

第 11 回「最尤推定とクラスタリング」

工学部 37021404 中村裕大

1. ポスタリゼーションがどのようなアルゴリズムで実装されているのか論ぜよ。

今回は授業で触れられた k-means 法を用いた N 色ポスタリゼーションについて、アルゴリズムがどのように構築されているか論ずる。

ポスタリゼーションとは

- 画像処理の一つで、色や階調を減らして**画像を単純化する方法**
- 元の画像を一定の階調レベルに分割し、各領域内で最も代表的な色や値を使用して再構築

k-means 法とは

- データをクラスタリングするための単純かつ効果的な以下のようなアルゴリズム
 - I. k 個の初期中心点をランダムに選択
 - II. 各データポイントを、最も近い中心点に割り当てる
 - III. 各クラスターの中心点を、クラスターに所属するデータの平均位置に更新

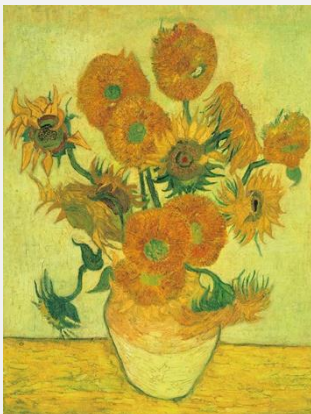
上記より k-means 法を用いて N 色ポスタリゼーションのアルゴリズムを実装してみる。

N-means ポスタリゼーションアルゴリズム

- I. 処理する画像からランダムに N 個の色を選択
- II. 画像中の全てのピクセルを最も近い色(RGB 値)に割り当てる

Python で実装してみた結果

Git: https://github.com/KameKingdom/-----11-/blob/main/N_means_postalization.py



元画像



N=2

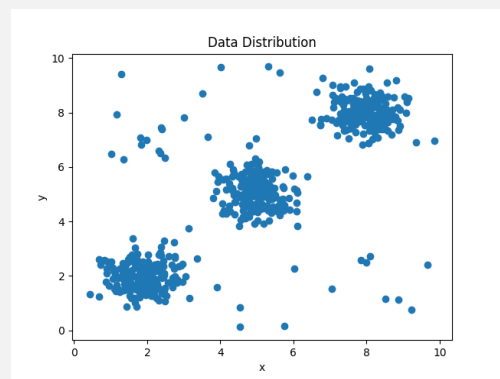


N=5

2. クラスタが存在しそうな二次元データを用意し、条件を変えて、クラスタリング処理を実行して 収束状況や性能について議論せよ。

今回評価するクラスタリング処理はプログラムより「EM アルゴリズム」と「SGD アルゴリズム」の2つである。評価方法は「収束速度」「分離度」「外れ値への頑健性」を主軸とする。

データ：気象庁の[サイト](#)から昨日(2023 / 6 / 29)の世界の気候データを取得し、3 か国のデータを csv で保存し、平均気温と降水量(二次元データ)でクラスタリングを試す。
結果：悲惨な分布でクラスタリングに適しておらず[プログラム](#)で作成することを決意



(人工的に)準備した二次元データ

収束速度 (Convergence speed) :

$$\text{convergence_speed} = \frac{\sum_{i=1}^{n-1} (\text{history}[i+1] - \text{history}[i])}{n-1}$$

分離度 (Separation degree) :

$$\text{separation_degree} = \frac{\sum_{i=1}^k \|\mu[i+1] - \mu[i]\|}{k}$$

外れ値に対する頑健性 (Robustness to outliers) :

$$\text{robustness_to_outliers} = \frac{\sum_{i=1}^k \text{std}(\|\text{faithful} - \mu[i]\|)}{k}$$

評価方法

評価方法について

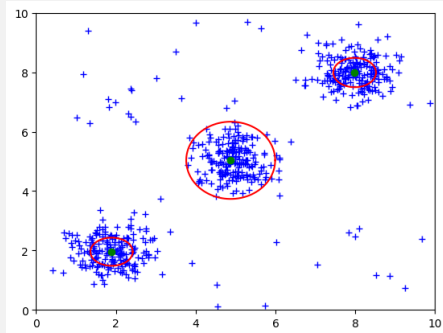
- 収束速度：変化量の絶対値の平均の総和、変化量が少ないほど速いと判断
- 分離度：クラスタの重心のユークリッド距離を計算
- 外れ値に対する頑健性：各データとクラスタの重心の距離の標準偏差の平均

EMアルゴリズム			
試行回数	収束速度	分離度	外れ値に関する頑健性
1	1.00E-17	4.28768078	2.73469451
2	1.67E-17	4.28768078	2.73469451
3	1.92E-17	6.430447887	2.73469451
4	1.78E-17	4.28768078	2.73469451
5	1.86E-17	6.43086642	2.73469451
平均	1.65E-17	5.14E+00	2.73E+00

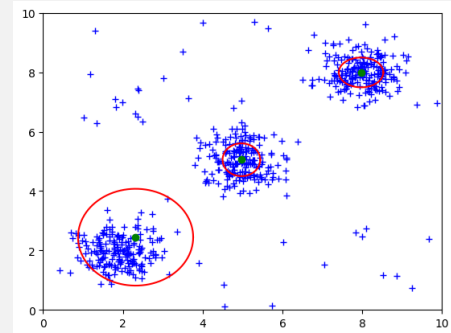
SGDアルゴリズム			
試行回数	収束速度	分離度	外れ値に関する頑健性
1	2.79E+00	4.287668808	2.734691542
2	2.44E+00	3.959392837	2.625563237
3	2.72667537	6.430446263	2.734694344
4	2.265105369	3.959373127	2.62557423
5	2.220959397	5.818593076	2.625567348
平均	2.49E+00	4.89E+00	2.67E+00

収束速度、分離度は EM アルゴリズムの方が優れており、外れ値への頑健性は僅かな差で SGD アルゴリズムが優れていた。実行速度も EM アルゴリズムの方が速かったので、総合的に EM アルゴリズムの方が高い評価値を得た。

クラスタリングの実験結果



EM アルゴリズム



SGD アルゴリズム

条件の変更

(ア) クラスタ数の変更

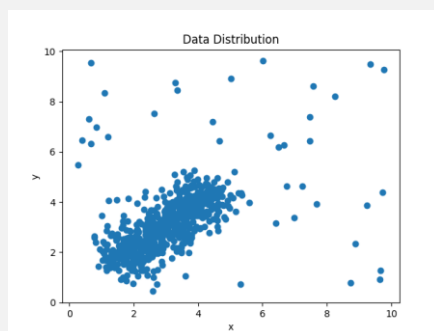
クラスタ数の変更は学習時間とページ数の関係上「EM アルゴリズム」のみの結果を示す(各クラスタ数のデータは 100 回の試行結果の平均値)。以下のように**分離度**と**外れ値に関する頑健性**は当然であるが、クラスタ数が増加するにつれ減少する結果となった。

EMアルゴリズム			
クラスタ数	収束速度	分離度	外れ値に関する頑健性
2	1.45E-17	6.223242892	2.587036445
3	2.06E-17	6.43086642	2.73469451
4	1.20E-16	4.270906165	2.487921624
5	1.94E-16	4.347996955	2.336287041
6	2.00E-16	4.748620083	2.092220804
7	1.99E-16	3.771083137	2.037801554
8	2.14E-16	3.887445546	2.056078439
9	2.05E-16	3.531163023	1.953574927
10	1.84E-16	3.451320671	1.92594313

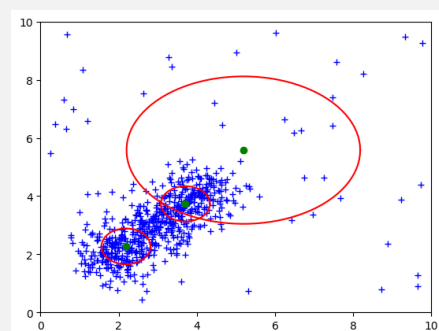
EM アルゴリズムのクラスタ数による変化

(イ) データセットの変更

前回のデータセットと同様の要素数 3 でデータ間の差を曖昧にし、以下のようなクラスタの判別が少し難化したものを使用して実験を行った。



変更後の二次元データ



n = 3

EMアルゴリズム			
クラスタ数	収束速度	分離度	外れ値に関する頑健性
2	6.07E-17	3.461662301	1.61783836
3	2.13E-16	3.3734364	1.611431936
4	1.61E-16	2.609732724	1.594136315
5	2.53E-16	3.065385301	1.499562919
6	2.78E-16	1.227769456	1.534537401

データ変更後の EM アルゴリズムのクラスタ数による変化

予想通り収束速度が遅くなり、分離度も低くなる結果となった。予想と反した結果を出した外れ値に関しては、上記の図のように巨大な分布を持つ円による影響で小さくなっていると考えられる。

参考文献

- [1] ceptree(2017)「正規分布間の KL ダイバージェンス」
(<https://qiita.com/ceptree/items/9a473b5163d5655420e8>)

- [2] g-k(2019)「k-means 法を理解する」(<https://qiita.com/g-k/items/0d5d22a12a4507ecbf11>)

- [3] 片寄晴弘(2023)「EM アルゴリズムによる GMM サンプル」
(https://colab.research.google.com/drive/1NC2Sc-0cT6ftMM-K827YYU_gfDZTtZ9w?usp=sharing#scrollTo=XNNNNF8LDw8F)

- [4] 片寄晴弘(2023)「音楽数理情報処理の技術 3」
(<https://crestmuse.jp/klab/lecture/mi/chap11.pdf>)

作成資料

- [1] GitHub: (<https://github.com/KameKingdom/-----11->)