k-means Clustering

1. 概要
2. アルゴリズムと距離
3. 例

概要

特徴

- 教師なし学習のクラスタリングの一種、非階層型クラスタリング手法
- k個のクラスタの平均(重心とすることが多い)をとることから，k-meansと称された。
  - J. MacQueen: Some methods for classification and analysis of multivariate observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press. pp. 281–297.
- 大域的解を得られる保証がなく，局所解を取る場合がある。

定式化

Description [edit]

Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, ⋯, $\mathbf{x}_n$), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k ($\leq n$) sets $\mathbf{S} = \{S_1, S_2, \cdots, S_k\}$ so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

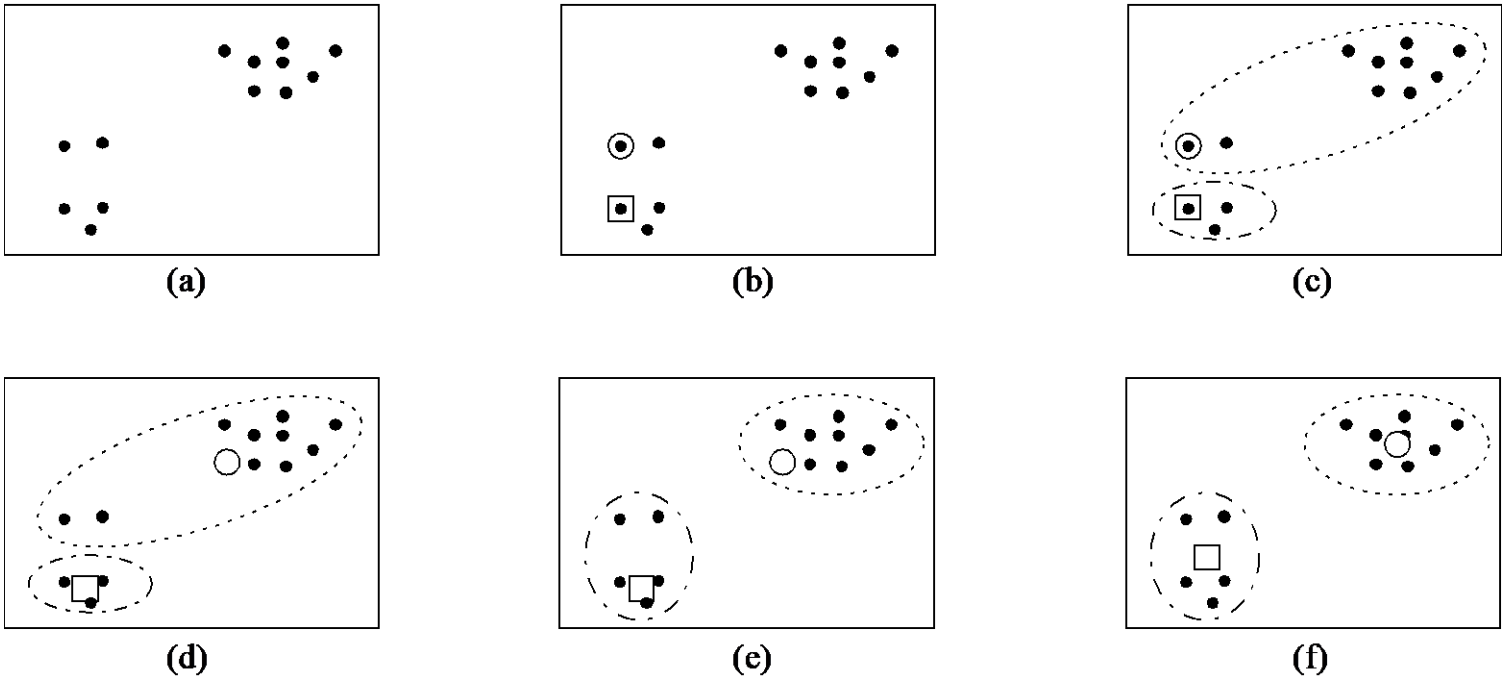
where $\boldsymbol{\mu}_i$ is the mean of points in S_i. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\mathbf{x},\mathbf{y} \in S_i} \|\mathbf{x} - \mathbf{y}\|^2$$

The Equivalence can be deduced from identity $\sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \sum_{\mathbf{x} \neq \mathbf{y} \in S_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\boldsymbol{\mu}_i - \mathbf{y})$. Because the total variance is constant, this is also equivalent to maximizing the squared deviations between points in *different* clusters (between-cluster sum of squares, BCSS).[1]

Quatatioin: https://en.wikipedia.org/wiki/K-means_clustering

アルゴリズムの概要



(a) 複数のデータ(黒丸)が得られたとする。各データのクラスは未定である。また，クラスター数は2とする。

(b) 2つのクラスタの中心(白抜きの丸(クラス1)と四角(クラス2))の初期値を，例えば図に示すデータの位置とする。

(c) 他のデータはそれぞれ，近い方のクラスタに属するものとする。2つの楕円(破線)は属するクラスタを表すが，この線自体は概念的なもので実際にはこの線は無く，実際には属するクラスの値(1か2)がデータに与えられる。

(d) クラス1，2，それぞれのクラスのデータの位置に基づき，改めてクラスタ中心を計算し，その位置にクラスタ中心が移動する。この際のクラスタ中心の計算は，重心を用いることが多い。

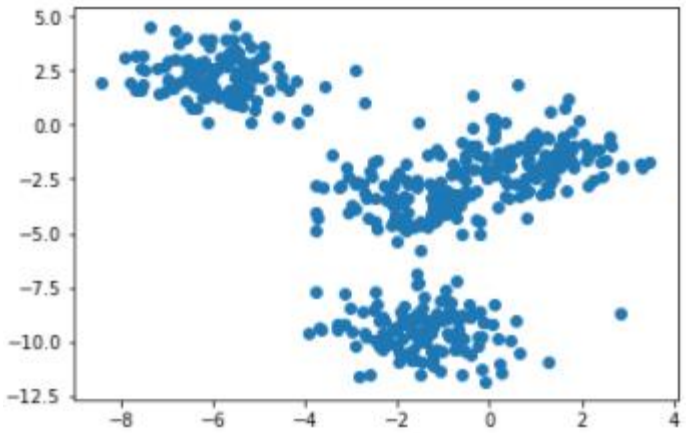
(e) (c)を実行，この結果，幾つかのデータは属するクラスが変わる。

(f) (d)を実行。この結果，クラスタ中心位置が変わる。クラスタ中心位置が収束するまで，この一連の操作と計算を繰り返す。

例　make_blobsによる練習

- **用いるパッケージ**
  - 下記の通り
- **演習の狙い**
  - グループ数4のデータを発生，
  - 見た目は3つとも4つとも見える。

2.3.2 K-means
http://scikit-learn.org/stable/modules/clustering.html
sklearn.cluster.Kmeans
http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

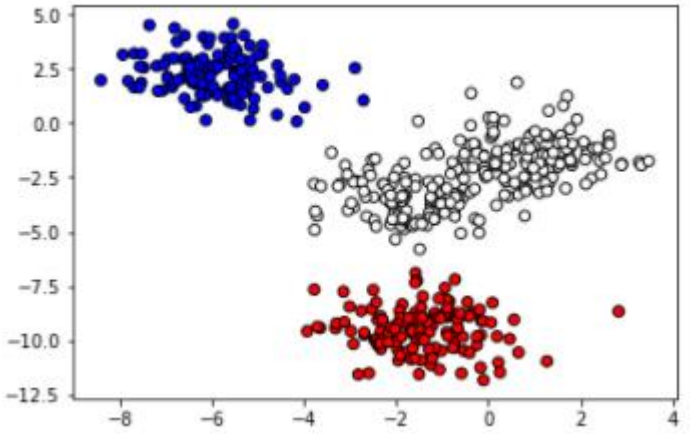


kMeans_Blobs

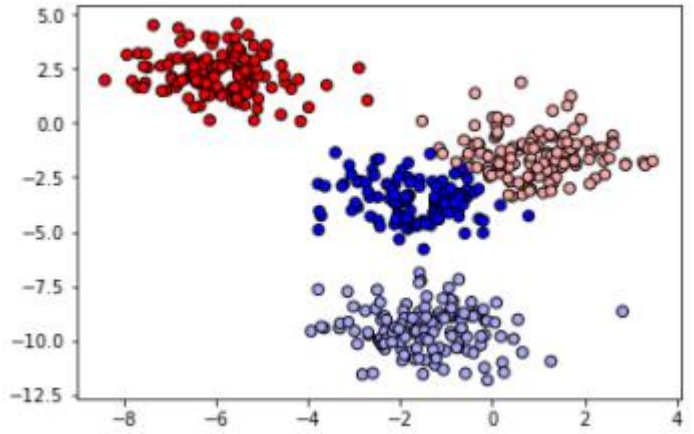
```
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs
X, y = make_blobs(n_samples=600, n_features=2, centers=4, cluster_std=1.0, random_state=2)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/6, random_state=0)
plt.scatter(X_train[:,0], X_train[:,1])
```

例　make_blobsによる練習

- クラスタ数＝3と設定した場合



- クラスタ数＝4と設定した場合



- 備考
  - どう評価する？

議論しましょう

例　make_blobsによる練習

備考

- 分類としての評価（左）は低性能であった。この見方はクラス番号の与える順番が異なるため（右）。

```
print('accuracy =%f ' % accuracy_score(y_train, y_train_est))
print(classification_report(y_train, y_train_est))
```

```
accuracy =0.002000
             precision    recall  f1-score   support

          0       0.00      0.00      0.00       130
          1       0.01      0.01      0.01       119
          2       0.00      0.00      0.00       125
          3       0.00      0.00      0.00       126

avg / total       0.00      0.00      0.00       500
```

```
y_train[0:40]
```

```
array([0, 3, 1, 3, 2, 1, 0, 2, 2, 0, 0, 0, 0, 3, 3, 0, 3, 3, 0, 3, 3, 0, 0,
       3, 3, 1, 0, 2, 1, 1, 2, 0, 0, 1, 2, 2, 2, 0, 2, 3])
```

```
y_train_est[0:40]
```

```
array([2, 0, 3, 0, 3, 3, 2, 1, 1, 2, 2, 2, 2, 0, 0, 2, 0, 0, 2, 0, 0, 2, 2,
       0, 0, 3, 2, 3, 3, 3, 1, 2, 2, 3, 1, 1, 1, 2, 1, 0])
```

- 試しに，y_train_estを補正した評価は次である。

```
print('accuracy =%f ' % accuracy_score(y_train, b))
print(classification_report(y_train, b))
```

```
accuracy =0.496000
             precision    recall  f1-score   support

          0       1.00      1.00      1.00       130
          1       0.94      0.99      0.97       119
          2       0.00      0.00      0.00       125
          3       0.00      0.00      0.00       126

avg / total       0.48      0.50      0.49       500
```

例：Wholesale customers Data

- **ポルトガルの卸売業者の顧客データ（2011年(通年)、通貨単位の年間支出）**
  - UCI MLR, https://archive.ics.uci.edu/ml/datasets/wholesale+customers
  - 440件の顧客データ，各行が顧客1件を指す。
  - 上記のデータファイル名の空白を下線に変えて，次にアップした。
    - df = pd.read_csv("https://sites.google.com/site/datasciencehiro/datasets/Wholesale_customers_data.csv")

kMeans_Wholesale

Channel	販路，1： Horeca （Hotel/Restaurant/Caféの略称），2： 小売
Region	顧客の地域，1：リスボン市，2：ポルト市，3：その他
Fresh	生鮮品の年間支出（通貨単位，m.u. = monetary unit）
Milk	牛乳の年間支出（通貨単位）
Grocery	食料雑貨の年間支出（通貨単位）
Frozen	冷凍食品の年間支出（通貨単位）
Detergents_Paper	洗剤と紙類の年間支出（通貨単位）
Delicassen	惣菜の年間支出（通貨単位）

例：Wholesale customers Data

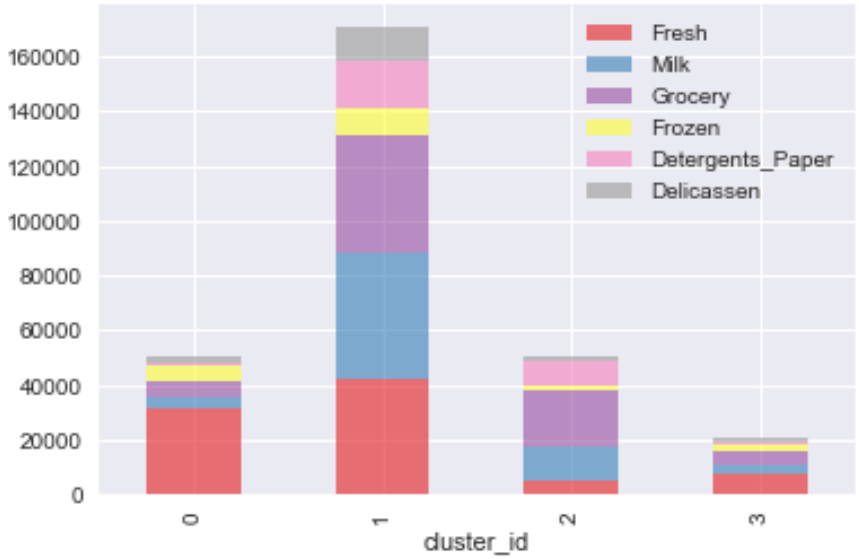
結果の考察

```
clstr = KMeans(n_clusters=4).fit_predict(cstmr_data)
```

クラスタ数4とした

```
df2.plot.bar( alpha=0.6, figsize=(6,4), stacked=True, cmap='Set1')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x19641984080>
```



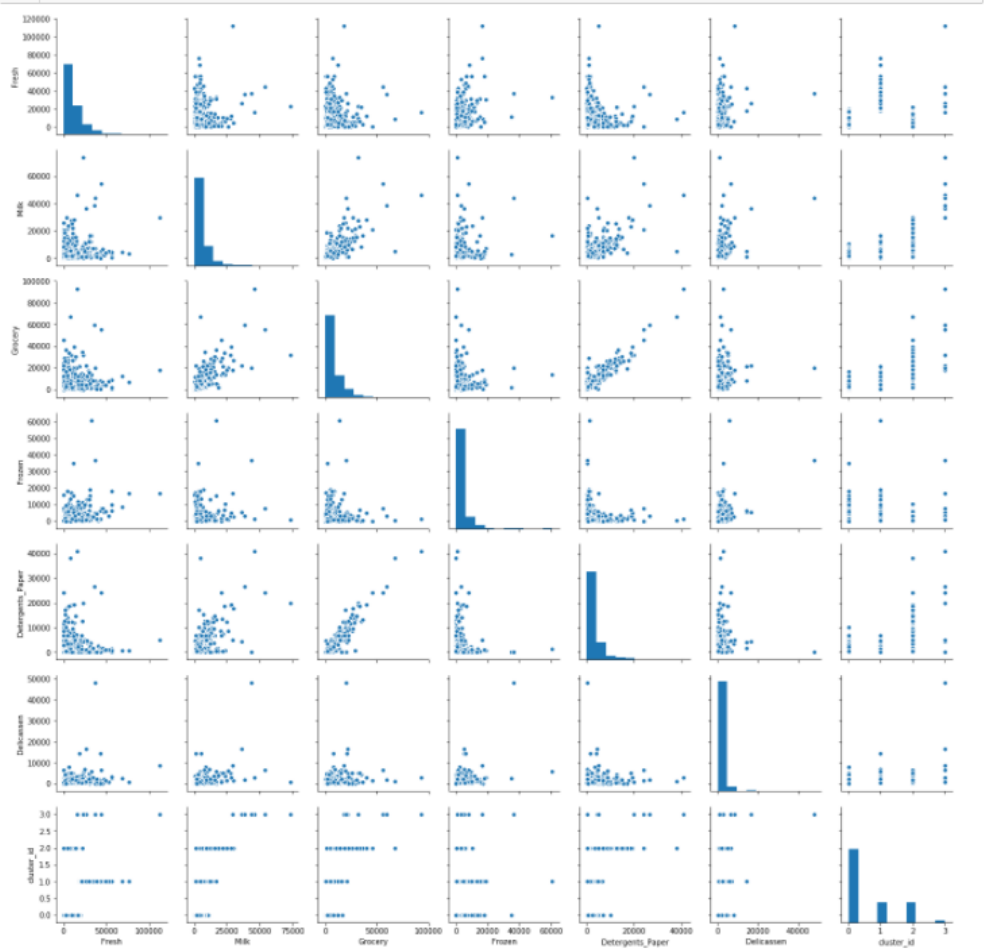
議論しましょう

```
結果の考察

クラスタ番号 = 0 顧客 ( 79件) 、Fresh (生鮮品) やFrozen (冷凍食品) の支出額が比較的高い
クラスタ番号 = 1 顧客 (  7件) 、全てのジャンルで支出額が高い
クラスタ番号 = 2 顧客 ( 77件) 、Milk, Grocery, Detergents_Paperの支出額が比較的高い
クラスタ番号 = 3 顧客 (280件) 、全体的に支出額が低い傾向
```

例：Wholesale customers Data

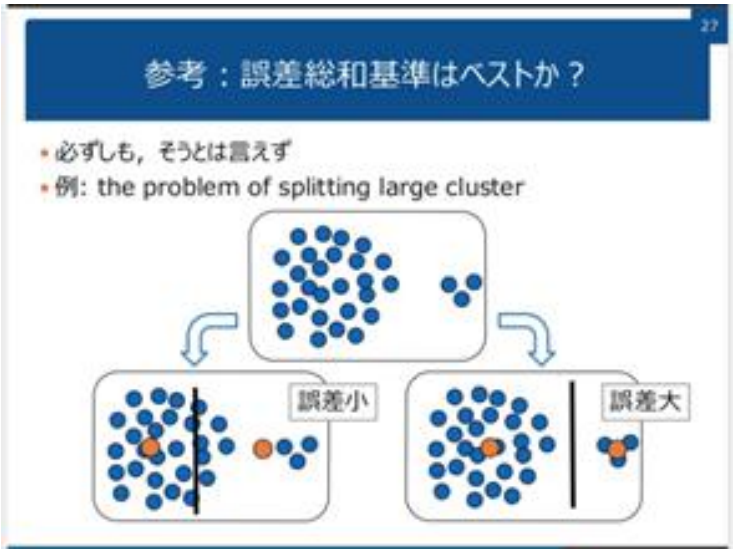
- ヒートマップを見て，２変数を適当に取り上げてクラスタリングを行った。
- この次に，主成分分析を行い，次元縮約を行ってから，クラスタリングを行い，考察を行うことを試みることも，良く行われている（各自で）。

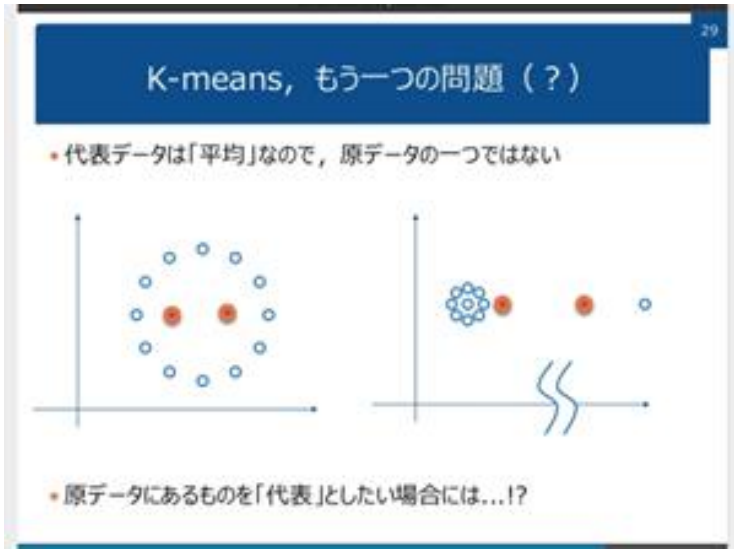


議論しましょう

参考

問題点





- 誤差総和基準はベストか？
- 代表データは原データとならない可能性が高い

- クラスタは超球状（２次元ならば円状）を仮定している。よって，超球に入り込んだ多種データをの抽出は困難である。
  - ✓ S.Guha, R.Rastogi, and K.Shim: CURE: An Efficient Clustering Algorithm for Large Databases, in Proc. of the ACM SIGMOD International Conference on Management of Data, pp.73-80 (1998)

引用:
データサイエンス概論第一=2-2 クラスタリング
https://www.slideshare.net/SeiichiUchida/22-77834256
神嶌：クラスタリング（クラスター分析）:http://www.kamishima.net/jp/clustering/