非線形複数クラスタリングにおけるノイズ推定

伊藤香織　(指導教員：吉田裕亮)

1　はじめに

多量のデータに内在する情報や構造を分析する手法の一つとして, クラスタリングが挙げられる. クラスタリングとは, 分類すべきデータ間に定義された類似性や距離に基づいてグループ分けを行い, 潜在する情報や構造を取り出す手法である. 実世界においては線形分離可能なデータよりも線形分離不可能なデータの方が多く存在するため, クラスタリングの非線形化が必要となる. また, 実データに対して考えるとき, すべてを正確に判別することは現実的ではない場合もある. 実際には, どのクラスタにも属さないノイズが混入していることもあり得ると考えられるためである.

このような線形分離不可能なノイズの含まれるデータに対し, 従来の手法よりも汎用性の高いクラスタリングの手法について考察する.

2　クラスタリング

クラスタリングは大きく分類すると, 階層的手法と非階層的手法の 2 つに分けられ, 非階層的手法の代表例に K-平均法がある. K-平均法は非常に有用なクラスタリング手法であるが, 反復演算を必要とする点と収束解が必ずしも目的関数を最適にするものではないという欠点がある. スペクトラルクラスタリングではクラスタリングの問題を固有値問題として定式化することによってこれらの問題を避けるアルゴリズムを構成することができる.

また K-平均法はデータを最も近いクラスタに分類するという線形なクラスタリング手法なのでデータの形によってはうまくいかない場合もある. しかし, スペクトラルクラスタリングは与えられたデータをカーネル法を用いて高次元の特徴空間上に写像してからクラスタリングを行うので, 非線形なクラスタに拡張され非線形性なクラス形状を持つデータもうまくクラスタリングすることができる.

3　ガウスカーネル

非線形なクラス形状をもつ複雑なデータを扱うためにカーネル関数を用いる. カーネル関数 $k(x, x')$ とはデータ変数の集合の 2 つの要素 x, x' に対し x, x' のそれぞれの特徴ベクトル $\phi(x)$, $\phi(x')$ どうしの内積

$$k(x, x') = \phi(x)^T \phi(x')$$

として定義される. カーネルには様々なものがあるが, その中でもガウスカーネル

$$k(x, x') = \exp(-\beta \|x - x'\|^2)$$

を用いて特徴空間上に写像した行列は相関行列と同じような振る舞いをすることが知られている. ここで $\|\cdot\|^2$ は通常のユークリッド距離の 2 乗で $\beta \in R$ は適当なパラメータである.

4　スペクトラルクラスタリング

スペクトラルクラスタリングはサンプル点をグラフ構造として考える. 各頂点がサンプル点で枝にはサンプル点同士の近さを表す重みがついているとし, 例えばサンプル点を 2 つのグループに分けると, それに伴いグラフも 2 分割される. 分割されたグループ間を結ぶ枝のことを分割のカットと呼び, このカットの重みの合計が小さくなるようにグループ分けを行う. 式であらわすと以下のようになる.

$$\min_{\beta} \sum_{i,j} K_{i,j}(\beta_i - \beta_j)^2 = \beta^T P \beta, \quad \beta_i = \pm 1$$

ここで P は対角行列 Λ を $\Lambda_{ii} = \sum_{j=1}^{n} K_{i,j}$ として $P = \Lambda - K$ と書ける. β は 2 値ベクトルという制限がある. これは整数計画化問題と呼ばれ, 一般には解くのが困難である.

そこで, 整数という制約を取り払って任意のベクトルに ${}^T\beta\lambda\beta = 1$ という条件のもと, 制約を緩めることにより推定を行うことになる. この場合, 最小固有値 0 が存在するが, これはすべてのサンプルを 1 つにまとめてしまうという意味のない解のため, 実際には 2 番目以降の固有ベクトルの成分符号に基づいてクラスタリングを行う.

5　データ間距離の計算

カーネル法では非線形なデータを一度高次元の特徴空間上に表現し, 写像することで解析しやすいデータに変換し, その特徴空間上で線形なモデルを組み立て問題を解く. このとき, 特徴空間上での内積をカーネル関数を用いて計算することにより計算量を抑えることができるという利点がある.

各クラスタの重心とサンプルデータの距離を測るために距離 d_i は以下のようにカーネルを用いて特徴空間上で計算する.

$$\begin{aligned} d_i &= \|x_i - \mu\|^2 \\ &= K(x_i, x_i) - \frac{2}{n} \sum_{j=1}^{n} K(x_i, x_j) \\ &\quad + \frac{1}{n}^2 \sum_{j=1}^{n} \sum_{l=1}^{n} K(x_j, x_l) \end{aligned}$$

6　バギング

決して精度の高くない学習器 M 個と N 個のデータがあるとする. 学習器 h_i はある問題 x に対して Yes No で判別を行うことができ

$$h_i(x) = \begin{cases} 1 & (yes), \\ -1 & (no). \end{cases}$$

を出力する. N 個のデータに対しそれぞれを返す結果が異なるような学習器 h_i を h_1 から h_M まで作る.

これらの学習器にデータを使って判別結果をいくつも作り, 実際に判別器として用いるときは多数決をとる.

$$Answer = \begin{cases} yes & (y \geq 0), \\ no & (y < 0). \end{cases}$$

7 提案手法

まずデータを与え, そのデータから与えられるカーネル行列に非線形にクラスタリングする. クラス分けされた各クラスタに対して新たなカーネルを用意しカーネル行列を計算する. カーネルによって特徴空間上に写像されたデータに対し各クラスタの重心とデータの距離を計算し, その重心から一定の距離より遠いデータをノイズとする.

本研究では, 以下のような流れでクラスタリングを行うことを提案する.

1. サンプルデータのガウスカーネル行列

$$K = \begin{pmatrix} k(x_1, x_1) & k(x_2, x_1) & \dots & k(x_n, x_1) \\ k(x_1, x_2) & k(x_2, x_2) & \dots & k(x_n, x_2) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_1, x_n) & k(x_2, x_n) & \dots & k(x_n, x_n) \end{pmatrix}$$

   を構成し, そのスペクトルクラスタリングを行う.

2. クラス分けされたうちの各クラスタに対し, 適当なパラメータを選び, もう一度ガウスカーネル行列を計算する.

3. この新たなガウスカーネル行列を用いクラスタの重心とデータの距離を計算し, その値がある閾値より大きいデータをノイズと推定する.

4. ノイズ推定操作を複数回行いノイズ部分と判断されたデータを多数決をとって最終的な判別を行いノイズとノイズ外データを抽出する.

8 実験例

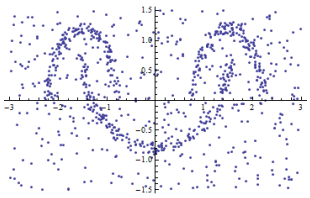

図 1: サンプルデータ

図 1 のような, 線形で分けることのできない 3 つの群からなるサンプルデータを用意する. 150 個ずつの 2 つの群と 200 個の 1 つの群に, ノイズとして一様乱数 400 個を加えた, 計 900 個のサンプルデータとなっている.

スペクトラルクラスタリング実行

スペクトラルクラスタリングを実行した結果, 図 2 のようにノイズを含め非線形な 3 つのクラスに分かれた. このときパラメータ β の値は 1200 である.

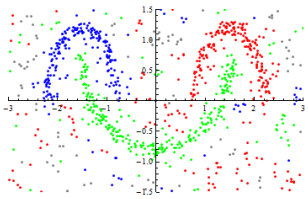

図 2: スペクトラルクラスタリングの実行結果

ノイズ推定

3 つのうちの 1 つのクラスタに対しデータ間距離を計算し, ある閾値より大きいものをノイズとした. この操作をパラメータを変えて複数回行い判別されたいくつものデータに対し多数決をとる.

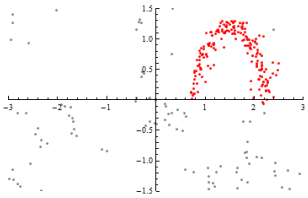

図 3: ノイズの抽出結果

この例ではパラメータを 6 つ用意し, 閾値は 1 と固定しノイズ推定を行った. また残り 2 つのクラスタに対しても同様の操作を施す.

9 実験結果

最終的には以下のような良好な判別結果が得られた.

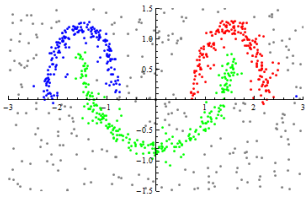

図 4: 実験結果

10 まとめ

ノイズを含む複雑な複数クラスタを含むサンプルデータを非線形にクラスタリングし, ノイズ推定を行うことができた.

今後の課題として適切なパラメータ β の導出が挙げられる. β の値は実際のクラスタリングの結果を目で見て判断する必要があるので最適なクラスタリング結果を返すような β の評価式を設定したい.

参考文献

[1] 赤沼昭太郎, カーネル多変量解析〜非線形データの新しい展開〜, 岩波書店, 東京, 2009.

[2] 西田英郎, クラスター分析とその応用, 内田老鶴圃, 東京, 1988.

[3] クラス外ノイズを考慮したスペクトラルクラスタリング, お茶の水女子大学情報科学科論文, 2012.