第8回　教師なし学習

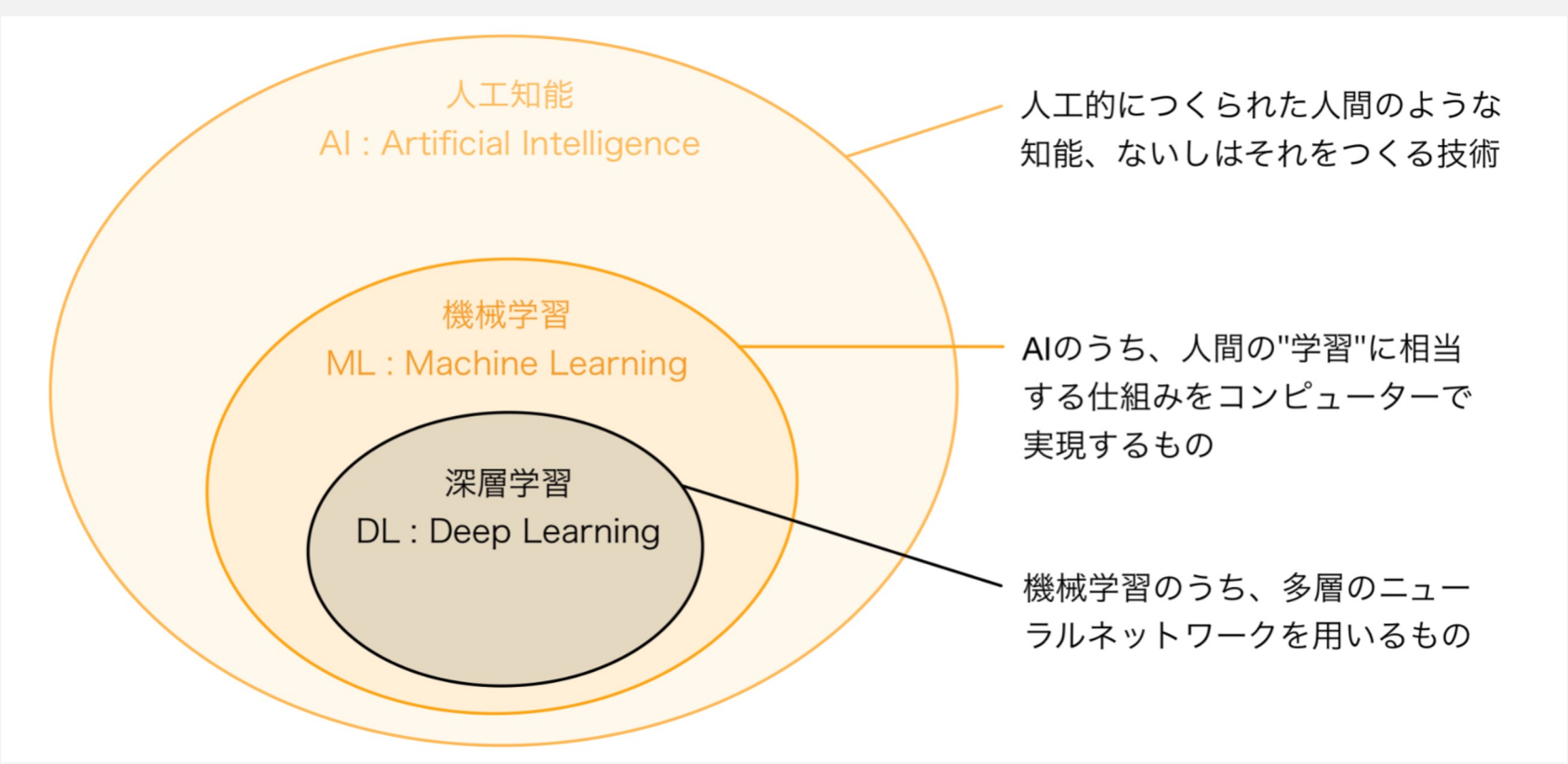
松尾・岩澤研究室
Matsuo-Iwasawa Lab UTokyo

講師・スライド作成：木島 悠輔

2024/11/26

目次

人工知能 (AI)・機械学習 (ML)・深層学習 (DL)



機械学習 (ML)

機械学習とは

＊人工知能のプログラム自身が学習する仕組み
・経験からの学習により自動で改善するコンピュータアルゴリズムもしくはその研究領域をさす

機械学習の代表的なタスク

① 教師あり学習 (Supervised Learning) - 正解ラベル付きの学習用データで学習

・与えられたデータ(入力)を元に、そのデータがどんなパターン(出力)になるのかを識別・予測する

e.g. 過去の売上から、将来の売上を予測したい
与えられた動物の画像が、何の動物かを識別したい

② 教師なし学習 (Unsupervised Learning) - 正解ラベルのない学習用データで学習

・(入力)データそのものが持つ構造・特徴を探索する

e.g. ECサイトの売上データから、どういった顧客層があるのかを認識したい
データの各項目間にある関係性を把握したい

③ 強化学習 (Reinforcement Learning) - 明確なデータからではなく、状態・行動・報酬から学習

・ある環境下で、目的とする報酬(スコア)を最大化するための行動を学習する

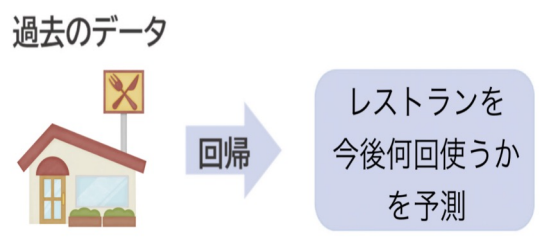
e.g. 将棋などのゲーム攻略、自動運転などのロボット制御

機械学習の分類

教師あり学習

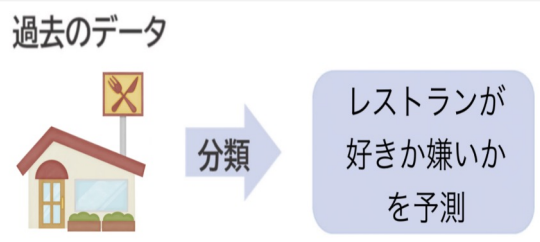
・回帰 - 数値(連続値)予測

連続する値の傾向を元に予測を行う。



・分類 - カテゴリ(離散値)予測

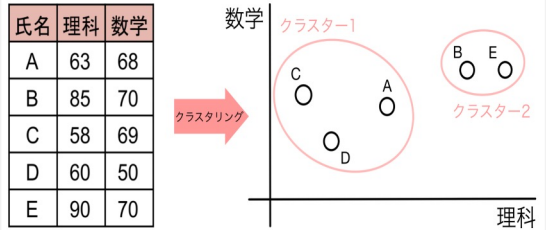
分析したいデータが所属するクラス分けを予測する。



教師なし学習

・クラスタリング - グループ構造探索

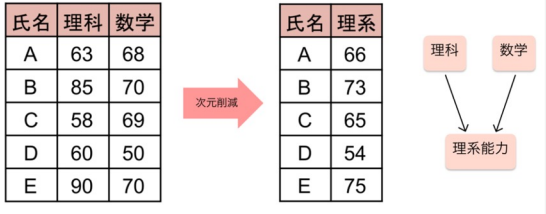
特徴が近いデータを集団に分ける。



氏名	理科	数学
A	63	68
B	85	70
C	58	69
D	60	50
E	90	70

・主成分分析 - データの次元削減

複数の説明変数をより少ない数の説明変数に変換し、情報を要約する。



氏名	理科	数学
A	63	68
B	85	70
C	58	69
D	60	50
E	90	70

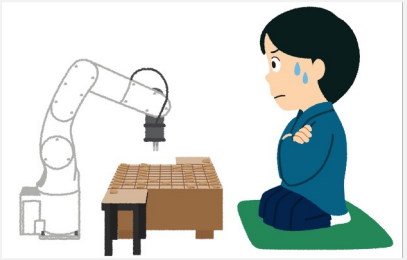
氏名	理系
A	66
B	73
C	65
D	54
E	75

強化学習

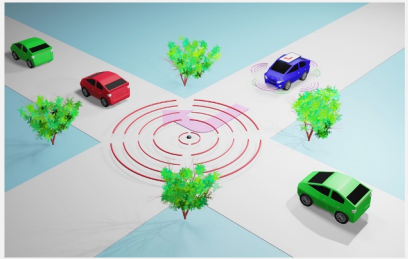
・最適化 - 最適なシステム制御の実現

ある環境下での最適な行動を見つける。

・ゲーム攻略



・ロボット制御

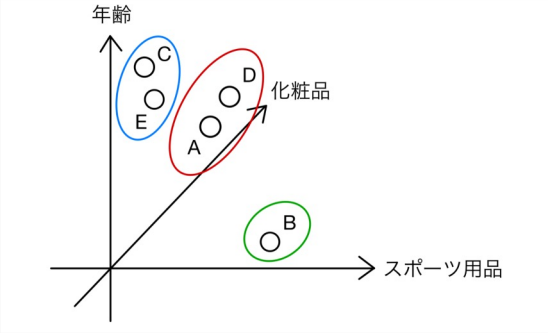


クラスタリング

実務でのクラスタリング活用例　基本編

顧客セグメンテーション

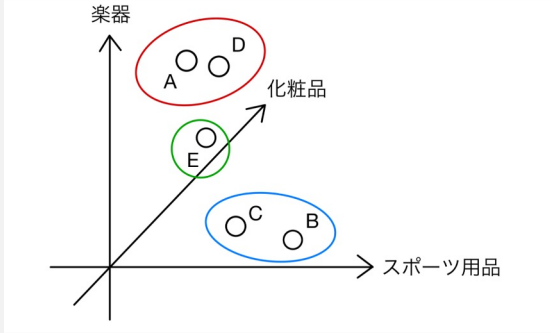
	年齢	性別	化粧品	スポーツ用品
Aさん	18	女性	20	1
Bさん	21	男性	2	8
Cさん	50	女性	12	0
Dさん	24	女性	25	2
Eさん	48	女性	14	1



マーケティングにおいて、顧客の性別や年齢、趣味嗜好などをもとに顧客市場を分類していくことは非常に重要。最適な顧客層に対して商品の訴求を行うことが可能。

商品店舗分類

	化粧品	スポーツ用品	楽器	ゲーム
店舗A	60	10	90	20
店舗B	10	120	10	30
店舗C	20	100	20	50
店舗D	100	30	80	10
店舗E	50	40	30	100



同じ商品が購買されやすい店舗をグループ化し、在庫管理や併売分析を行う。同じグループに属する店舗に共通の地域性などを見つけ出すことが可能。

教師あり/なし学習(クラスタリング)のコードの違い

教師あり学習

```
df = pd.read_csv('gci.csv')

x = df[['column1', 'column2', 'column3']]
y = df['target']

x_train, x_test, y_train, y_test = train_test_split \
(x, y, test_size = 0.2, random_state = 0)

model = LinearRegression()
model.fit(x_train, y_train)
model.score(x_test, y_test)
```

- データを説明変数xと目的変数yに分割
- 説明変数と目的変数を訓練データとテストデータに分割
- fitメソッドには説明変数と目的変数をセットで渡す

教師なし学習(クラスタリング)

```
df = pd.read_csv('gci.csv')

x = df[['column1', 'column2', 'column3']]

model = KMeans(n_clusters=6, random_state=0)
model.fit(x)
model.labels_
df['cluster'] = model.labels_

cluster_mean = df.groupby('cluster').mean()
cluster_mean.plot(kind = 'bar')
```

- データから説明変数のみを選ぶ(目的変数は不要)
- 予測はしないのでデータをsplitで分割しない
- fitメソッドには説明変数のみ渡す

k-means (k平均法)　＊クラスタリングの代表例＊

互いに近いデータ同士は同じクラスタであるという考えに基づく、データ群をk個に分類する手法
(クラスタ数を自動推定してくれる、X-meansやG-meansといったアルゴリズムもある)

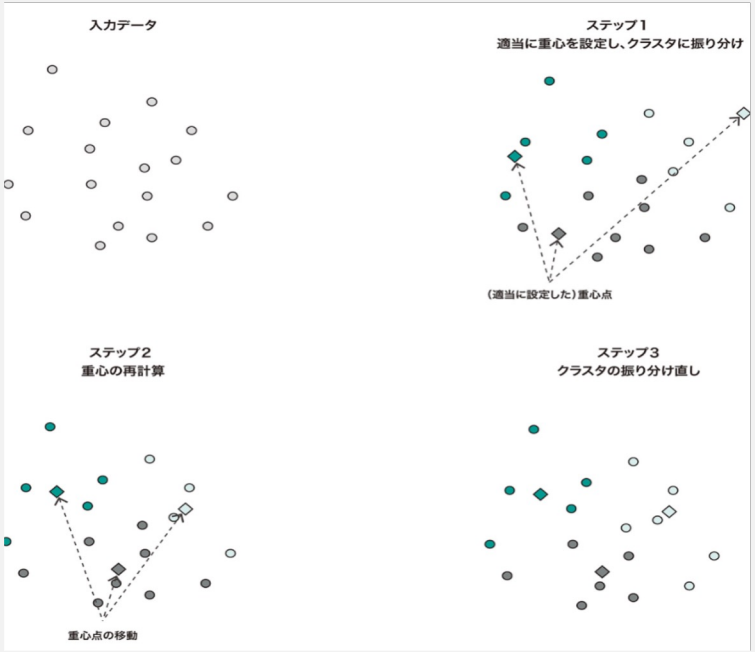
0. 入力データをプロットする。
(アルゴリズムの説明のため可視化)

1. まずは適当に各データをk個のクラスタに振り分ける。

2. 各クラスタの重心を求める。

3. 求まったk個の重心と各データとの距離を求め、各データを
最も距離が近い重心に対応するクラスタに振り分け直す。

4. 重心の位置が(ほとんど)変化しなくなるまで2,3を繰り返す。

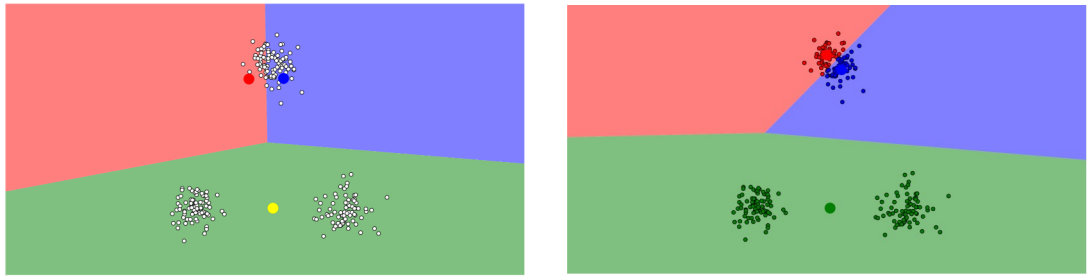


図は参考文献01より引用

k-meansの注意点とk-means++による解決策

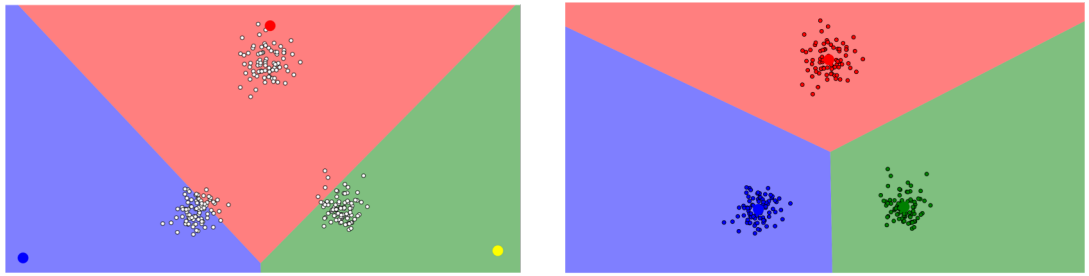
① k-meansの注意点 (初期の重心点の位置の偏り)

・初期のランダムに配置した重心点の位置により、最終的なクラスタリングの結果は変わりうる。初期の重心点が互いに近い距離に配置されると、直感に反したクラスタリングとなることがある。



② k-means++による解決策 (初期値の選択の改良)

・k-means++は、データ点を考慮した上で、初期の重心点を互いになるべく遠ざけて配置し、直感に反したクラスタリングとなることを防ぐ、k-meansの上位互換モデル。**実際にクラスタリングを行う際は、暗黙のうちに、k-meansを使うのではなくk-means++を使う**ことが多い。

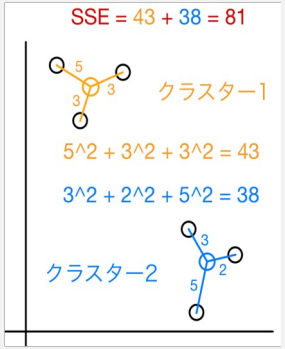


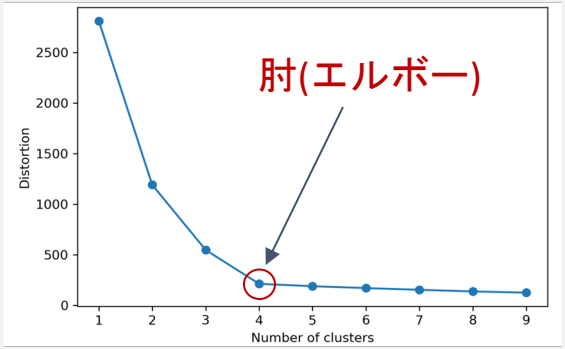
図は参考文献02より作成

k-meansの最適なクラスター数を調べる手法

「エルボー法」と「シルエット分析」

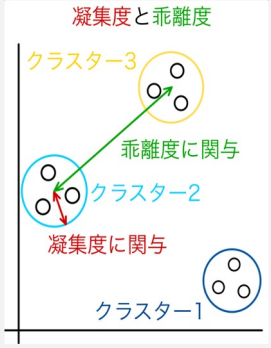
エルボー法

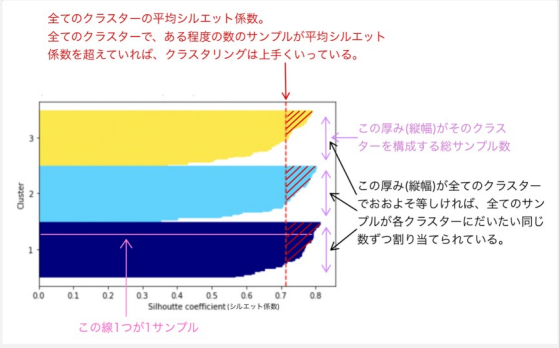




クラスターごとの各データと重心点との距離の2乗の総和を、全クラスタ合計した値であるクラスター内誤差平方和(SSE)を計算しプロットする手法。SSE値が小さいほど歪みが小さいモデルと言える。**SSE値が肘(エルボー)のようにガクッと曲がる点を最適なクラスター数**とみなす。

シルエット分析



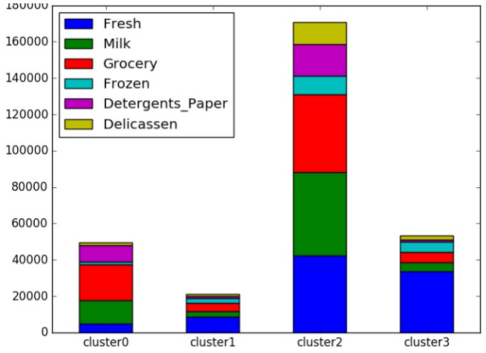


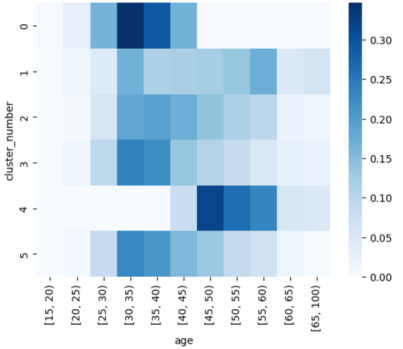
「クラスター内の各データの凝集度」と「別クラスターからの乖離度」から**シルエット係数**を計算し、クラスターごとにソートし棒グラフにする手法。シルエット係数は[-1,1]の区間に収まり、**1に近いほど性能が高く、負であればクラスターへの割り当てが間違っている可能性**がある。

クラスタリング結果の解釈と注意点

① クラスタリング結果の解釈

・**groupby メソッドでクラスタごとに集計する手法が代表的。**各特徴量の平均値や割合を集計し、それらを可視化する方法がよく使われる。散布図や散布図行列、棒グラフ、ヒートマップなどがよく使われる。





② クラスタリング結果の解釈における注意点

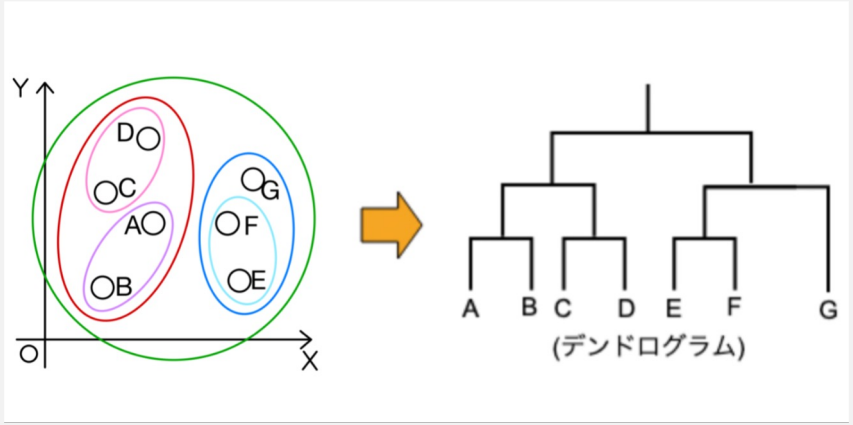
・**クラスタリングは結果を解釈する過程で必ず主観がはいる。**非階層クラスタリングでは、「何個のクラスターに分けるか」「あるクラスターに要素Aと要素Bが多いということはXということである」などの主観がはいる。階層クラスタリングでは、「デンドログラムのどこを閾値とするか」などの主観がはいる。

・**必ずしも解釈可能な結果になるとは限らない。**エルボー法やシルエット分析を行っても美しい結果が得られないことも多々ある。直感に反した分類となったり、綺麗に分類できても分類の意味を人が解釈できないこともある。**仮説やドメイン知識に基づく特徴量選択など、試行錯誤を前提に分析することが重要**である。

クラスタリングの種類①

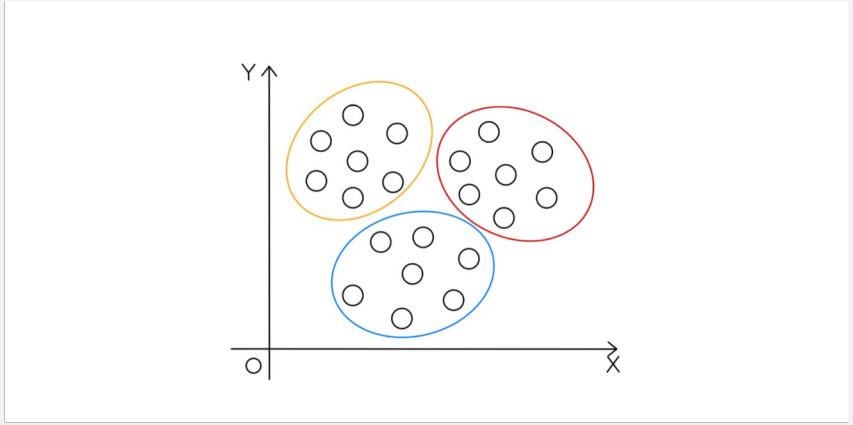
「階層的クラスタリング」と「非階層的クラスタリング」

階層的クラスタリング



最も似ている2つのクラスターを順に結合していき、その過程を**デンドログラム**などで視覚化することで、データの特徴を把握することができる手法。**ウォード法**が有名。

非階層的クラスタリング

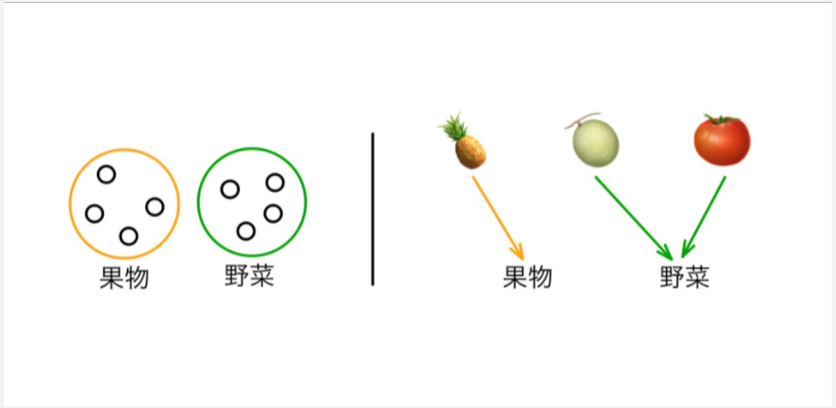


階層的な構造を持たず、事前にいくつのクラスターに分けるかを決め、サンプルを決めたクラスター数に分割していく手法。**k-means**が有名。

クラスタリングの種類②

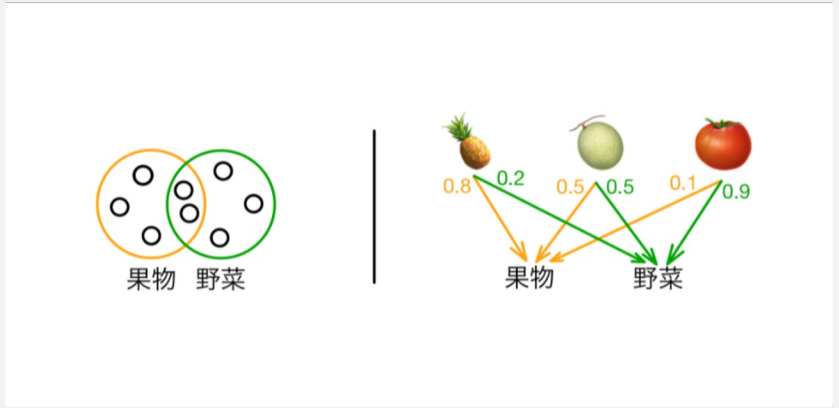
「ハードクラスタリング」と「ソフトクラスタリング」

ハードクラスタリング



各データを1つのクラスターのみに所属するように割り当てる手法。一般的にクラスタリングといえばハードクラスタリングをさす。**k-means**が有名。

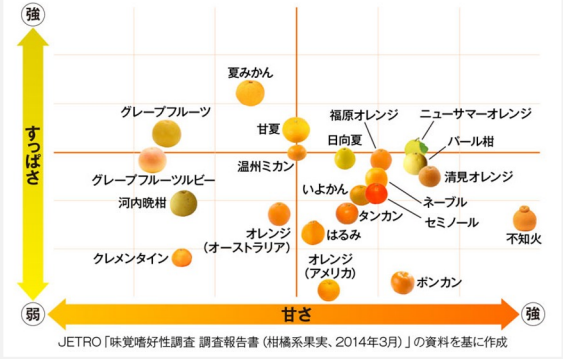
ソフトクラスタリング



各データが複数のクラスターに所属することを許して割り当てる手法。各データが複数のクラスターに所属する確率を計算する。**混合分布モデル**が有名。

実務でのクラスタリング活用例　発展編

自然言語処理

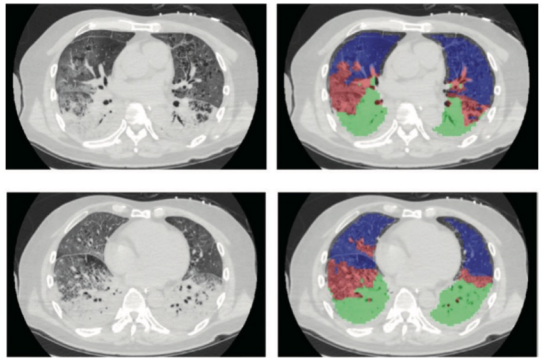


出典:色も形も味もさまざま！“みかんチャート”

https://weathernews.jp/s/topics/202001/140125/

SNSへの投稿やアンケートから同じような反応をしている顧客をグループ化してアプローチしたり、会社の口コミで同じような言葉が使われているものをグループ化し、似たような会社を調べたりすることができる。

画像処理

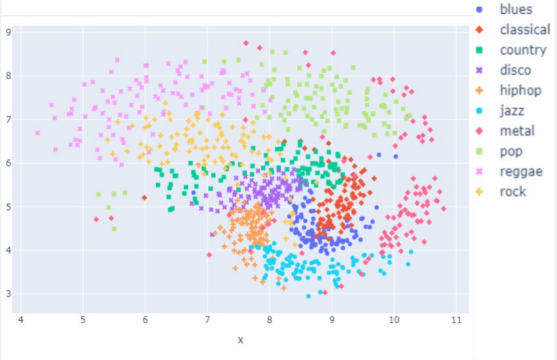


出典:COVID-19診断支援AI開発における名古屋大学の取り組み

https://www.jstage.jst.go.jp/article/mit/39/1/39_13/_article/-char/ja

画像のグルーピングで同じような画像を集めたり、画素により写真中の領域をグループ化し陰影に対応させたり、画像内の似たような色をグループ化し単一の色に置き換えデータ容量を削減したりすることが可能。

音声処理



出典:音楽をメルスペクトログラム+UMAP + G-meansでクラスタリングしてみた

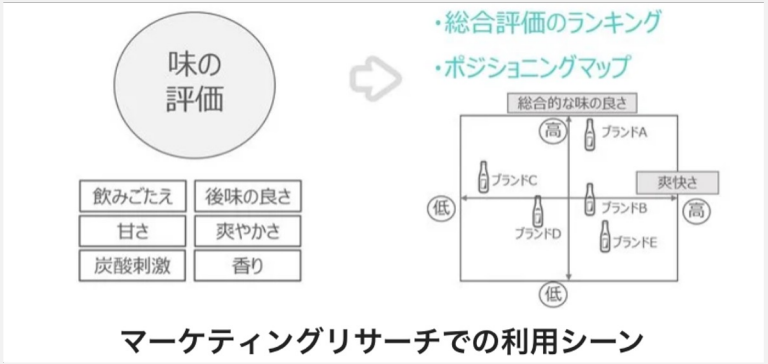
https://zenn.dev/migawari1253/articles/e6f7df001a355c

様々なジャンルの膨大な数の音楽を「カフェに最適な音楽」「ダンス向けクラブミュージック」「寝る時に聴く音楽」「ラップミュージック」のように分けることが可能。同じジャンルのプレイリストを作成できる。

主成分分析

実務での主成分分析活用例

マーケティングリサーチ



出典:主成分分析とは？事例を用いて結果の見方や注意点をわかりやすく解説

https://surveroid.jp/mr-journal/data_analysis_method/aqQEF

それぞれの評価をランキングにすることもできるが、たくさんの項目があるので煩雑になる。そこで主成分分析を使ってまとめてしまい、味やブランドの評価、企業の評価を少ない変数で要約することで、総合評価ランキングやポジショニングマップを作成することができる。

データサイエンス

出典:主成分分析とは？事例を用いて結果の見方や注意点をわかりやすく解説

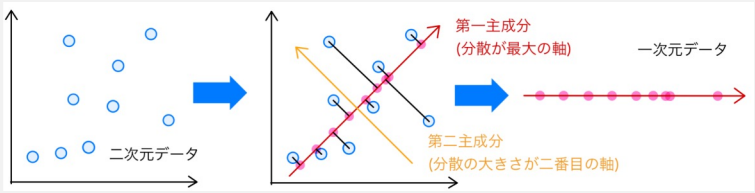
https://surveroid.jp/mr-journal/data_analysis_method/aqQEF

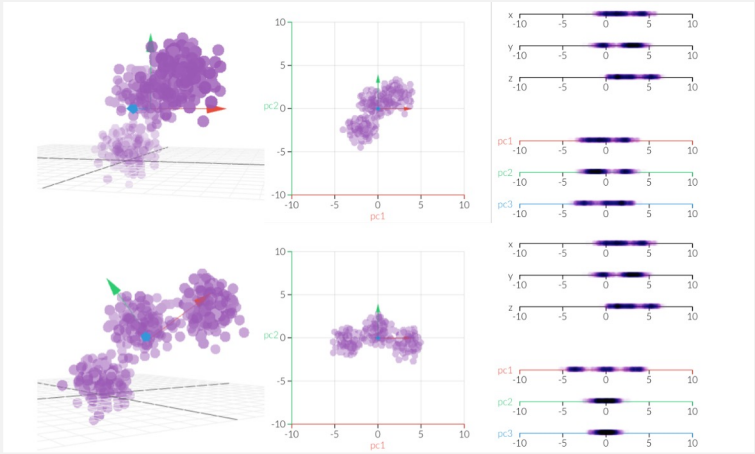
たくさんのデータがある時、膨大なデータ量やカラムをそのまま使うと計算コストがかかったり、予測モデルを作る際に過学習に陥ったりする。そこで主成分分析で次元削減し、特徴量を要約することで、それらの問題を解決できることがある。

主成分分析 (PCA)　＊次元削減の代表例＊

データの特徴量間の相関を分析することでデータの構造をつかみ、データの特徴量を要約する手法
(次元削減の手法として、PCAの他にSVDやt-SNE、UMAPといったアルゴリズムもある)

0. 入力データをプロットする。
(アルゴリズムの説明のため可視化)
1. 各変数間の共分散を計算し、共分散行列を作成する。
2. 共分散行列の固有値・固有ベクトルを求める。
3. 最大の固有値を持つ固有ベクトルが第一主成分となる。
4. 元のデータを固有ベクトルに投影し、新しいデータを作成する。





図は参考文献03より引用

教師あり/なし学習(主成分分析)のコードの違い

教師あり学習

```
df = pd.read_csv('gci.csv')

x = df[['column1', 'column2', 'column3']]
y = df['target']

x_train, x_test, y_train, y_test = train_test_split \
(x, y, test_size = 0.2, random_state = 0)

model = LinearRegression()
model.fit(x_train, y_train)
model.score(x_test, y_test)
```

- データを説明変数xと目的変数yに分割
- 説明変数と目的変数を訓練データとテストデータに分割
- fitメソッドには説明変数と目的変数をセットで渡す

教師なし学習(主成分分析)

```
df = pd.read_csv('gci.csv')

x = df[['column1', 'column2', 'column3', 'column4', 'column5']]

model = PCA(n_components=2, whiten=True)
model.fit(x)

print(model.components_)
print(model.explained_variance_)
print(model.explained_variance_ratio_)
```

- データから説明変数のみを選ぶ(目的変数は不要)
- 予測はしないのでデータをsplitで分割しない
- fitメソッドには説明変数のみ渡す

主成分分析の学習結果の確認

① 固有ベクトル(components_属性)

・print(model.components_)で出力され、主成分分析によって得られた新規の軸の向きを表す。軸を表すものなので、＋−の符号が反転していても問題ない。固有ベクトルは長さが1の単位ベクトルであり、互いに全て直交している。

② 固有値(explained_variance_属性)

・print(model.explained_variance_)で出力され、各主成分の分散を表す。元データの分散値の総和と、各主成分の分散値の総和は一致する。1つの基準として、固有値が1より大きい主成分を採用する(Kaiser基準)というものがある。

③ 寄与率(explained_variance_ratio_属性)

・print(model.explained_variance_ratio_)で出力され、各主成分がどの程度元データの情報を反映しているかを表す。固有値を用い、「各主成分の分散値」が「分散値の総和」の何%を占めているかという寄与率が分かる。

主成分分析の注意点

① 解釈の難しさ

・主成分分析を用いて得られる結果は、統計的な指標や数値情報である。しかし、見つかった主成分が具体的にどのような意味を持つのかは、分析者の解釈に委ねらる。分析対象とする変数の設定、第一主成分における定義、第二主成分以降の意味づけなどは分析者が判断する。

② 情報の取りこぼし

・主成分分析は、データを要約する手法であるので、どうしても取りこぼされる情報が発生する。主成分の数を増やせば増やすほど網羅的に情報を残せるが、次元削減の目的が失われ、そこまでのメリットは無いので多くの場合データは捨てられる。そこで捨てられたデータに重要な分析すべき情報が含まれている可能性も否定はできない。

③ 外れ値の影響

・外れ値は通常のデータパターンから大きく逸脱した値であり、分析結果に悪影響を及ぼす可能性がある。主成分分析はデータの分散を最大化する方向を求める手法である。そのため、外れ値が分散に大きく影響すると、主成分の方向や寄与率が歪められてしまう。これにより、分析結果が歪んだり、軸の解釈が困難になってしまう問題が発生する。

ML用語集

教師あり/なし学習まわりの用語集

＊教師あり学習
- 回帰問題
- 分類問題
- 半教師あり学習
- ブートストラップサンプリング
- 単純パーセプトロン
- 多層パーセプトロン
- 擬似相関
- 線形回帰
- 単回帰分析
- 重回帰分析
- ラッソ回帰
- リッジ回帰
- ロジスティック回帰
- 自己回帰モデル(AR)
- ベクトル自己回帰モデル(VAR)
- 決定木
- 剪定
- アンサンブル学習
- バギング
- ランダムフォレスト
- (勾配)ブースティング
- XGBoost
- LightGBM
- CatBoost
- AdaBoost
- スタッキング
- サポートベクターマシン(SVM)
- マージン最大化
- カーネルトリック
- シグモイド関数
- 隠れ層
- 活性化関数
- ソフトマックス関数
- 誤差逆伝播法
- k近傍法(k=NN)

＊教師なし学習
- クラスタリング
- 次元削減
- 階層的クラスタリング
- 非階層的クラスタリング
- ハードクラスタリング
- ソフトクラスタリング
- k-means(k平均法)
- ウォード法
- デンドログラム(樹形図)
- トピックモデル
- 潜在的ディリクレ配分法(LDA)
- 主成分分析(PCA)
- 特異値分解(SVD)
- t-SNE
- アソシエーション分析
- 協調フィルタリング
- コンテンツフィルタリング
- コールドスタート問題

参考文献

01 深層学習教科書 ディープラーニング G検定 公式テキスト 第2版

02 Visualizing K-Means Clustering

- https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

03 Principal Component Analysis

- https://setosa.io/ev/principal-component-analysis/

04 総務省統計局 機械学習 (教師あり学習)

- https://www.stat.go.jp/teacher/dl/pdf/c4learn/materials/fourth/dai3.pdf

05 総務省統計局 機械学習 (教師なし学習)

- https://www.stat.go.jp/teacher/dl/pdf/c4learn/materials/fourth/dai4.pdf

06 スッキリわかるPythonによる機械学習入門

07 Kaggleで勝つデータ分析の技術


松尾・岩澤研究室
Matsuo-Iwasawa Lab UTokyo