本レジュメの最後に付録として Appendix のページを添付している。適宜参照されたい。

9 章前書きは読み合わせ。

今回はデータ集合ベクトルを K 個のグループにクラスタリングするアルゴリズムを取り扱う。アルゴリズムの思想としては、データ空間上におけるデータ間の距離を計算し[*1]、これを評価することで各データが所属するクラスターを見出す、というものである。

はじめに一般的なアルゴリズムの中身、フレームワークを導入する。そのあとで実際の間欠泉データ[*2]を用いてアルゴリズムの動き方について具体例を確かめる。

さらにより一般的に拡張された K-means クラスタリングのアルゴリズムの例をいくつか紹介する[*3]。

最後に K-means アルゴリズムを画像分割と画像圧縮に適用した例を見る。そして画像分割を細かくすること[*4]と画像圧縮を大きくすることの間にはトレードオフの関係があることを見る。このトレードオフ関係を見るためには元画像と圧縮画像の情報を送るために、どれだけビット数が必要になるか評価、比較してやれば良い。今回画像分割と画像圧縮はキャンセルで。次回お願いします。

1 K-means クラスタリング

本セクションの流れは次の通り。

まず K-means[*5]アルゴリズムについて、そのアルゴリズムの中身、フレームワークを理解する。そのために最小化すべき目的関数を定義し、クラスタリングの初期配置から収束までの手続きに関する代数操作を追うことにする。

そして K-means アルゴリズムを具体的な事例（間欠泉データ）に適用し、アルゴリズムの振る舞いを視覚的に理解する。データ点が分類されていく様子とそれに伴う目的関数の変動を確かめる。

続いて、より一般的なクラスタリングのアルゴリズム手法（K-medoids アルゴリズム）についても取り扱う。これは目的関数の定義においてユークリッド距離ではない非類似度を導入したものとなっている。

最後に K-means クラスタリングの問題点を整理し、この問題を解決するための手法を簡単に紹介する。

それでは最初に K-means クラスタリングのアルゴリズムを定式化するために、いくつか必要な量を定義しよう。詳細は教科書 140 ページ中段に譲るが、結論として次の目的関数 J を最小化してやればいいことが知られている。

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \tag{1}$$

ここで $\boldsymbol{x}_n$ は各データ点、$\boldsymbol{\mu}_k$ はクラスター k の中心、r_{nk} は 2 値指示変数である。そしてこの J を最小化するような $\{r_{nk}\}, \{\boldsymbol{\mu}_k\}$ の値を求めることがゴールとなる。

J を最小化し $\{r_{nk}\}, \{\boldsymbol{\mu}_k\}$ の値を求めるアルゴリズムは次の通り。

・$\boldsymbol{\mu}_k$ の初期値を選ぶ[*6]

[*1] 厳密に言うとデータ間の距離ではなくて、データと各クラスター中心との距離を計算することになる。ちなみに、ここで言う距離はユークリッド距離に限らない一般的な「距離」である。

[*2] 詳細は上巻付録 A を参照。でも今回はあまりデータの中身については気にしなくて良さそう。

[*3] この段階で断っておくと、今回輪講においては、アルゴリズムの高速化スキーム及びオンライン版手法については読み合わせにとどめる。

[*4] つまり画質を良くすること。

[*5] K-means、つまり「平均」と呼ばれる所以は後ほど説明する。

[*6] $\boldsymbol{\mu}_k$ の初期値を「適切に」選ぶ方法は不明。今回は次のステップ以降について詳しく議論を行う。

・$\boldsymbol{\mu}_k$ を固定しつつ r_{nk} を動かして J を最小化する[*7]
・r_{nk} を固定しつつ $\boldsymbol{\mu}_k$ を動かして J を最小化する[*8]

上で説明したアルゴリズムを代数操作として理解してみよう。まず r_{nk} に関する最大化を考える。今回、説明のために簡単な例を設定してみる（See App. 【r_{nk} に関する J の最小化】）。そのあとで教科書の説明を読み合わせる。結論として、我々は各データ点のクラスターへの割り当てについて次の式を得る。

$$r_{nk} = \begin{cases} 1 & \text{when } k = \arg\min_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

次に $\boldsymbol{\mu}_k$ に関する最大化を考える[*9]。いま J は $\boldsymbol{\mu}_k$ について二次関数となっており、次のように J を $\boldsymbol{\mu}_k$ で偏微分しイコール 0 とすることで簡単に最小化を実行できる。

$$\begin{aligned} \frac{\partial J}{\partial \boldsymbol{\mu}_k} &= \frac{\partial}{\partial \boldsymbol{\mu}_k} \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \\ &= \frac{\partial}{\partial \boldsymbol{\mu}_k} \sum_{n=1}^{N} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \ (\because\ k = k \text{ 以外は消える。}) \\ &= -\sum_{n=1}^{N} r_{nk} 2(\boldsymbol{x}_n - \boldsymbol{\mu}_k) \end{aligned}$$

ここで $\frac{\partial J}{\partial \boldsymbol{\mu}_k} = 0$ とすると

$$\begin{aligned} \sum_{n=1}^{N} r_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) &= 0 \\ \sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n &= \sum_{n=1}^{N} r_{nk}\boldsymbol{\mu}_k \\ \boldsymbol{\mu}_k &= \frac{\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n}{\sum_{n=1}^{N} r_{nk}} \end{aligned} \tag{3}$$

この式の分母は k 番目のクラスターに割り当てられたデータの数に等しい。それゆえ、式 3 において $\boldsymbol{\mu}_k$ は k 番目のクラスターに割り当てられたデータの平均値[*10]として解釈することができる。このように平均値として解釈できる側面を持つことから、本アルゴリズムには”K-means”という名前が付いているのである。

まとめると、この K-means クラスタリングはデータ点のクラスターの割り当てとクラスター平均の計算を繰り返すアルゴリズムと言える。そしてこのアルゴリズムは有限の回数で収束することが知られており、以下でそれを示す。ごめん。わからなかった。

次にこの K-means クラスタリングを実際のデータ（間欠泉データ）に適用した例を取り扱う。図 9.1 を読み合わせよう。ポイントは各サブプロットごとにアルゴリズムにおける位置づけを理解することである。サブプロット (a) から (i) についてそれらの位置づけを記しておく：
・(a) $\boldsymbol{\mu}_k$ の初期値を設定
・(b) 各データ点をそれらに近い $\boldsymbol{\mu}_k$ を平均とするクラスターに割り当て

[*7] このステップを EM アルゴリズムにおける E（expectation）ステップと呼ぶ。
[*8] このステップを M（maximization）ステップと呼ぶ。
[*9] もちろん r_{nk} は固定されている。
[*10] あるいはデータベクトルの重心として解釈してもよい。

・(c) $\boldsymbol{\mu}_k$ の再計算
・(d) 各データ点をそれらに近い $\boldsymbol{\mu}_k$ を平均[*11]とするクラスターに割り当て
・(e) $\boldsymbol{\mu}_k$ の再計算
・(f) 各データ点をそれらに近い $\boldsymbol{\mu}_k$ を平均[*12]とするクラスターに割り当て
・(g) $\boldsymbol{\mu}_k$ の再計算
・(h) 各データ点をそれらに近い $\boldsymbol{\mu}_k$ を平均[*13]とするクラスターに割り当て
・(i) $\boldsymbol{\mu}_k$ の再計算

コスト関数 J の振る舞いを図 9.2 で確認しよう。ここで疑問。3 回目の M ステップで収束し、最後（4 回目）の EM ステップでは点の割り当てもクラスター平均も変化しない、とある。しかし、図 9.1 において (g) から (h) に行くとき、サブプロット中央部分の青点が 1 つ赤点に変化しているように見える。これは前半の「点の割り当てもクラスター平均も変化しない」という主張と矛盾しているのではないだろうか。

初期値の選択、アルゴリズムの高速化スキーム、そしてオンライン確率アルゴリズムに関する内容は読み合わせとする。

ここでデータ点とクラスター平均までの距離としてユークリッド距離を用いない、より一般的なアルゴリズムについて紹介する。ユークリッド距離を用いる場合の不利な点として、データがカテゴリカルデータの場合は「距離」という値に意味がなくなってしまうということがあげられる[*14]。

ここからは我々はデータ点 $\boldsymbol{x}, \boldsymbol{x}'$ に対して「距離」とは異なる一般的な非類似度 $\mathcal{V}(\boldsymbol{x}, \boldsymbol{x}')$ を導入し、次の目的関数 $\tilde{J}$ を最小化することを考える。

$$\tilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(\boldsymbol{x}, \boldsymbol{\mu}_k) \tag{4}$$

これは K-medoids アルゴリズムと呼ばれるアルゴリズムを与える。先の K-means アルゴリズムと同様に、E ステップにおいては $\boldsymbol{\mu}_k$ を固定し $\mathcal{V}(\boldsymbol{x}, \boldsymbol{\mu}_k)$ が最小となるようにデータ点の割り当てを行う。M ステップが何言いたいのかさっぱり。

最後に K-means アルゴリズムの問題点を述べておく。各繰り返しにおいて、データ点は「必ず」1 つの（最も近い）クラスターに割り当てられる[*15]ことから、割り当てられたクラスターとは別に距離が近いクラスターも存在しうる。このような場合、確率的アプローチの観点からすると、必ずしも最適なクラスター割り当てとは限らない。そのような確率的な不確定性も考慮した定式化[*16]については次の節で取り扱うことにする。

1.1 画像分割と画像圧縮

次回お願いします。

[*11] (c) で計算したクラスターの中心に変わっている。
[*12] (e) で計算したクラスターの中心に変わっている。
[*13] (g) で計算したクラスターの中心に変わっている。
[*14] 教科書に 2.3.7 節を参照するようにコメントあるが、イマイチ関連性がわからなかった。2.3.7 節に書いてある内容はスチューデント分布がガウス分布に対して robust であるということである。
[*15] ハード割り当て、と呼ぶ。
[*16] ソフト割り当て、と呼ぶ。

App. r_{nk} に関する J の最小化

簡単な設定（$K=2$, $r_{nk} \in \{0, 1\}$; $k=1,2$, $N=10$; $n=1,\cdots,10$）において次の2点を理解する。

✓ J を最小化するための手続き

✓ データのクラスター割り当てに関する帰結

今回の設定から目的関数 J は次のように書ける。

$$J = \sum_{n=1}^{10} \sum_{k=1}^{2} r_{nk} \|x_n - \mu_k\|^2$$

$$= \sum_{k=1}^{2} r_{1k} \|x_1 - \mu_k\|^2 + \cdots + \sum_{k=1}^{2} r_{10,k} \|x_{10} - \mu_k\|^2$$

$$= r_{11} \|x_1 - \mu_1\|^2 + r_{12} \|x_1 - \mu_2\|^2 \quad \leftarrow n=1$$

$$+ \quad \cdots\cdots$$

$$+ r_{10,1} \|x_{10} - \mu_1\|^2 + r_{10,2} \|x_{10} - \mu_2\|^2 \quad \leftarrow n=10$$

各 n について赤 or 青のいずれかが残る。

いま J を最小化するには次の手続きに従えばよい。

・各 n についてノルム $\|x_n - \mu_1\|^2$ と $\|x_n - \mu_2\|^2$ を比較

⇓

・ノルムが小さいほうの k（1 or 2）を特定

⇓

・その k を下付き文字に持つ r_{nk} を 1 とすればOK

またこの手続きは結局のところ次を意味する。

・クラスター1に近いデータ点はクラスター1に割り当て

・クラスター2に近いデータ点はクラスター2に割り当て