリングを実現するために適応的にデータ間の距離関数を変化させるアプローチである．本研究では，Connectivity Kernel[Fischer03] を制約付き k-means に組み込むことで 2 つのアプローチを同時に実現する方法を提案する．

一方，制約はクラスタリングを行う前に予め分かっている場合もあるが，タスクによってはユーザとのインタラクションの過程において追加される場合もある．後者の場合，制約を与えることは一般にユーザにとって負担となる作業であるため，できるだけ精度向上の見込まれる制約を与えた方がよい．このようなアプローチは，実験計画法や能動学習において理論的に研究されているが，制約付きクラスタリングにおける制約選択方法については十分に研究されているとはいえない．本研究では，この制約選択の方法についても提案を行い，その有効性について検証する．

2. 制約付き k-means アルゴリズム

制約付きクラスタリングでは，一般的に must-link, cannot-link と呼ばれる 2 種類の制約を利用する．前者は必ず同じクラスタに属さなければならないデータペアとして与えられ，後者は必ず異なるクラスタに属さなければならないデータペアとして与えられる．本研究では，Wagstaff らが提案した COP-KMEANS と呼ばれる制約付き k-means アルゴリズム [Wagstaff01] に Connectivity Kernel を組み込んだ方法を提案する．ただし，オリジナルの COP-KMEANS はクラスタ中心をクラスタ内の重心としていたが，Kernel 計算の都合上，クラスタ中心は代表点を選択する k-medoid 型の方法を用いる．

データ集合を D，クラスタ数を K，制約集合を CS としたときの制約付きクラスタリングアルゴリズム (k-medoid-cs) を以下に示す．

連絡先: 岡部正幸，豊橋技術科学大学情報メディア基盤センター
okabe@imc.tut.ac.jp

1. クラスタ中心候補をランダムに K 個選択する．選択された候補をそれぞれ $c_i(i = 1 \sim k)$ とする．
2. 中心候補以外の各データ x_i について，x_i との距離の近い順に中心候補 c_k をソートし，c_k のクラスタメンバとの間で制約 CS に違反しない中心候補があればそのクラスタに割り振る．
3. すべてのデータを割り振り終えたら，クラスタ内距離（クラスタ中心とメンバの距離の総和）の総和 D_K^{orig} を求める．

$$D_K = \sum_{k=1}^{K} \sum_{i \in C_k} (x_i - c_k)^2$$

ここで C_k は各クラスタ集合である．

4. 中心候補以外の各データ x_i について，各中心候補 c_k と交換，つまり中心の役割を交代した場合の D_K^{tmp} を新たに計算し，$D_K^{tmp} - D_K^{orig} < 0$ となる交換の中で最も減少が大きい交換を行い，2 に戻る．$D_K^{tmp} - D_K^{orig} < 0$ となる交換が見つからない場合は終了．

2 の処理は実際には組み合わせ探索となるため，制約数が増えると計算コストがかさむ．

3. Connectivity Kernel

k-means アルゴリズムにカーネル関数を組み込むことは容易であるため，適切な Kernel 関数を利用することで元の特徴空間では生成困難なクラスタを写像先の特徴空間にて生成できる可能性がある．特徴空間の写像関数を ϕ，カーネル関数を $k(u, v) = \phi(u) \cdot \phi(v)$ とすると，写像先におけるクラスタ内距離の総和 D_K は，下記のように表せる．

$$D_K = \sum_{k=1}^{K} \sum_{i \in C_k} (\phi(x_i) - \phi(c_k))^2 \quad (1)$$

$$= \sum_{k=1}^{K} \sum_{i \in C_k} \{k(x_i, x_i) - 2k(x_i, c_k) + k(c_k, c_k)\} \quad (2)$$

本研究ではカーネルに Connectivity Kernel を用いる．このカーネルは実際にはここのカーネル値の行列として与えられ

る．このカーネルは path-based clustering に基づいて考案されたもので，データ集合から生成された近傍グラフを利用してデータ間の距離を定義し，カーネル値を計算する．具体的な計算は以下のように行う．

グラフの頂点 i, j 間を結ぶ経路の集合を P_{ij} とし，各経路を p_{ij} とする．また p_{ij} の m 番目に通る頂点と $m+1$ 番目の頂点とのユークリッド距離を $d'_{m,m+1}$ とする．このとき，i, j 間の新たな距離 d_{ij} を以下のように定義する．

$$d_{ij} = \min_{p_{ij} \in P_{ij}} \{\max_{m}\{d'_{m,m+1}\}\}$$

この式は各経路の最大距離を持つ辺のうちの最小値を求めていることを表している．直感的には頂点 i, j 間で必ず超えないといけない溝の深さの最小値を計算しているといえ，同じクラスタ内のデータ間距離は小さく，異なるクラスタ間のデータ間距離は大きくなることを想定した距離の定義となっている．

この d_{ij} を要素に持つ距離行列を D とし，下記の処理を施すことにより，半正定値となる Connectivity Kernel 行列 D^c が得られる．

$$D^c = -\frac{1}{2} Q D Q$$

ここで $Q = I_n - \frac{1}{n} e_n e_n^T, e_n = (1, 1, ..., 1)^T$ は中心化行列である．

3.1 制約を考慮したカーネル行列の計算

前節でのカーネル行列の計算は制約を考慮しないものであったが，ここでは制約を考慮したカーネル計算について述べる．具体的には以下の処理を行う．

- must-link: すべての must-link な頂点 i, j のユークリッド距離 $d'_{ij} = 0$ とする．
- cannot-link: すべての cannot-link な頂点 i, j のユークリッド距離 $d'_{ij} = \infty$ とする．つまりリンクをなくす．
- 上記 2 つの処理を行ったあと，D^c を再計算する．

以上の処理を行うことにより，データ間の距離を制約に応じて変化させることができる．

4. 能動的な制約候補の選択

制約を加えることでクラスタリング精度の向上が見込まれるが，より早く精度を向上させるには大きな効果の見込まれる制約を選択することが重要となる．本節では制約候補となるデータペアの選択方法について考える．

制約候補の選択は k-近傍グラフの頂点ペアをそのユークリッド距離でソートしたリストを用いる．Connectivity Kernel はウォード法 (Ward method) を用いて計算するがその際にこのリストを用いる．リストの最上位と最下位には既知の must-link と cannot-link がそれぞれリストされており，制約候補はそれらをのぞく頂点ペアから選択する．制約として効果が期待されるのは，頂点ペアの距離が短いにもかかわらず実は cannot-link である場合と頂点ペアの距離が長いにもかかわらず実は must-link である場合と考え，その可能性を構築されたクラスタ群において同一クラスタに属しているか，属していないかで判断する．つまり，上位の頂点ペアで頂点同士が同一クラスタに属していないもの，下位の頂点ペアで頂点同士が同一クラスタに属しているものを制約候補として選び実際に must-link か cannot-link なのかを判定する．上位と下位のペアは 2 つずつ同時に調べていき，該当するペアが先に見つかったものを制約候補とする．該当するペアが同時に見つかった場合は 2 つのうち一つをランダムに選択する．

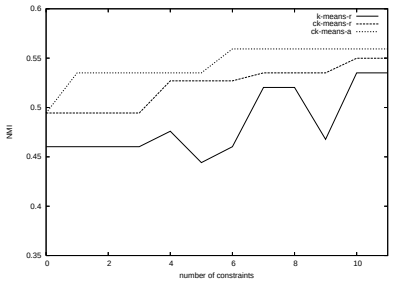

図 1: 実験結果

5. 実験

提案手法の効果を調べるため，UCI データセットの **soybean**（データ数 47，クラス数 4，属性数 35）を用いた実験を行った．

比較する方法は，以下の 3 つである．1) 制約付き k-means にランダムに制約を追加する方法 (k-means-r)，2) Connectivity Kernel を利用した制約付き k-means にランダムに制約を追加する方法 (ck-means-r)，3) Connectivity Kernel を利用した制約付き k-means に 4 章で説明した制約選択を適用した方法 (ck-means-a)，の 3 つである．

クラスタ数はクラス数と同じ値を予め与えておく．クラスタリング結果の評価は，正規化相互情報量（NMI: normalized mutual information）を用いて行う．

$$\mathrm{NMI}(C, T) = \frac{\mathrm{MI}(C, T)}{\max(\mathrm{H}(C), \mathrm{H}(T))}$$

ここで，C は生成されたクラスタ集合，T は正解クラスタ集合であり，MI は相互情報量，H はエントロピーをさす．

図 1 に実験結果を示す．提案手法の効果が若干ではあるが見て取れる．

6. まとめ

本研究では，制約付き k-means に Connectivity Kernel を適用することでクラスタ中心への割り振り時だけでなく，データ間の距離も制約に応じて変化させるクラスタリング方法を提案した．また，制約選択を行う際の方法についても提案した．実験では，それほど大きな効果は見られなかったが，今後様々なデータを用いて特性を明らかにしていく予定である．

参考文献

[Basu04] Basu, S, Bilenko, M and Mooney, R. J: A probabilistic Framework for Semi-Supervised Clustering, In *Proceedings of KDD'04* (2004)

[Fischer03] Fischer, B, Roth, V and Buhmann, M: Clustering woth Connectivity Kernel, In *Proceedings of NIPS'03* (2003)

[Wagstaff01] Wagstaff, K and et al.: Constrained K-means Clustering with Background Knowledge, In *Proceedings of ICML'01* (2001)