9.1 k-means クラスタリング

平成 28 年 9 月 11 日

概 要

PRML の「9.1 k-means クラスタリング」についての実装と考察

目 次

1　問題設定

K-means クラスタリングで分類を行う.

2　アルゴリズム

まず, クラスターの中心であるプロトタイプ $\boldsymbol{\mu}_k$ $(k = 1, \ldots, K)$ を導入する.
K-means アルゴリズムでは, ベクトルの集合 $\boldsymbol{\mu}_k$ だけでなく全データが点をうまく各クラスターに対応させて, 各データ点から対応する μ_k への二乗距離の総和を最小にすることが目標である.
そのためには, 目標関数 J

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \quad (9.1)$$

を最小化する. ここで r_{nk} は 1-of-k 符号化法を用いて

$$r_{nk} = \begin{cases} 1 & k = argmin_j \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \text{のとき} \\ 0 & otherwise \end{cases} \quad (9.2)$$

とすればよいことは明らかである.
また, r_{nk} を固定して, J を $\boldsymbol{\mu}_k$ で微分すると

$$2 \sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) = \mathbf{0} \quad (9.3)$$

となり,

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}} \quad (9.4)$$

を得る.

K-means アルゴリズム

1. まず, K 個のプロトタイプ μ_k を導入する.
2. 各データ点に対し最も近いプロトタイプを探し, そのクラスターに割り当てる.
3. プロトタイプは, そのクラスターに割り当てられているデータ点の平均で, 再度プロトタイプを計算する.
4. 2,3 を繰り返し, データ点のクラスターへの再割り当てが発生しなくなった時点で終了する.

3　コード

K-means クラスタリングによる分類のコード (K-means.py).

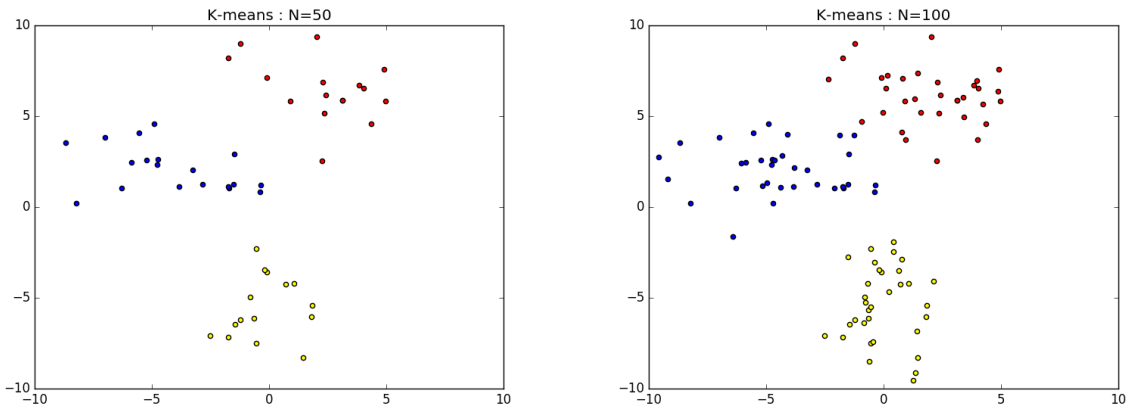
```
mu=np.array([[3,3],[-3,3],[0,-3]])
r=np.ones((N,K))
dist=np.zeros((N,K))
diff=r

while norm(diff)!=0:
```

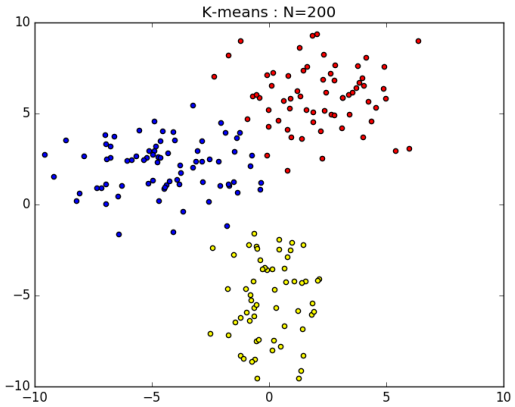
```
  diff=r
  print(norm(diff),mu[0,:],mu[1,:],mu[2,:])
  for n in range(N):
    for k in range(K):
      dist[n,k]=norm(x[n,:]-mu[k,:])
  r=np.zeros((N,K))
  num=np.zeros(K)
  for n in range(N):
    near=np.argmin(dist[n,:])
    for k in range(K):
      if k==near:
        r[n,k]=1
        num[k]+=1
  diff-=r

  mu=np.zeros((K,2))
  for n in range(N):
    for k in range(K):
      mu[k,:]+=r[n,k]*x[n,:]

  for k in range(K):
    mu[k,:]/=num[k]

C=[[],[],[]]
for n in range(N):
  for k in range(K):
    if r[n,k]==1:
      C[k].append(n)
```

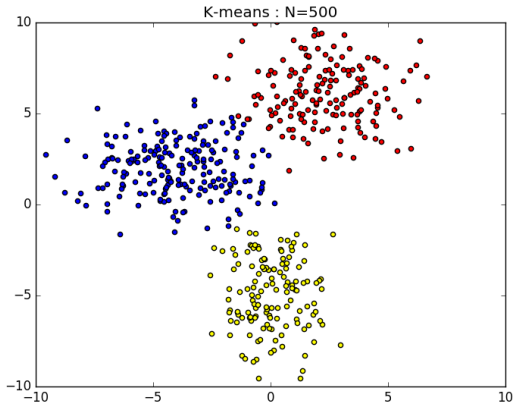
4 結果

3 クラス分類を行った.

図 1: $N = 50, 100, 200, 500$

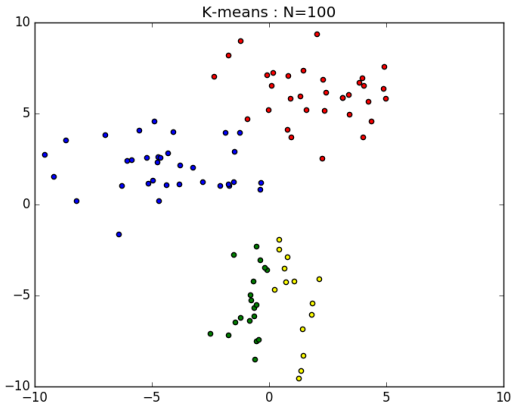


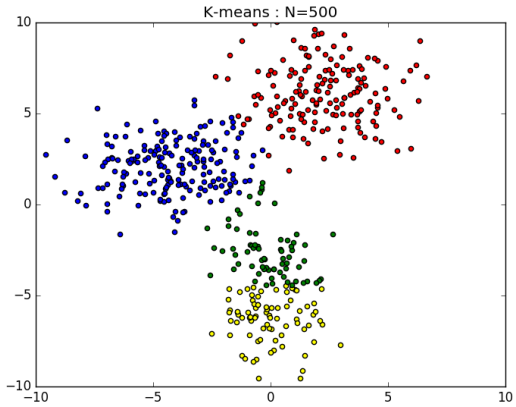




4 クラス分類を行った.(初期プロトタイプ $(3,3),(3,-3),(-3,3),(-3,-3)$)

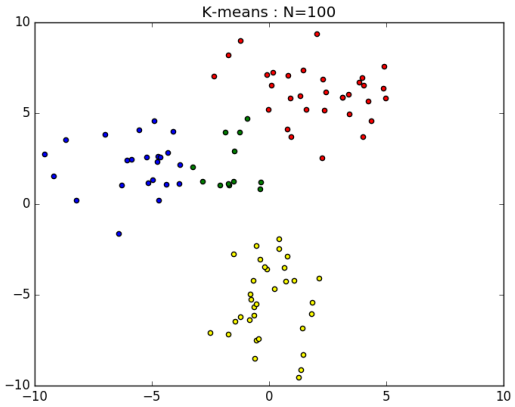
図 2: $N = 100, 500$

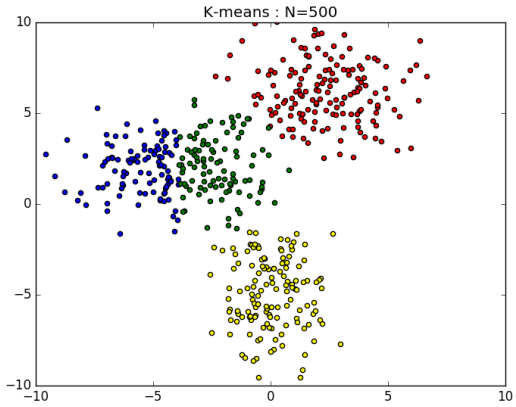




初期プロトタイプを変えてみた.(初期プロトタイプ $(3,3),(3,-3),(0,-3),(0,0)$)

図 3: $N = 100, 500$





5　まとめ

N=500 のとき, r についての収束は, 38.7298334621 22.360679775 22.360679775 22.360679775 22.360679775 0.0 と 6 回ほどで収束した. 結構速く分類ができることが分かった.
初期プロトタイプも大事で, 大きく結果が変わることも分かった.
クラスター数が元々わかっていて, 各クラスターの位置がおおよそわかっているときには使いやすいと思う.