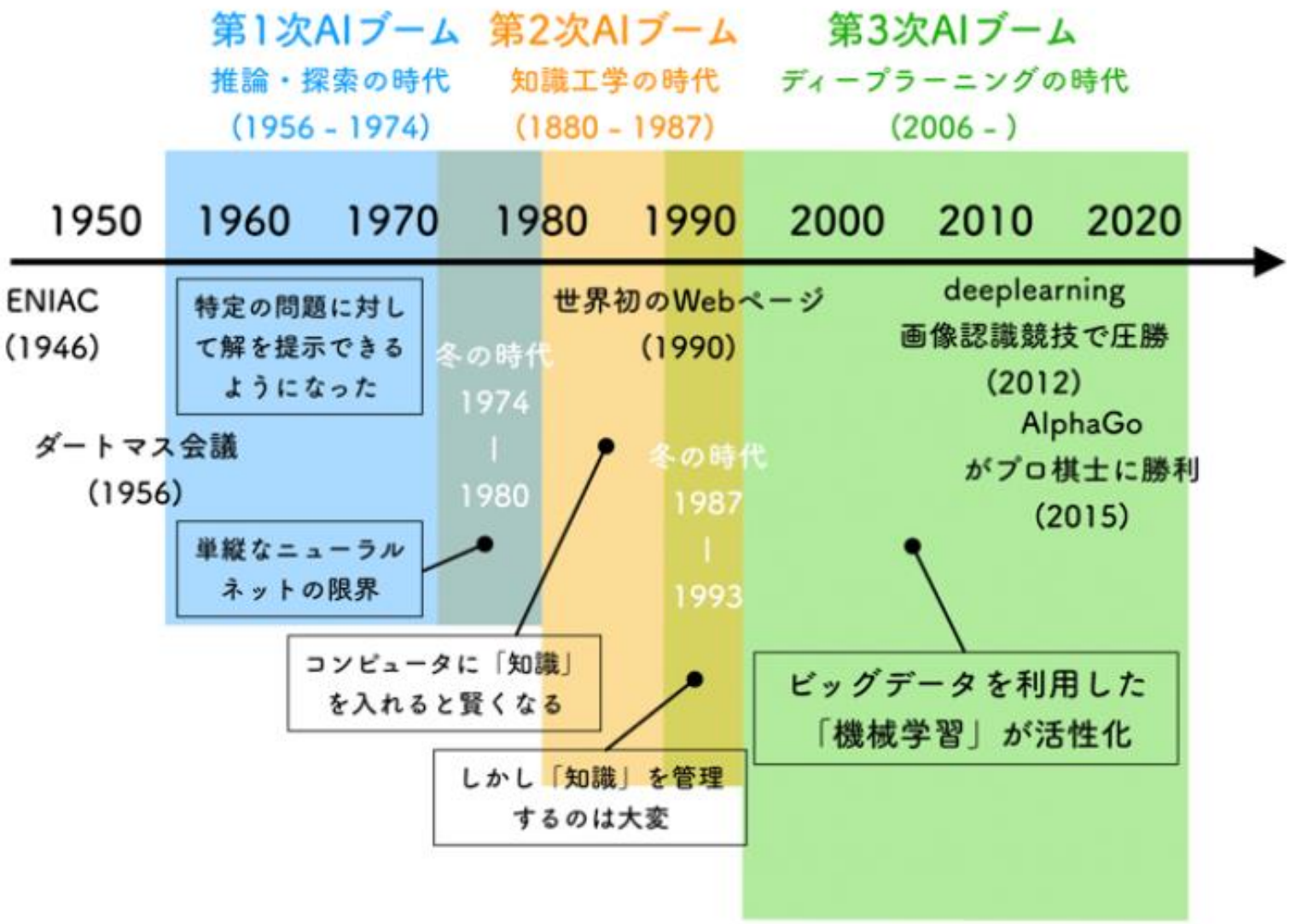
知識工学

第八回　機械学習

今日の内容

- 機械学習の種類や基礎的な手法について学ぶ
- ９０年代後半から００年代あたり



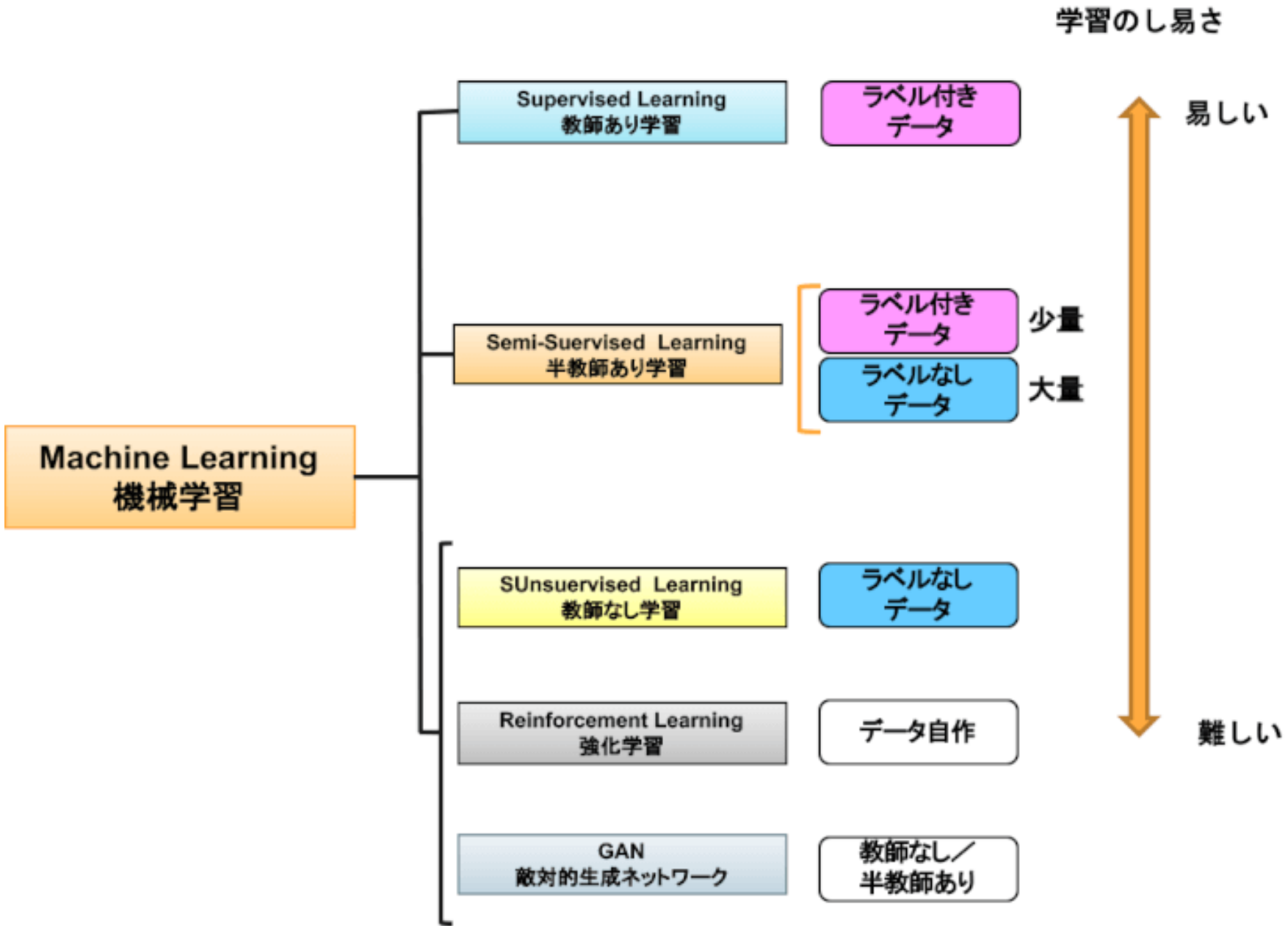
機械学習とは

- コンピュータプログラムがタスクのクラスTと性能指標Pに関し経験Eから学習するとは、T内のタスクのPで測った性能が経験Eにより改善される事を言う。（トム・M・ミッチェル）

わかりやすく言うと


“特徴”をつかみ
“法則化”する
データから反復学習し
学習結果を法則化(モデル化)
機械
学習
法則を
“自動化”する
ノンプログラミングで
システム化し再現性を作る

機械学習の種類


学習のし易さ
Machine Learning
機械学習
Supervised Learning
教師あり学習
ラベル付き
データ
Semi-Suervised Learning
半教師あり学習
ラベル付き
データ
少量
ラベルなし
データ
大量
SUnsuervised Learning
教師なし学習
ラベルなし
データ
Reinforcement Learning
強化学習
データ自作
GAN
敵対的生成ネットワーク
教師なし／
半教師あり
易しい
難しい

説明変数と目的変数

期末試験の点数を予想しようと思ったら，どのような変数が説明変数となる？

変数の種類

質的変数：数値で測れない性質．
カテゴリやラベル（定性的）

量的変数：数値で計測できるもの．量（定量的）

- 桃色，植物，ハナミズキ

- 花びら４枚，１輪，葉２枚

- 紫色，花，あやめ

- 花びら４枚，２輪，葉９枚

バイアス

- ある視点からの見方（偏見）を，エキスパートシステムにインプットすることで，効率的な学習ができる

生データ
26, 178, 65

年齢，身長，体重

歯の数，フォロー数，
フォロワー数

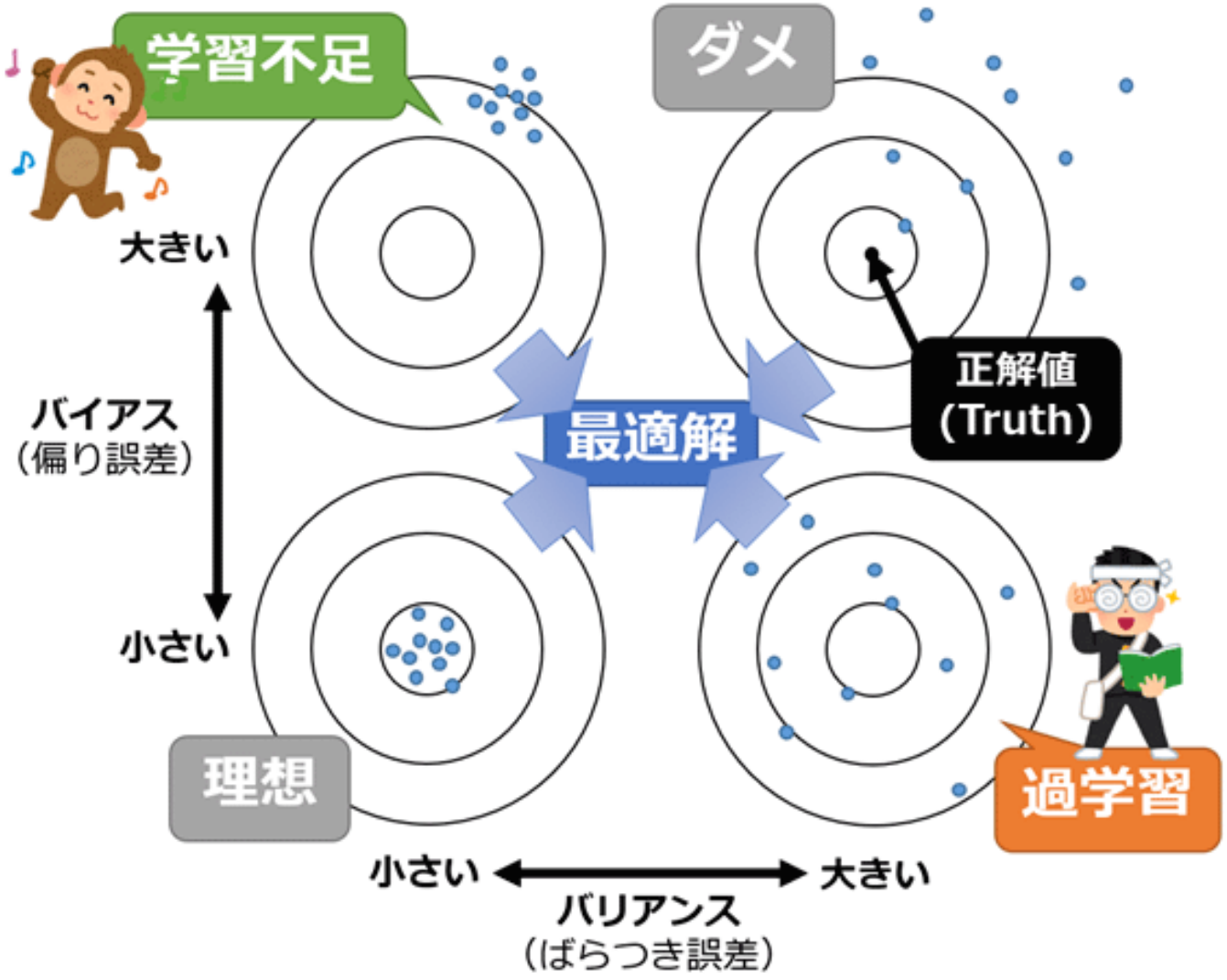
バイアスとバリアンス

バイアス：正解値からのズレ

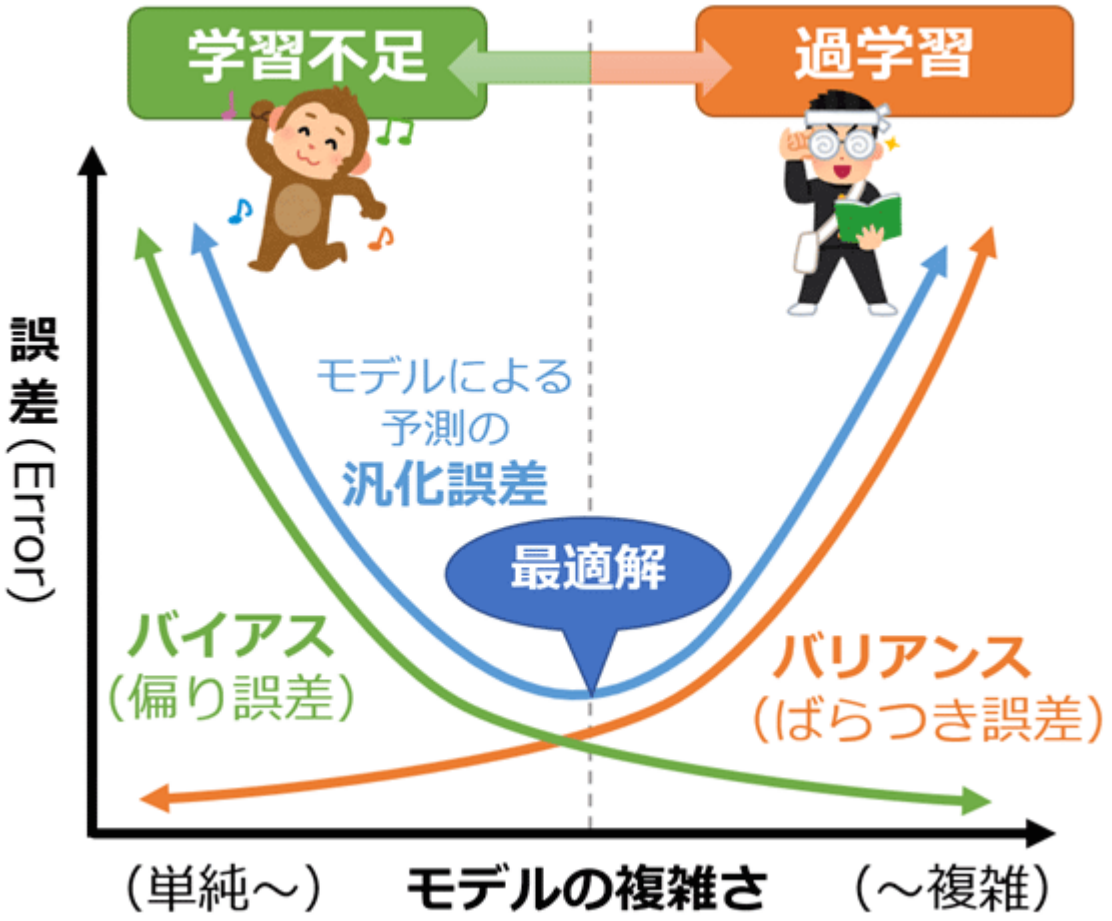
バリアンス：分散

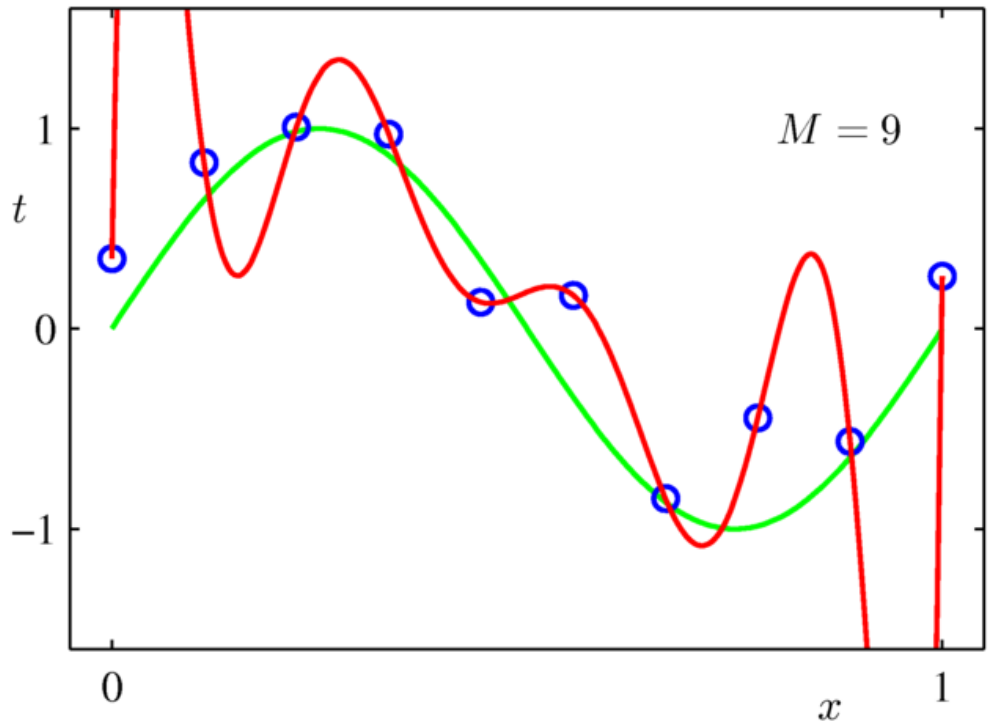
- 中心が正しい値
- 予測は，中心から離れず，ノイズによるばらつきが小さいこと（左下）が望ましい．

- この2つはトレード・オフの関係にある



トレードオフ


学習不足
過学習
誤差（Error）
モデルによる
予測の
汎化誤差
最適解
バイアス
（偏り誤差）
バリアンス
（ばらつき誤差）
（単純～）
モデルの複雑さ
（～複雑）


$M = 9$
1
0
−1
t
0
1
x

帰納学習（一般的な機械学習）

- 帰納学習とは，多くの事例から知識を獲得すること
- 「これまでの職場全てで，上司がうざかった」⇒ 一般的に上司はうざい

訓練データ

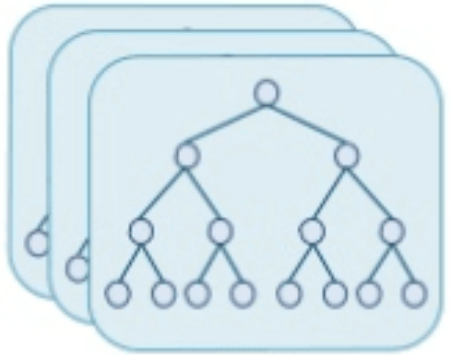
正データ

負データ

訓練

人という概念を学習
（モデル構築）

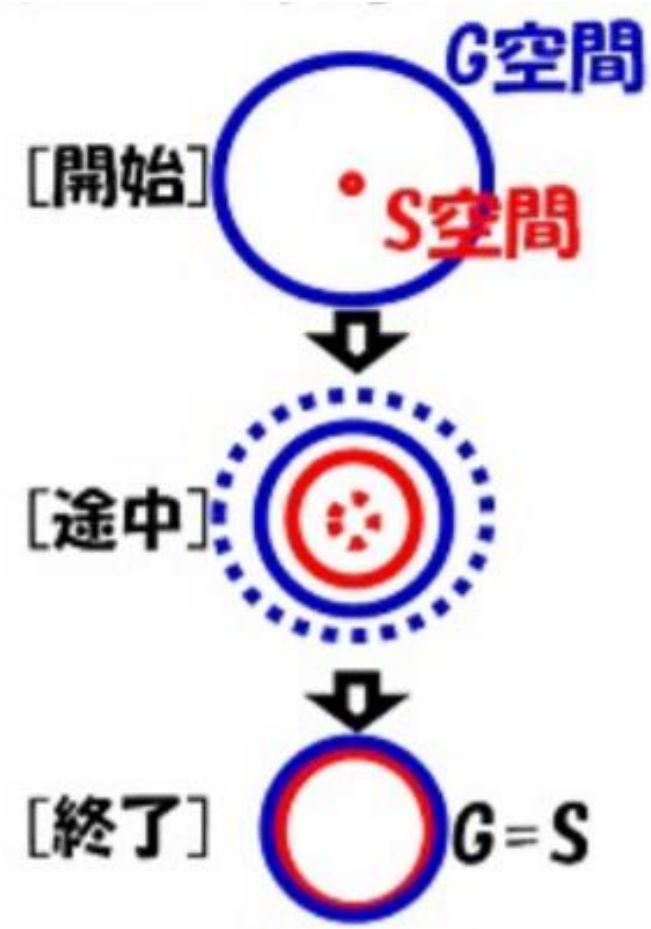
得られた知識



第一部の終わり

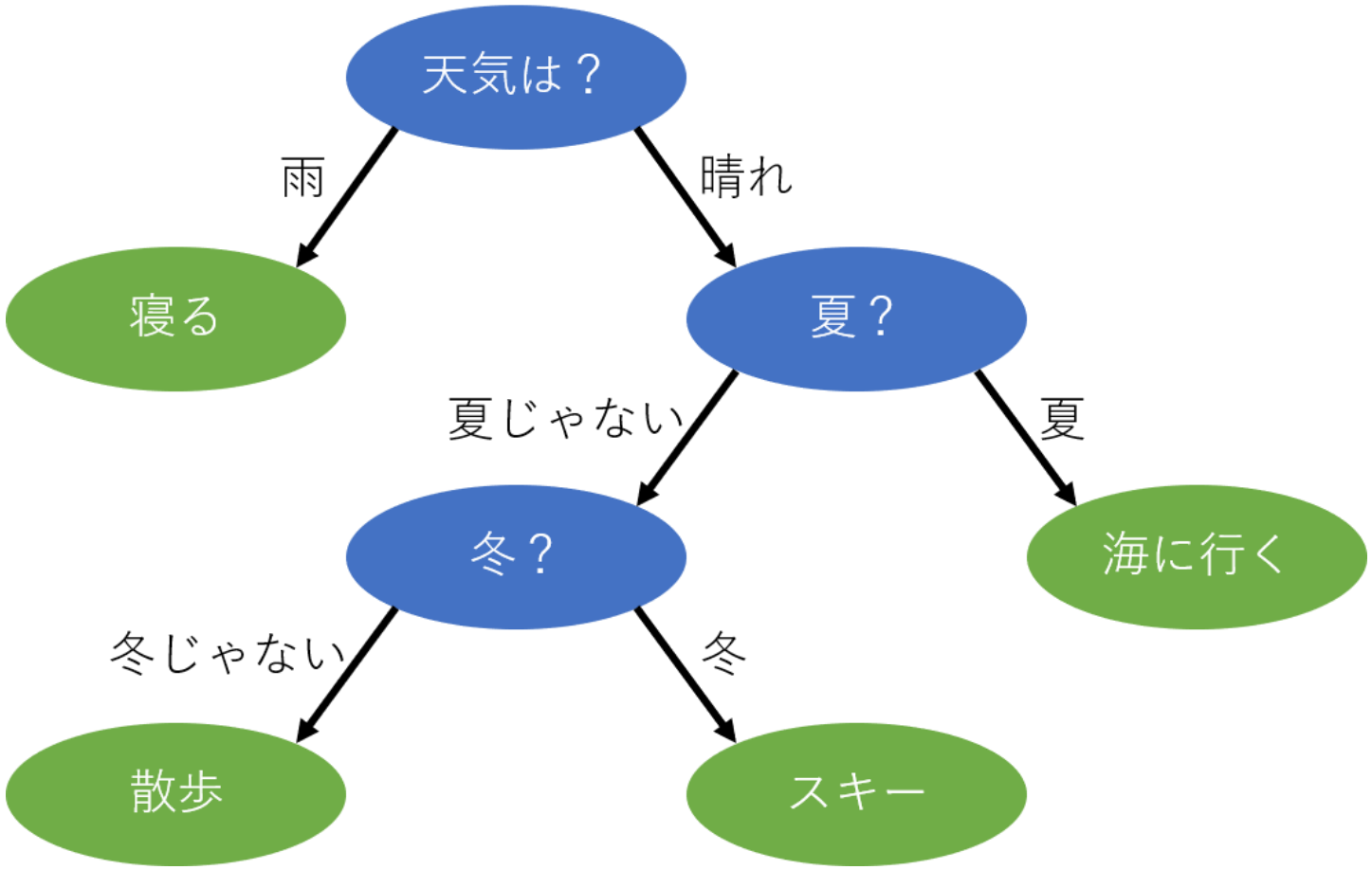
バージョン空間法

- 人間がYesかNoを判断するだけで一般化したルールを作れる
- 最も一般化された空間Gと，最も特殊化された空間Sをすり合わせて，一致したら終了.
- データが与えられるたびに，それにそぐわないルールが消えていく



決定木

- 教師あり学習
- ラベル
- 学習データから木構造を構築し，新規のデータが来た場合はそれに沿って予測する



決定木の特徴

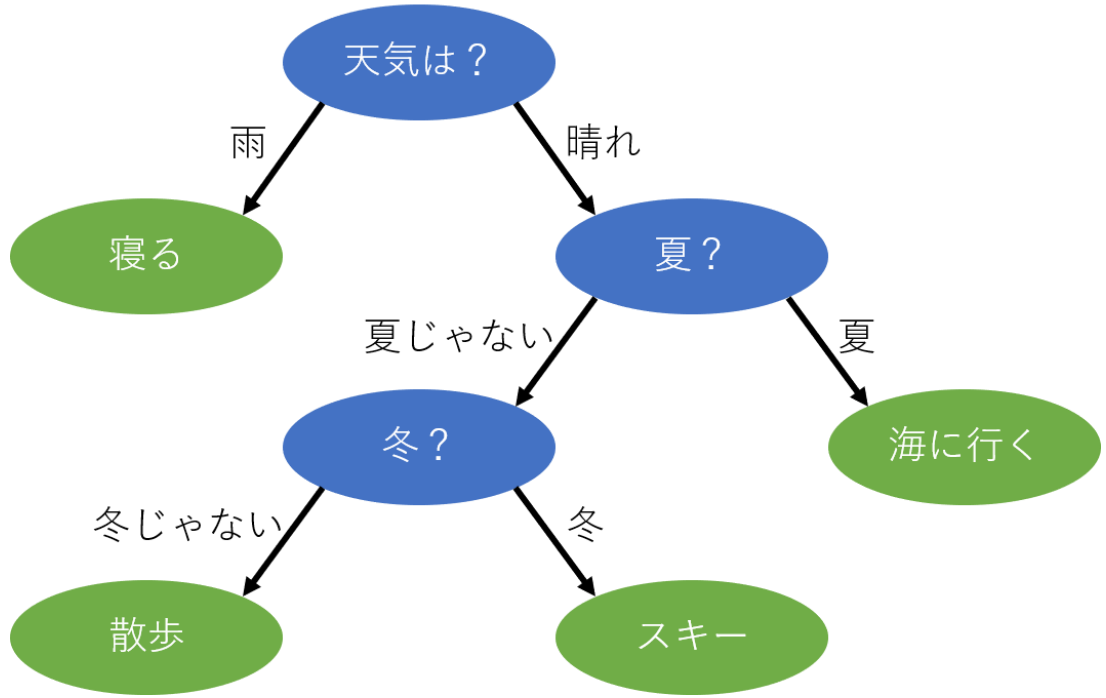
学習データ

天気	季節	行動
雨	夏	寝る
晴れ	夏	海へ
晴れ	冬	スキー
晴れ	秋	散歩
雨	春	散歩

説明変数　　目的変数

構築

数値ではなく，ラベルのデータに向いている
なぜその予測結果となったのかを視覚的に理解できる



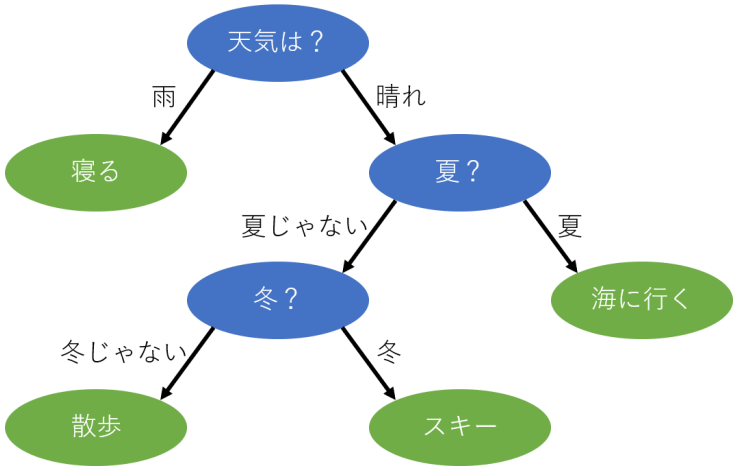
予測

未知のデータ

天気	季節	行動
雨	秋	？

決定木の構築

- 天気が先ではなく季節を先に持ってきたり，春や秋の分岐を取り入れても，分類自体は行える
- どのような木を構築するか
  - なるべく単純な木が優先される（オッカムの剃刀）
- 定量的にシンプルさを評価するために，**平均情報量（エントロピー）**という指標を用いる



シャノン情報量i

$$i(x) = -\log_2 P(x)$$

- iは情報量．P(x)は，その物事が起こる確率を示す．
- マイナスが付いているため，確率が小さいほどi(x)は大きくなる
- ２つの事柄の情報量は，和で示すことができる

- 珍しい出来事であるときのほうが，多くの情報量がある
  - 例）1～40の目があるルーレットで，1～20のどれかが出るという情報より，１～10のどれかが出るという情報のほうが珍しい．かつ偶数となったらさらに珍しい（情報量が多い）

平均情報量（エントロピー）

$$H[x] = -\sum_{x} p(x) \log p(x)$$

すべての項目に対する，得られる情報量の期待値

- ルーレット(1~40)で，１～10かつ偶数であると知っているとき，出た目の情報に対する平均情報量

$$\mathrm{H(x)} = -\frac{1}{5} \times \{\log_2 \frac{1}{5} + \log_2 \frac{1}{5} + \log_2 \frac{1}{5} + \log_2 \frac{1}{5} + \log_2 \frac{1}{5}\}$$
$$\fallingdotseq 2.32$$

- ルーレット(1~40)で，出た目について何も知らない時，出た目の情報に対する平均情報量

$$\mathrm{H(x)} = -\frac{1}{40} \times \{\log_2 \frac{1}{40} + \log_2 \frac{1}{40} + \ldots + \log_2 \frac{1}{40} + \log_2 \frac{1}{40}\}$$
$$\fallingdotseq 5.32$$

決定木の構築方法：ID3アルゴリズム

1. ルートノードNを作成し、全データを所属させる。
2. Nに所属するデータが、全て同じ分類先なら、処理を終了する。
3. Nに所属するデータの平均情報量を計算する。
4. 各変数を質問とした時の平均情報量を求める。
5. 各変数を質問としたときの情報利得を計算する。
6. 情報利得が最も大きい質問を選択し、それをノードとし、分割後のデータを新たなデータの集合とみなし、上記操作を繰り返す。

	食性	**発生形態**	**体温**	**分類**
ペンギン	肉食	卵生	恒温	鳥類
ライオン	肉食	胎生	恒温	哺乳類
ウシ	草食	胎生	恒温	哺乳類
トカゲ	肉食	卵生	変温	爬虫類
ブンチョウ	草食	卵生	恒温	鳥類

決定木の構築方法：ID3アルゴリズム

1. ルートノードNを作成し、全データを所属させる。
2. Nに所属するデータが、全て同じ分類先なら、処理を終了する。
3. Nに所属するデータの平均情報量を計算する。
4. 各変数を質問とした時の平均情報量を求める。
5. 各変数を質問としたときの情報利得を計算する。
6. 情報利得が最も大きい質問を選択し、それをノードとし、分割後のデータを新たなデータの集合とみなし、上記操作を繰り返す。

N
ペンギン
ライオン
ウシ
トカゲ
ブンチョウ

	食性	**発生形態**	**体温**	**分類**
ペンギン	肉食	卵生	恒温	鳥類
ライオン	肉食	胎生	恒温	哺乳類
ウシ	草食	胎生	恒温	哺乳類
トカゲ	肉食	卵生	変温	爬虫類
ブンチョウ	草食	卵生	恒温	鳥類

決定木の構築方法：ID3アルゴリズム

N

ペンギン
ライオン
ウシ
トカゲ
ブンチョウ

1. ルートノードNを作成し、全データを所属させる。
2. Nに所属するデータが、全て同じ分類先なら、処理を終了する。
3. Nに所属するデータの平均情報量を計算する。
4. 各変数を質問とした時の平均情報量を求める。
5. 各変数を質問としたときの情報利得を計算する。
6. 情報利得が最も大きい質問を選択し、それをノードとし、分割後のデータを新たなデータの集合とみなし、上記操作を繰り返す。

	食性	**発生形態**	**体温**	**分類**
ペンギン	肉食	卵生	恒温	鳥類
ライオン	肉食	胎生	恒温	哺乳類
ウシ	草食	胎生	恒温	哺乳類
トカゲ	肉食	卵生	変温	爬虫類
ブンチョウ	草食	卵生	恒温	鳥類

決定木の構築方法：ID3アルゴリズム

N

ペンギン
ライオン
ウシ
トカゲ
ブンチョウ

1. ルートノードNを作成し、全データを所属させる。
2. Nに所属するデータが、全て同じ分類先なら、処理を終了する。
3. Nに所属するデータの平均情報量を計算する。
4. 各変数を質問とした時の平均情報量を求める。
5. 各変数を質問としたときの情報利得を計算する。
6. 情報利得が最も大きい質問を選択し、それをノードとし、分割後のデータを新たなデータの集合とみなし、上記操作を繰り返す。

	食性	**発生形態**	**体温**	**分類**
ペンギン	肉食	卵生	恒温	鳥類
ライオン	肉食	胎生	恒温	哺乳類
ウシ	草食	胎生	恒温	哺乳類
トカゲ	肉食	卵生	変温	爬虫類
ブンチョウ	草食	卵生	恒温	鳥類

鳥類２件，爬虫類１件，哺乳類２件なので，

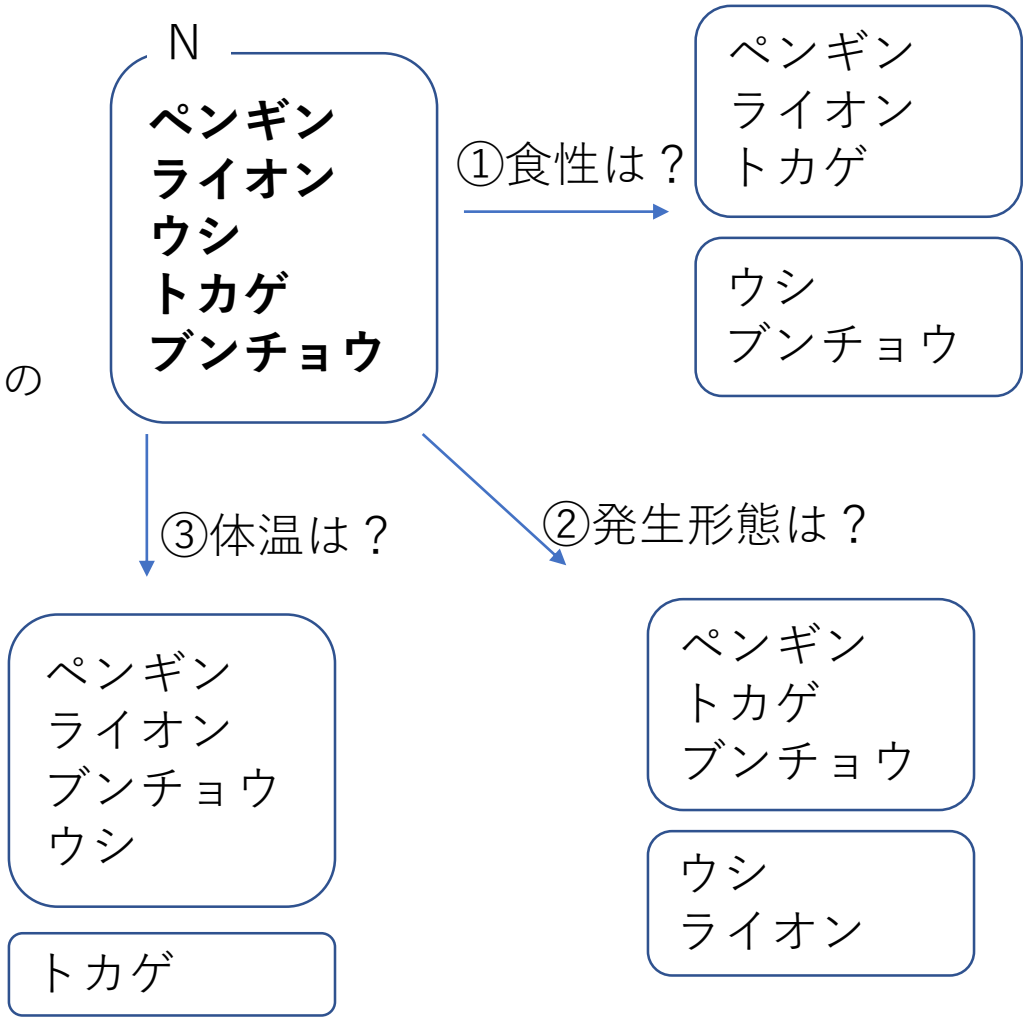
$$H(x) = -\frac{2}{5}\log_3\frac{2}{5} - \frac{1}{5}\log_3\frac{1}{5} - \frac{2}{5}\log_3\frac{2}{5} \simeq 0.960$$

補足： $H[x] = -\sum_x p(x)\log p(x)$

決定木の構築方法：ID3アルゴリズム

1. ルートノードNを作成し、全データを所属させる。
2. Nに所属するデータが、全て同じ分類先なら、処理を終了する。
3. Nに所属するデータの平均情報量を計算する。
4. 各変数を質問とした時の平均情報量を求める。
5. 各変数を質問としたときの情報利得を計算する。
6. 情報利得が最も大きい質問を選択し、それをノードとし、分割後のデータを新たなデータの集合とみなし、上記操作を繰り返す。

	食性	**発生形態**	**体温**	**分類**
ペンギン	肉食	卵生	恒温	鳥類
ライオン	肉食	胎生	恒温	哺乳類
ウシ	草食	胎生	恒温	哺乳類
トカゲ	肉食	卵生	変温	爬虫類
ブンチョウ	草食	卵生	恒温	鳥類



決定木の構築方法：ID3アルゴリズム

1. ルートノードNを作成し、全データを所属させる。
2. Nに所属するデータが、全て同じ分類先なら、処理を終了する。
3. Nに所属するデータの平均情報量を計算する。
4. 各変数を質問とした時の平均情報量を求める。
5. 各変数を質問としたときの情報利得を計算する。
6. 情報利得が最も大きい質問を選択し、それをノードとし、分割後のデータを新たなデータの集合とみなし、上記操作を繰り返す。

	食性	**発生形態**	**体温**	**分類**
ペンギン	肉食	卵生	恒温	鳥類
ライオン	肉食	胎生	恒温	哺乳類
ウシ	草食	胎生	恒温	哺乳類
トカゲ	肉食	卵生	変温	爬虫類
ブンチョウ	草食	卵生	恒温	鳥類

肉食は，鳥類１件，爬虫類１件，
哺乳類１件なので，

$$-\frac{1}{3}\log_3\frac{1}{3}-\frac{1}{3}\log_3\frac{1}{3}-\frac{1}{3}\log_3\frac{1}{3}=1.0$$

草食は，鳥類１件，爬虫類０件，
哺乳類１件なので，

$$-\frac{1}{2}\log_3\frac{1}{2}-\frac{0}{2}\log_3\frac{0}{2}-\frac{1}{2}\log_3\frac{1}{2}\simeq 0.631$$

よって，食性で分けた際の平均情報量の期待値は，

$$\frac{3}{5}\times 1.0+\frac{2}{5}\times 0.631=0.852$$

決定木の構築方法：ID3アルゴリズム

1. ルートノードNを作成し、全データを所属させる。
2. Nに所属するデータが、全て同じ分類先なら、処理を終了する。
3. Nに所属するデータの平均情報量を計算する。
4. 各変数を質問とした時の平均情報量を求める。
5. 各変数を質問としたときの情報利得を計算する。
6. 情報利得が最も大きい質問を選択し、それをノードとし、分割後のデータを新たなデータの集合とみなし、上記操作を繰り返す。

	食性	**発生形態**	**体温**	**分類**
ペンギン	肉食	卵生	恒温	鳥類
ライオン	肉食	胎生	恒温	哺乳類
ウシ	草食	胎生	恒温	哺乳類
トカゲ	肉食	卵生	変温	爬虫類
ブンチョウ	草食	卵生	恒温	鳥類

①食性で分けた際の平均情報量
　=0.852
②発生形態で分けたときの平均情報量
　=0.348
③体温で分けたときの平均情報量
　=0.505

情報利得：元々の平均情報量から、質問によるデータ分割後の平均情報量を引いた値.
分割した際にどのくらい情報量が減少したかを示す.

①0.96 – 0.852 = 0.108
②0.96 – 0.348 = 0.612
③0.96 – 0.505 = 0.455

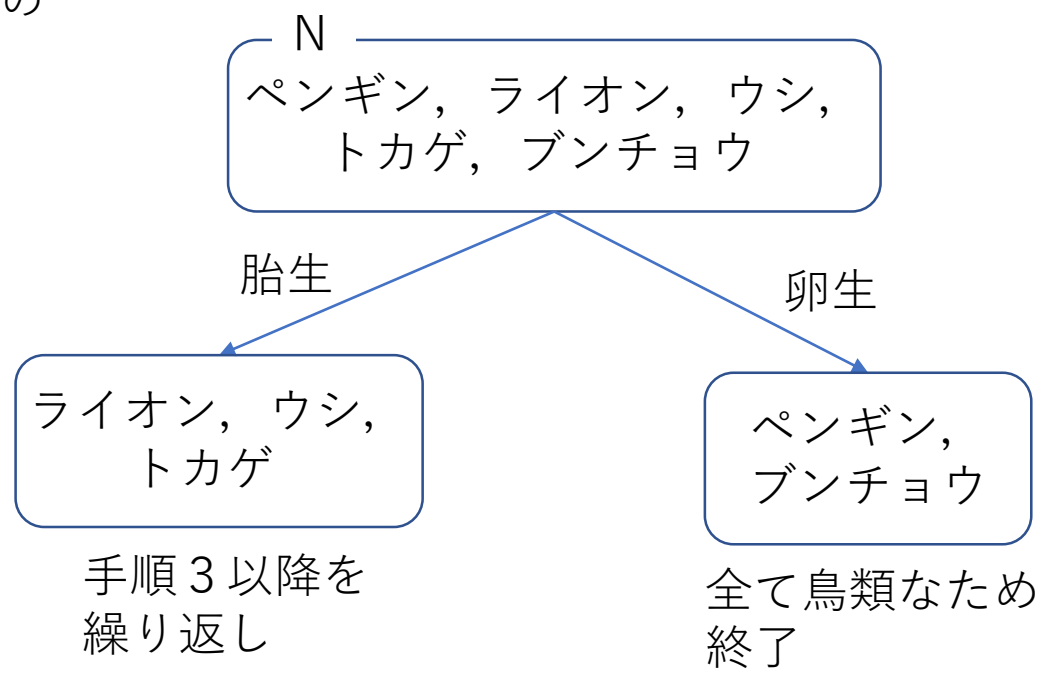
決定木の構築方法：ID3アルゴリズム

1. ルートノードNを作成し、全データを所属させる。
2. Nに所属するデータが、全て同じ分類先なら、処理を終了する。
3. Nに所属するデータの平均情報量を計算する。
4. 各変数を質問とした時の平均情報量を求める。
5. 各変数を質問としたときの情報利得を計算する。
6. 情報利得が最も大きい質問を選択し、それをノードとし、分割後のデータを新たなデータの集合とみなし、上記操作を繰り返す。

	食性	**発生形態**	**体温**	**分類**
ペンギン	肉食	卵生	恒温	鳥類
ライオン	肉食	胎生	恒温	哺乳類
ウシ	草食	胎生	恒温	哺乳類
トカゲ	肉食	卵生	変温	爬虫類
ブンチョウ	草食	卵生	恒温	鳥類

①0.96 – 0.852 = 0.108
②0.96 – 0.348 = 0.612
③0.96 – 0.505 = 0.455

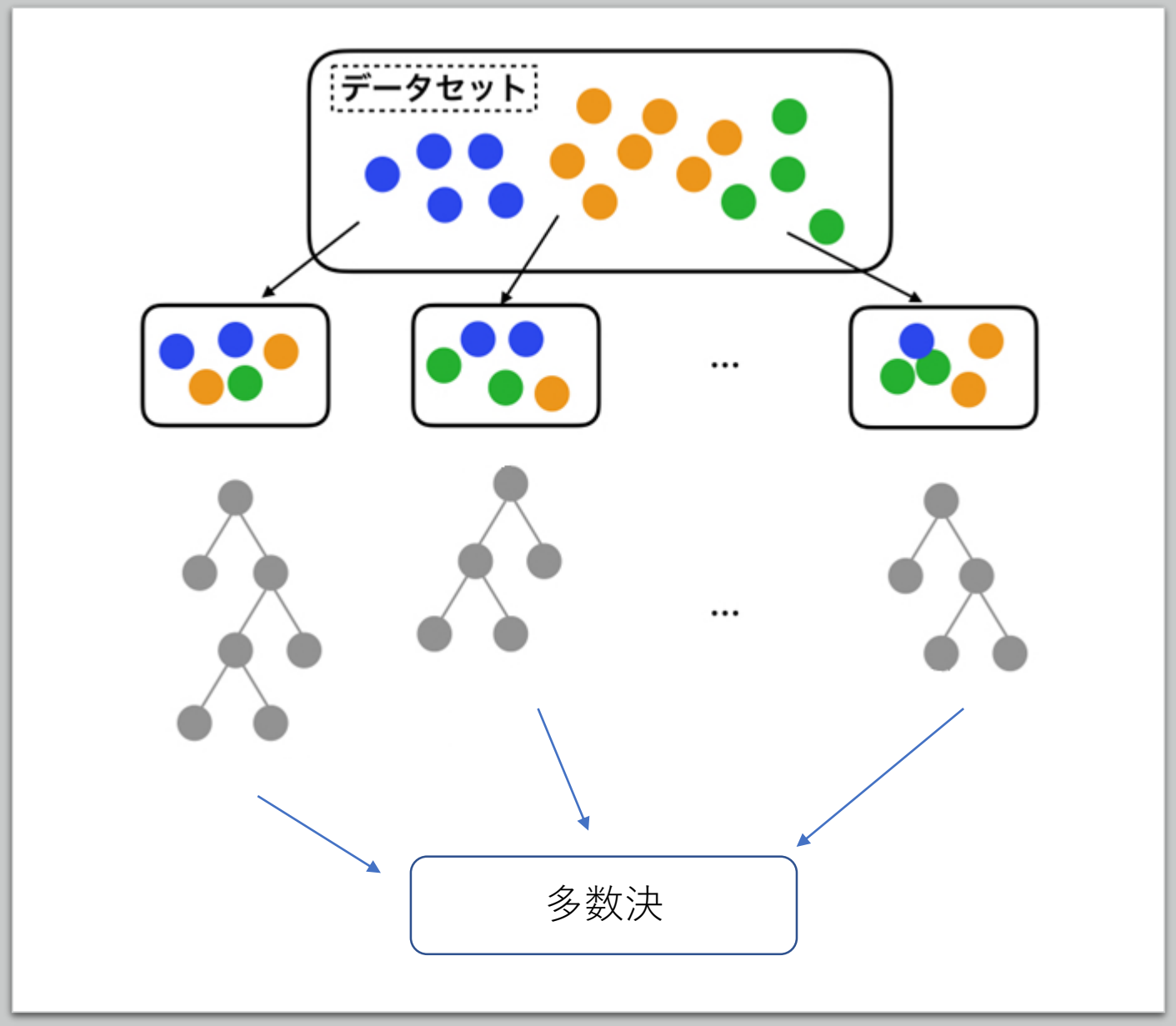
②が一番大きいため，有効であることを示している.
⇒まずは発生形態で分ける



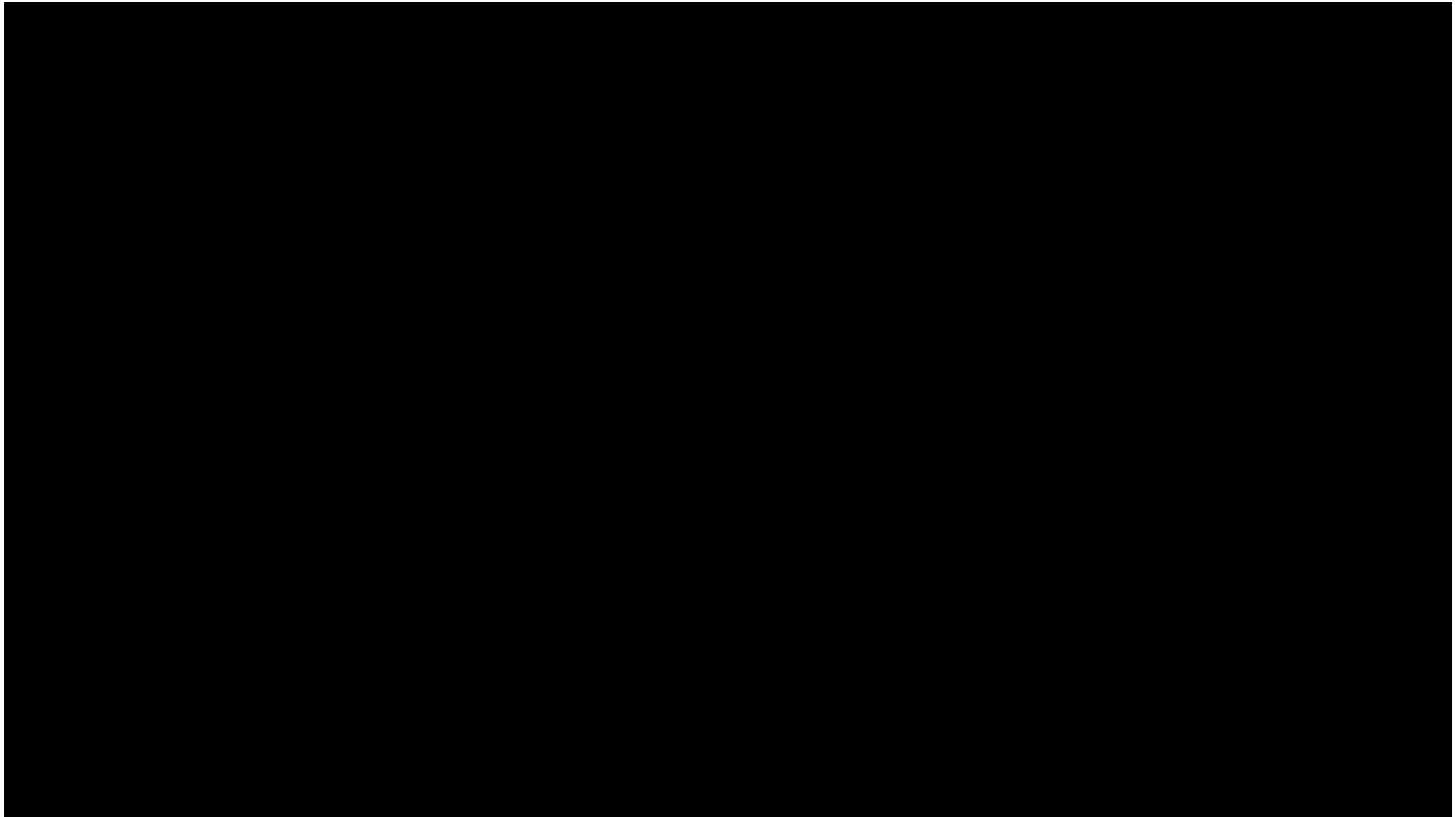
第二部の終わり

ランダム
フォレスト

- データをランダムに抽出し，そこから決定木を作る
- 新規データが与えられた際，それらの木を当てはめてみて多数決で決める
- 複数のモデルを作って一つの結果を出すことを**アンサンブル学習**という

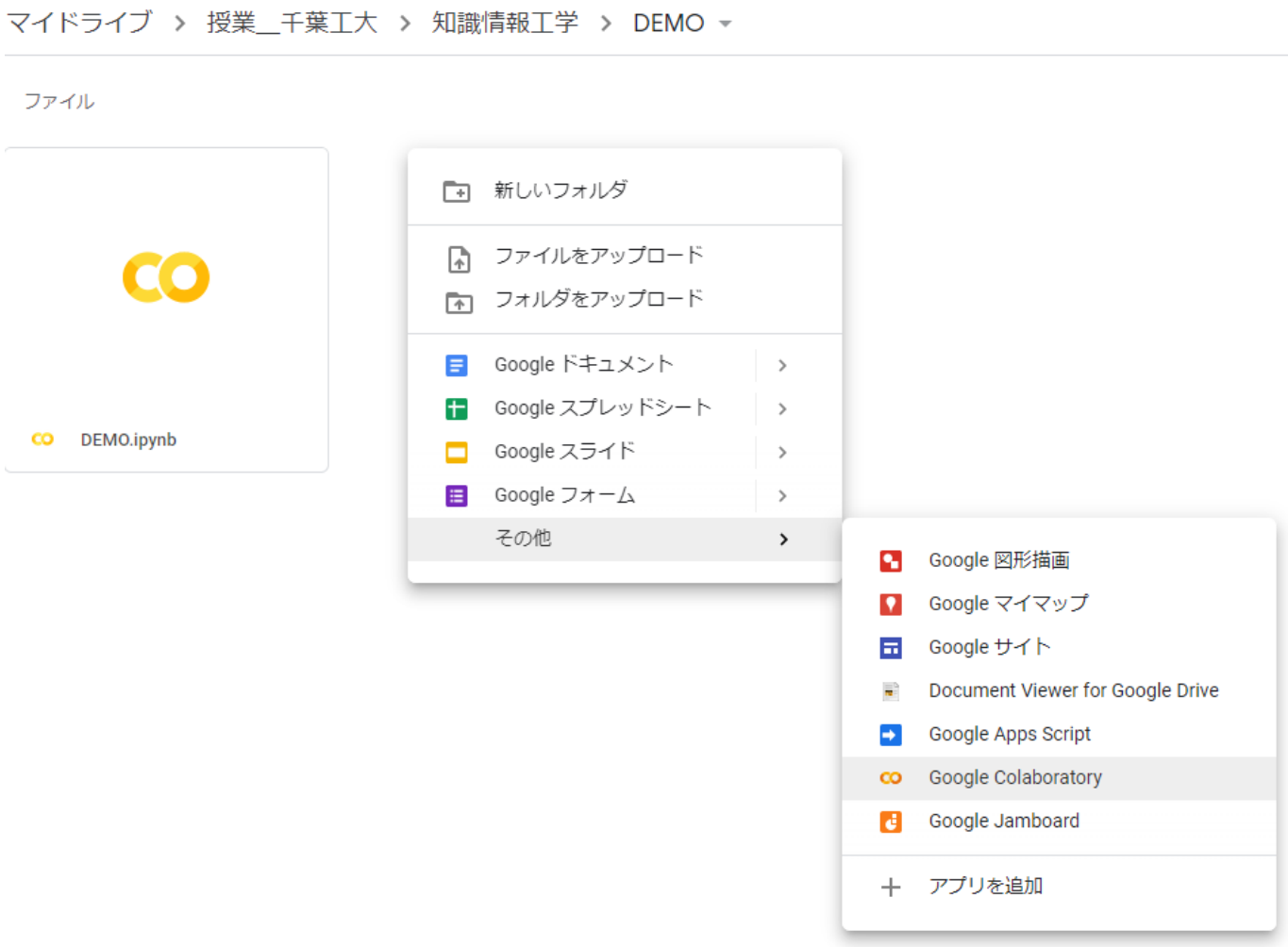


ランダムフォレスト（１：40～）



Google colaboratory

- Google driveで右クリック.
  その他からGoogle colaboratory を選択するだけでPythonを実行できる環境が得られる.
- GPUも使うことができる.

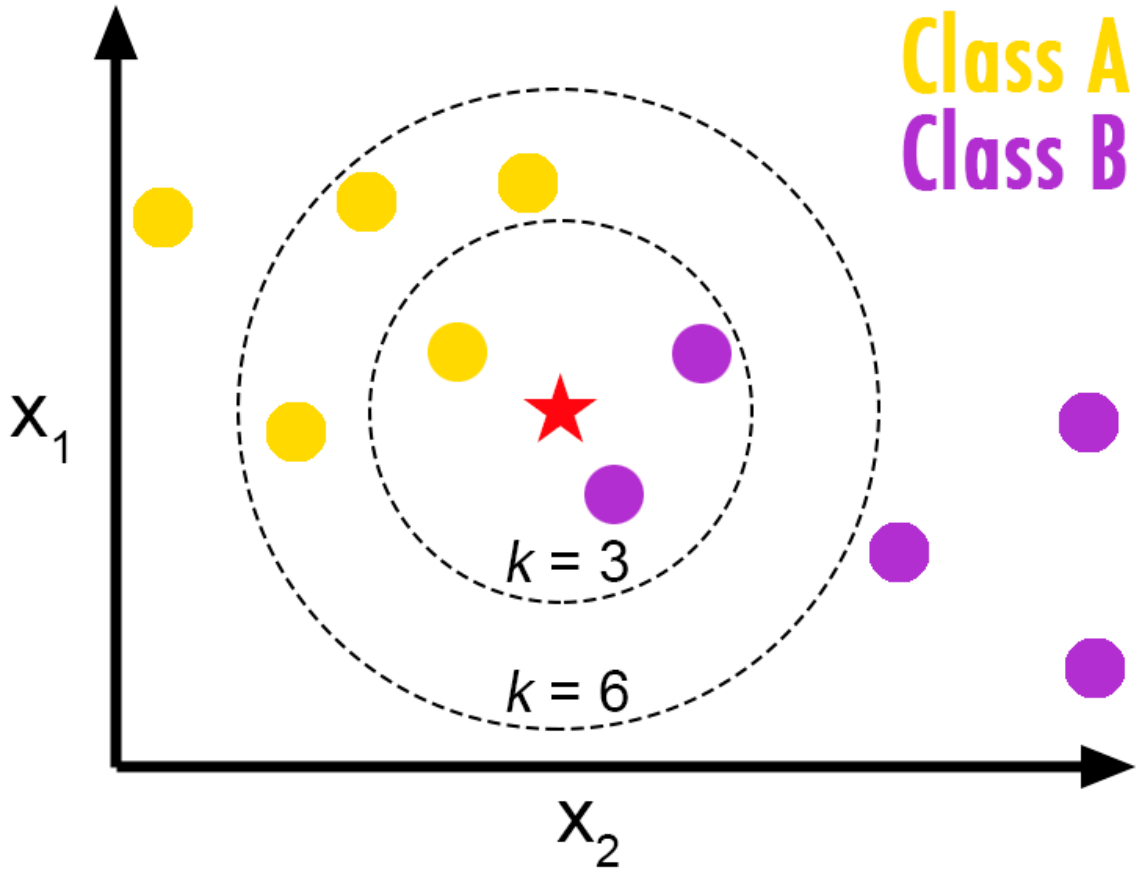


python

```
# random forest
from sklearn.ensemble import RandomForestClassifier
import pandas as pd
from sklearn import metrics
from sklearn.model_selection import train_test_split

df = pd.read_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv',sep=";",encoding="utf-8")

df.head()
train_x = df.drop(['quality'], axis=1)
train_y = df['quality']
(train_x, test_x ,train_y, test_y) = train_test_split(train_x, train_y, test_size = 0.3)
clf = RandomForestClassifier(max_depth=30, n_estimators=30, random_state=42)
clf.fit(train_x, train_y)#訓練用データで学習
y_pred = clf.predict(test_x)#テスト用データの予測
accuracy = metrics.accuracy_score(test_y, y_pred)
print('Accuracy: {}'.format(accuracy))
```

k-Nearest Neighbor (k近傍法)

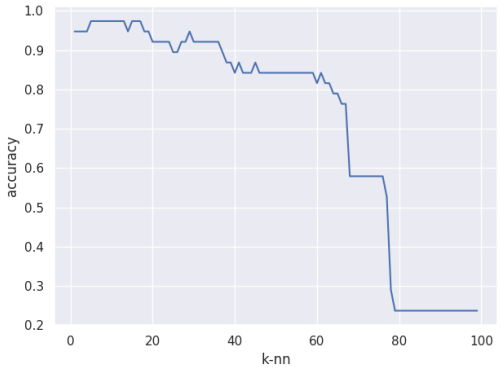
- 新規データが来た際，一番近いk個の学習データを見て，多数決で分類先を判定する
- 説明変数が数値で，目的変数がラベルやカテゴリの時に有効
- 右図の場合，新規データ★は，
  k=3のときはClass Bになり，
  k=6のときはClass Aと分類される



Pythonでのk近傍法（knn）

```
from sklearn.datasets import load_iris
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn import metrics
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier

iris = load_iris()
iris_target_data = pd.DataFrame(iris.target, columns=['Species'])
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)

X_train, X_test, Y_train, Y_test = train_test_split(iris_df, iris_target_data)
print(iris_df)
print(iris_target_data)
knn = KNeighborsClassifier(n_neighbors=6)
knn.fit(X_train, Y_train)
Y_pred = knn.predict(X_test)
print(metrics.accuracy_score(Y_test, Y_pred))
```

```
accuracy_list = []
sns.set()
k_range = range(1, 100)
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, Y_train)
    Y_pred = knn.predict(X_test)
    accuracy_list.append(metrics.accuracy_score(Y_test, Y_pred))

figure = plt.figure()
ax = figure.add_subplot(111)
ax.plot(k_range, accuracy_list)
ax.set_xlabel('k-nn')
ax.set_ylabel('accuracy')
plt.show()
```

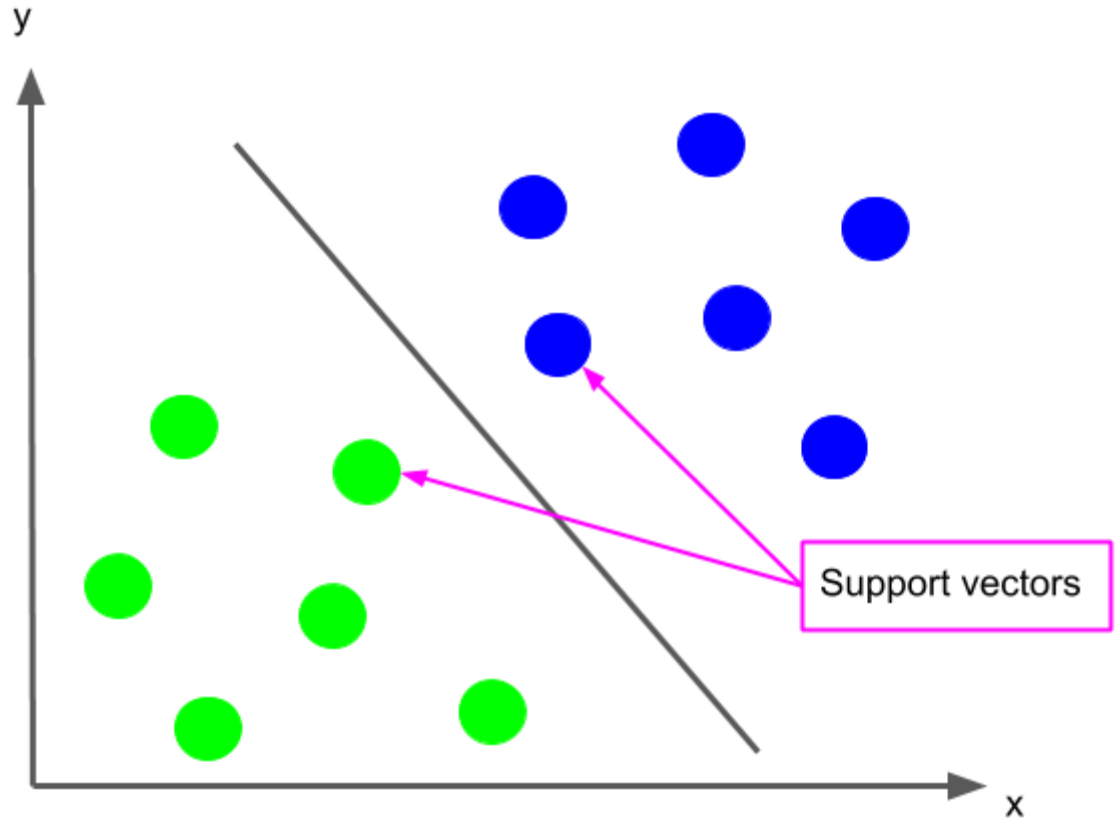


横軸は設定するｋの値

※ラベル1種類につき50個のデータセットなので急激に落ちる

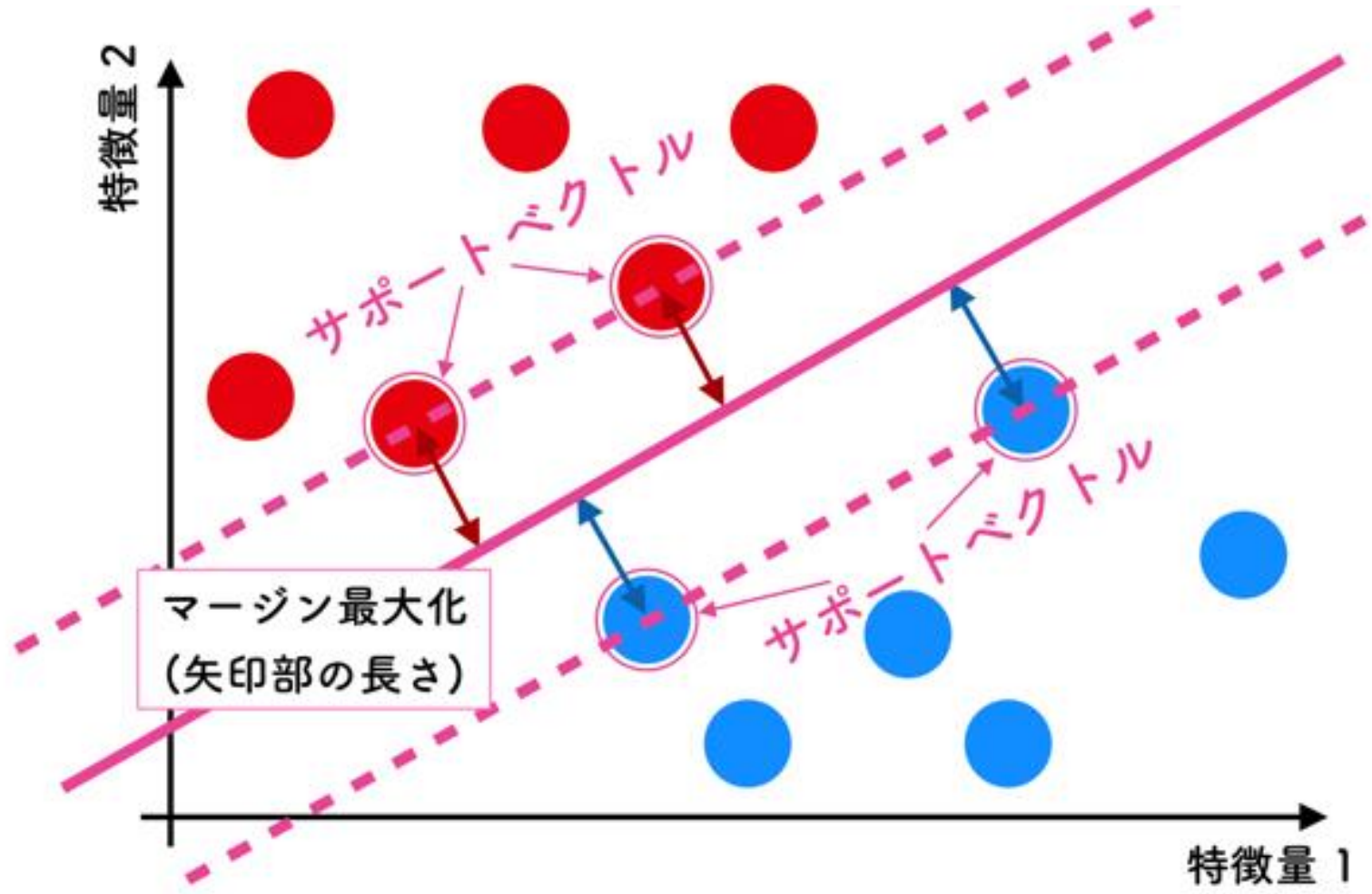
サポートベクターマシン（SVM）

- 学習し，データがどちらに分類されるかのモデルを作る.
- ２カテゴリのデータを上手く分けられる境界線を探す.
- 直線に最も近いデータが大きな役割を果たすためこのような名前となった

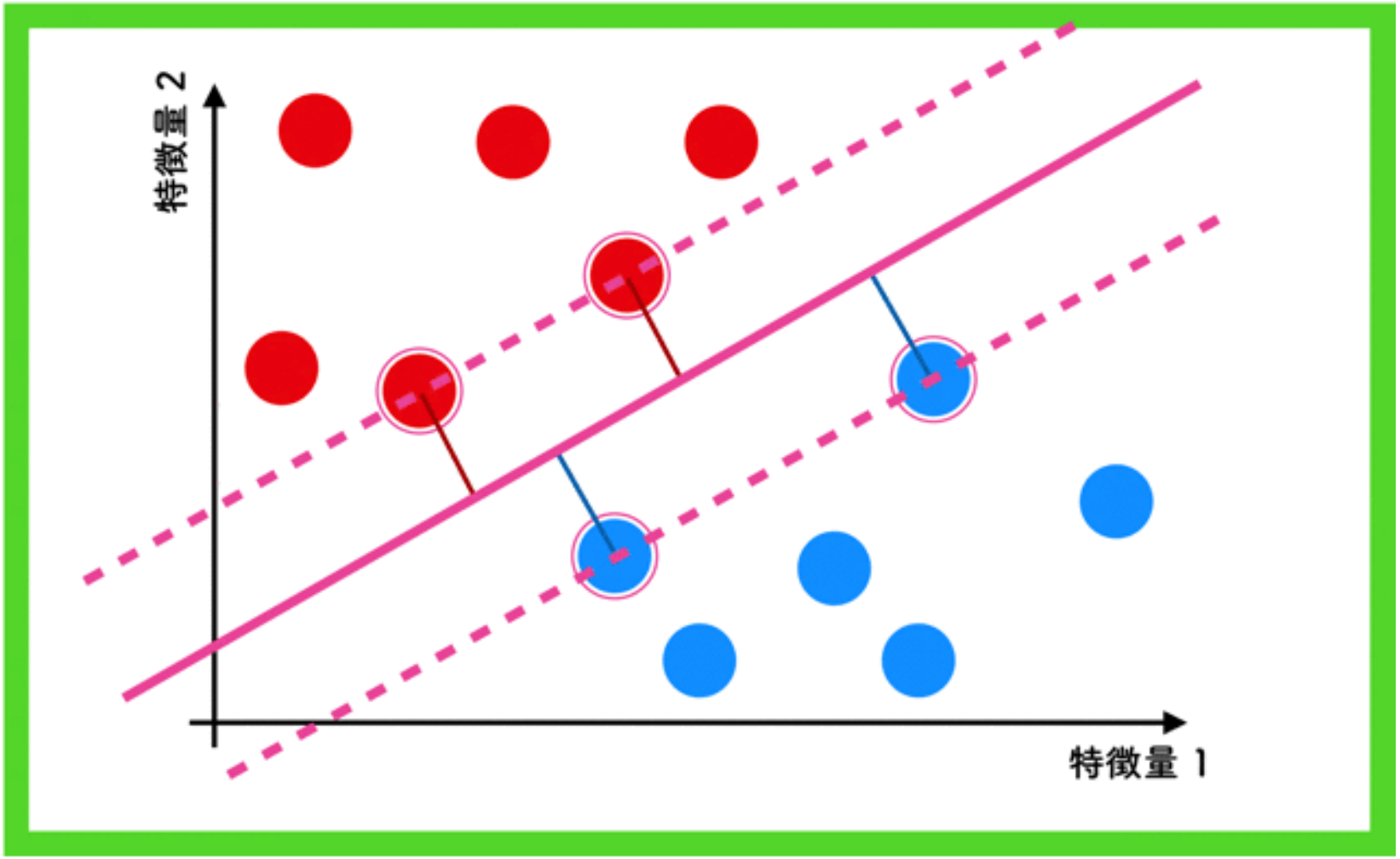


サポートベクターマシン（SVM）

- サポートベクトルと，境界線が一番遠くなるような線を選択する
- その距離のことを**マージン**と呼ぶ



マージン最大化(gif)



python

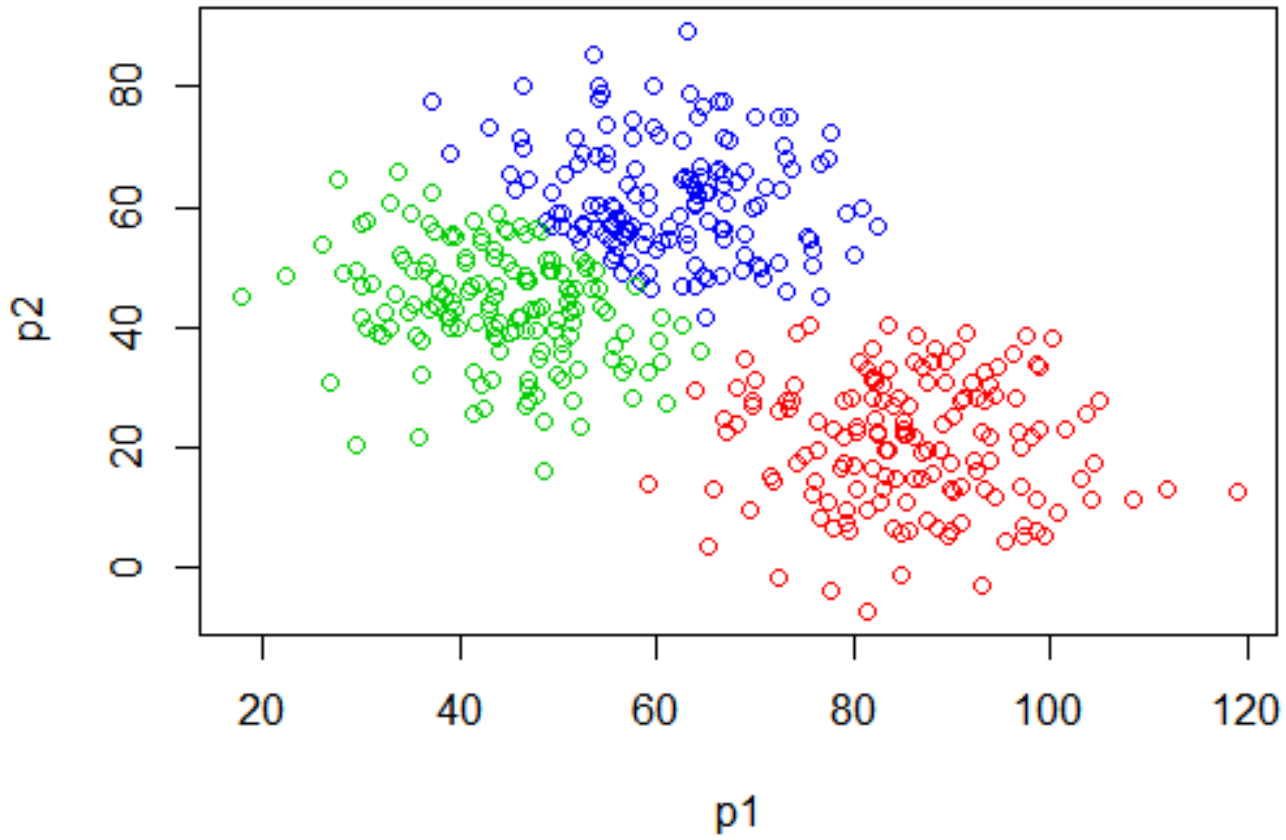
```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris

iris = load_iris()

dataset = pd.DataFrame(data =
iris['data'], columns =
iris['feature_names'])
dataset['species'] = iris['target']
dataset.head()

# 目的変数(Y)、説明変数(X)
Y = np.array(dataset['species'])
X =
np.array(dataset[iris['feature_names']])
```

```
# データの分割
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.3, random_state=0)

from sklearn.svm import SVC
model = SVC(gamma='scale')
model.fit(X_train, Y_train)

Y_pred = model.predict(X_test)

print(Y_test[:10])
print(Y_pred[:10])
print(metrics.accuracy_score(Y_test, Y_pred))
```

k-means (k平均法)

- 教師なし学習である.
- 与えられたデータセットに対し，似たもの同士をk個でクラスタリング（グループ化）する.
- 各グループに意味があるかどうかは解釈次第

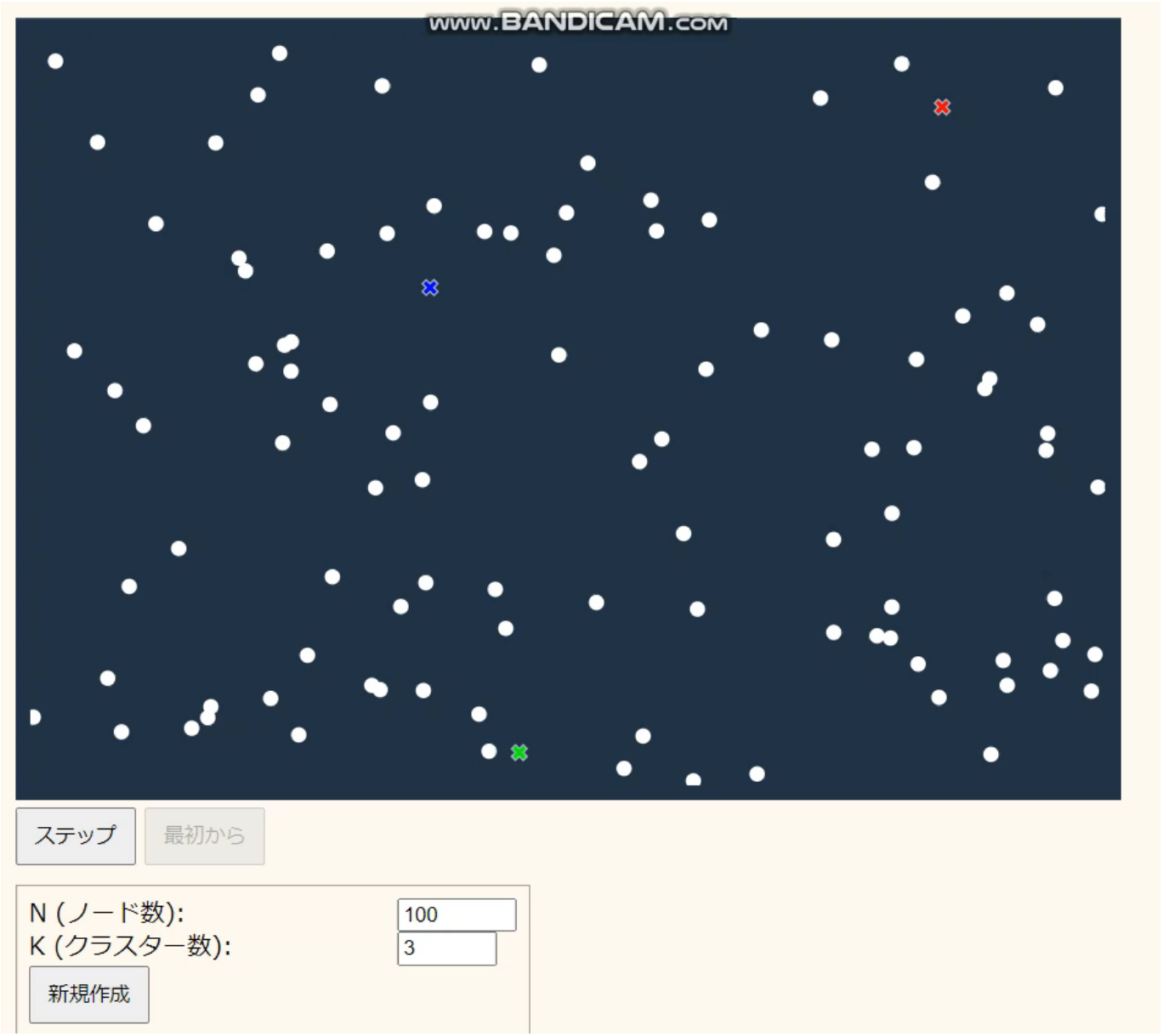
k=3のとき



k-meansの仕組み

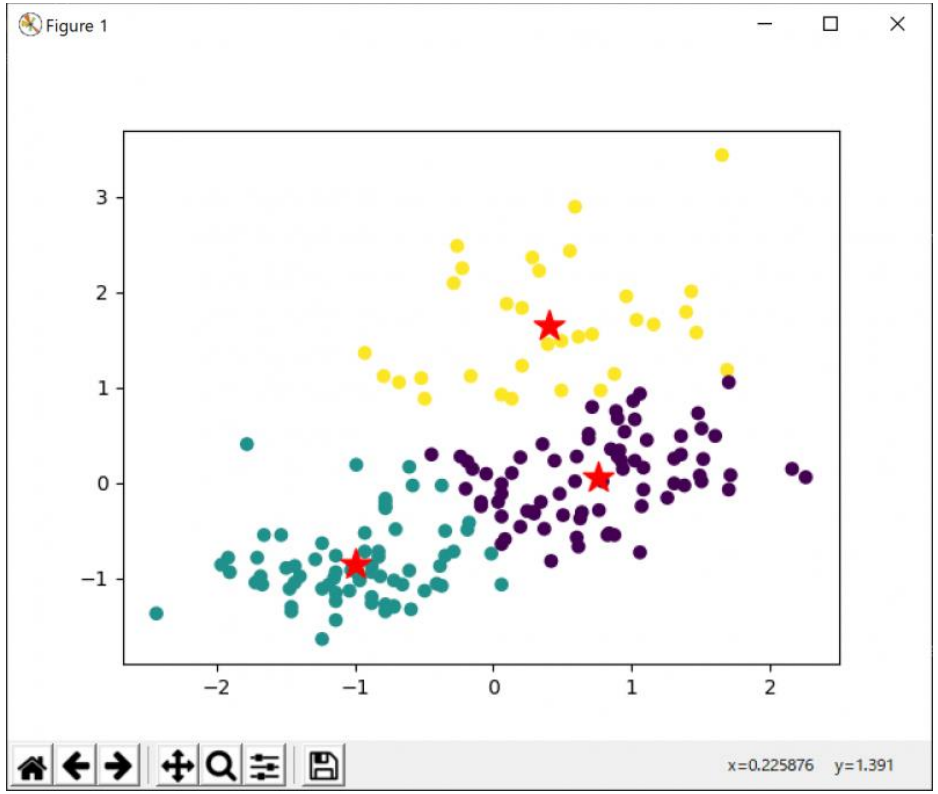
1. k個の点をランダムに配置する
2. 各データと各点の距離を計算する
3. データと一番近い点を結びつける
4. 各点をクラスタの重心に移動させる
5. ２～４を繰り返す

http://tech.nitoyon.com/ja/blog/2013/11/07/k-means/

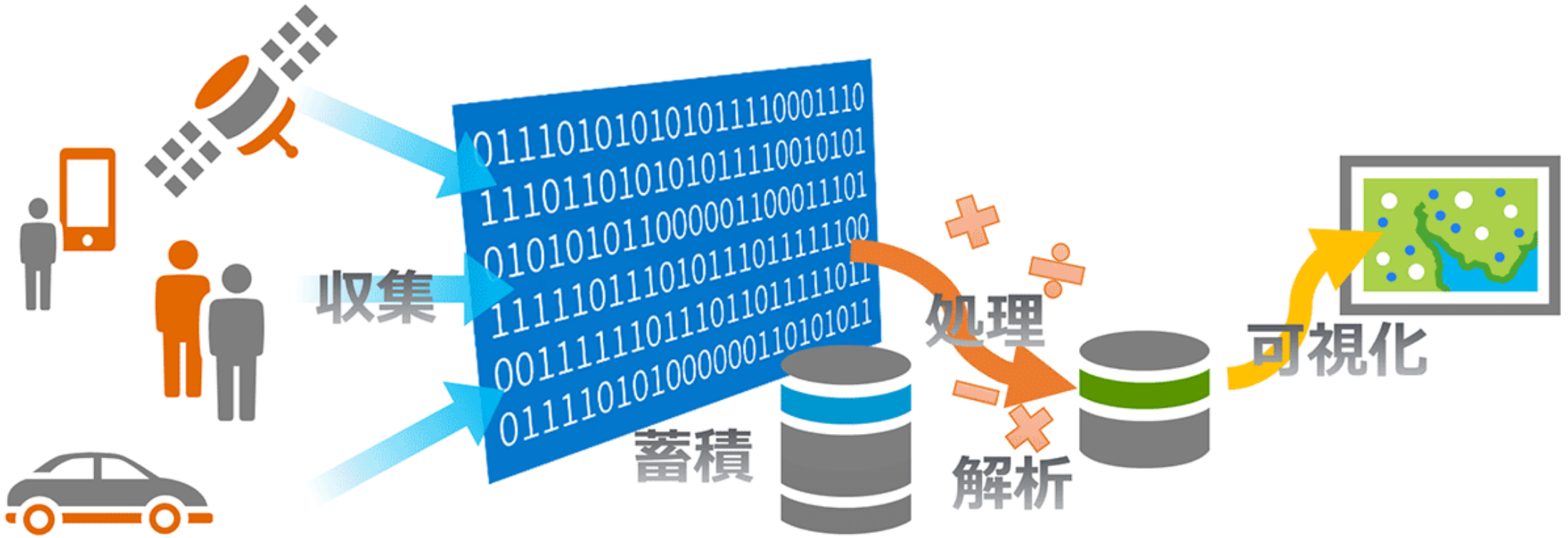


k-meansのソースコード

```
from matplotlib import pyplot as plt
from sklearn import datasets, preprocessing
from sklearn.cluster import KMeans
import numpy as np
import pandas as pd
# datasetの読み込み
wine_data = datasets.load_wine()
# DataFrameに変換
df = pd.DataFrame(wine_data.data, columns=wine_data.feature_names)
print(df.head())
# データの整形
X=df[["alcohol","color_intensity"]]
sc=preprocessing.StandardScaler()
sc.fit(X)
X_norm=sc.transform(X)
# クラスタリング
cls = KMeans(n_clusters=3)
result = cls.fit(X_norm)
# 結果を出力
plt.scatter(X_norm[:,0],X_norm[:,1], c=result.labels_)
plt.scatter(result.cluster_centers_[:,0],result.cluster_centers_[:,1],s=250, marker='*',c='red')
plt.show()
```



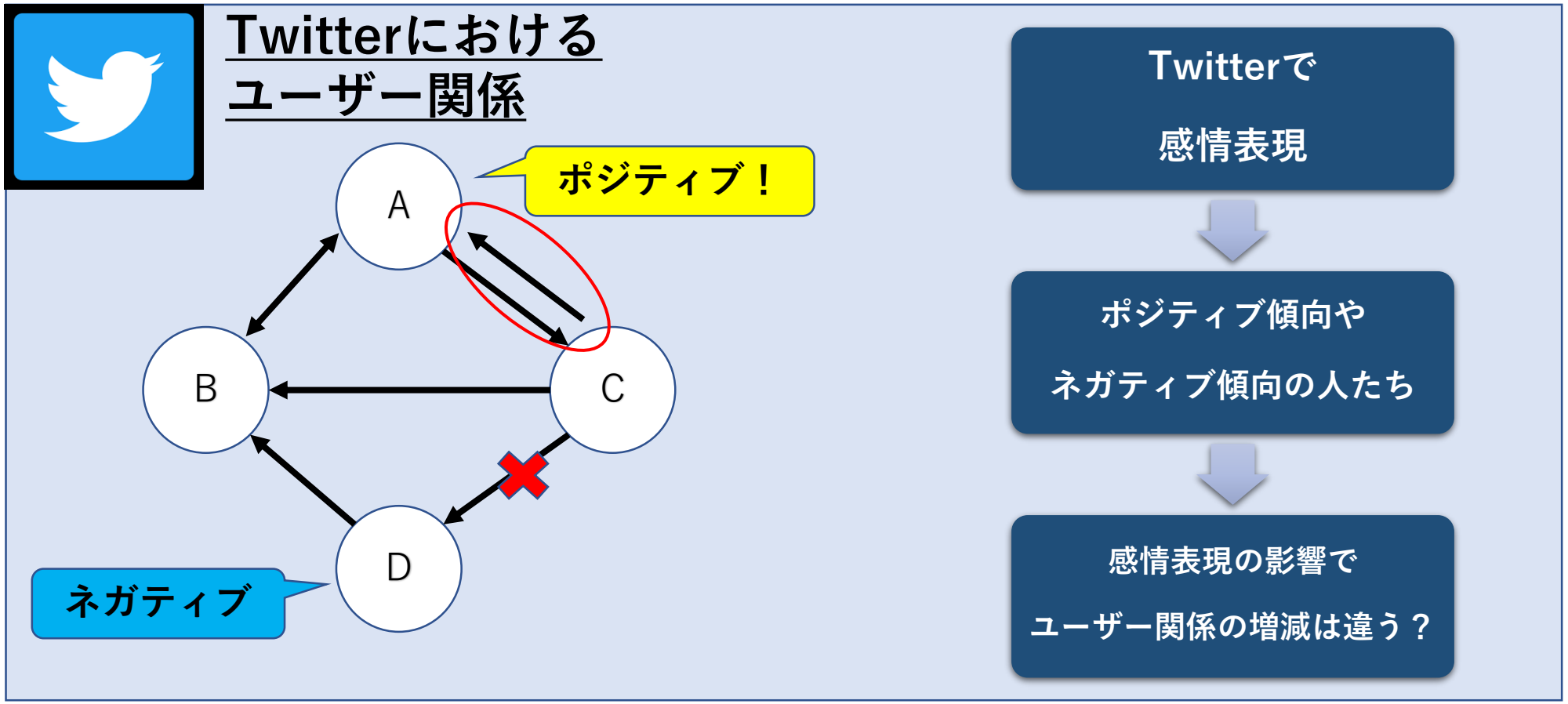
データマイニング



データマイニングとは

- データマイニングとは、未知の結果を予測するために、大量のデータセットに含まれている異常値、パターン、相関を発見するプロセス
- 膨大なデータから、人間では掴めないような知見を発見するため、機械学習がよく用いられる。
- 次からは、自分が行ったデータ処理の実例を紹介する。

Twitter上の感情表現がユーザー関係に及ぼす影響の分析



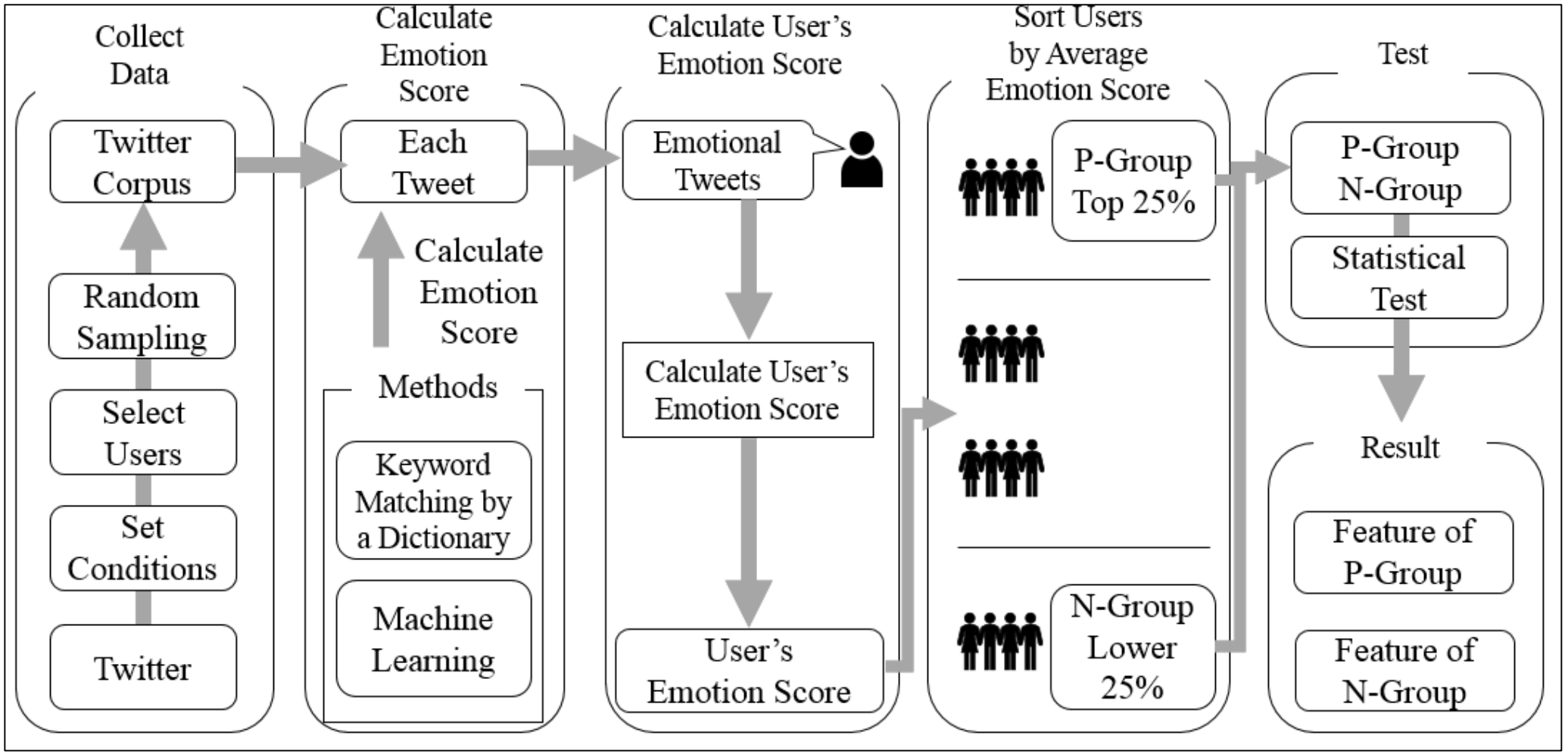
従来の研究

- **ユーザーの特徴や振る舞いに焦点を当てた研究[1, 2]が多い**
- **しかし，感情からユーザー関係に焦点を当てた研究は少ない**

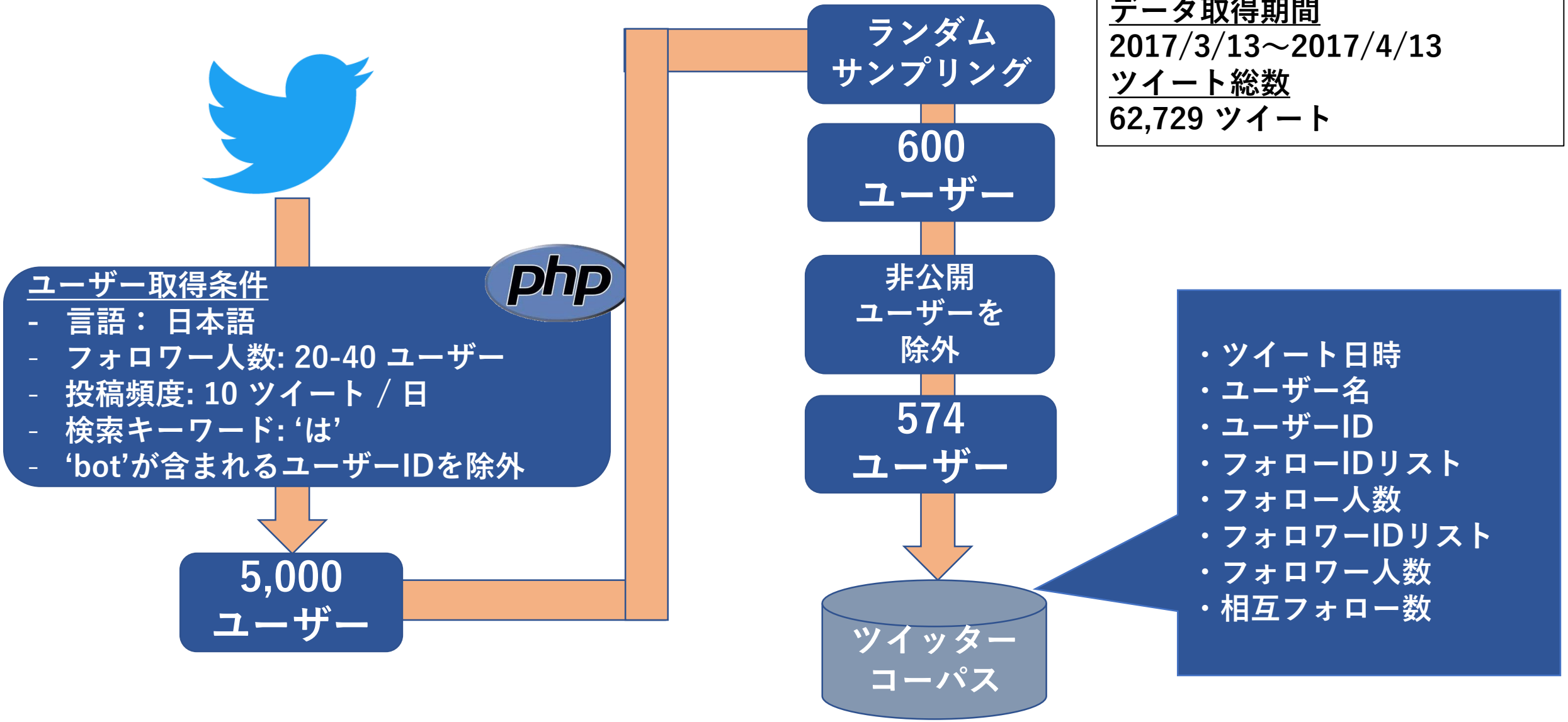
- **感情という観点から，ユーザーの振る舞いに対して研究を行う**

1. X. Ruan, S. Wilson, R .Mihalcea, Finding Optimists and Pessimists on Twitter, In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp.320–325, 2016
2. Fujita, M., Watanabe, J. I., Kawamoto, K., Akitomi, T., Ara, K., A method for analyzing influence of emotions of posts in SNS conversations. In Proceedings of the Social Intelligence and Technology, 20–27, 2013

アプローチの概要

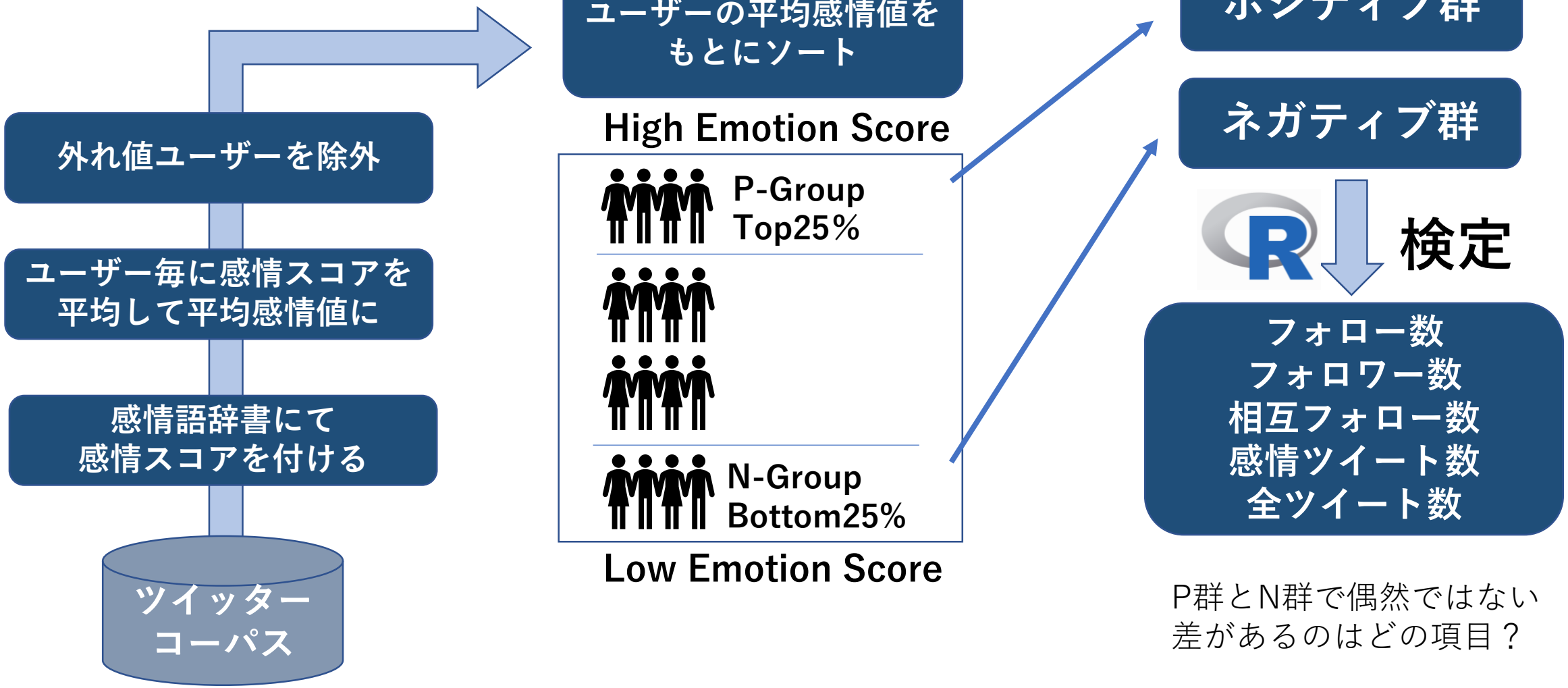

Collect Data
Twitter Corpus
Random Sampling
Select Users
Set Conditions
Twitter
Calculate Emotion Score
Each Tweet
Calculate Emotion Score
Methods
Keyword Matching by a Dictionary
Machine Learning
Calculate User's Emotion Score
Emotional Tweets
Calculate User's Emotion Score
User's Emotion Score
Sort Users by Average Emotion Score
P-Group Top 25%
N-Group Lower 25%
Test
P-Group N-Group
Statistical Test
Result
Feature of P-Group
Feature of N-Group

データ取得方法



データ取得期間
2017/3/13～2017/4/13
ツイート総数
62,729 ツイート

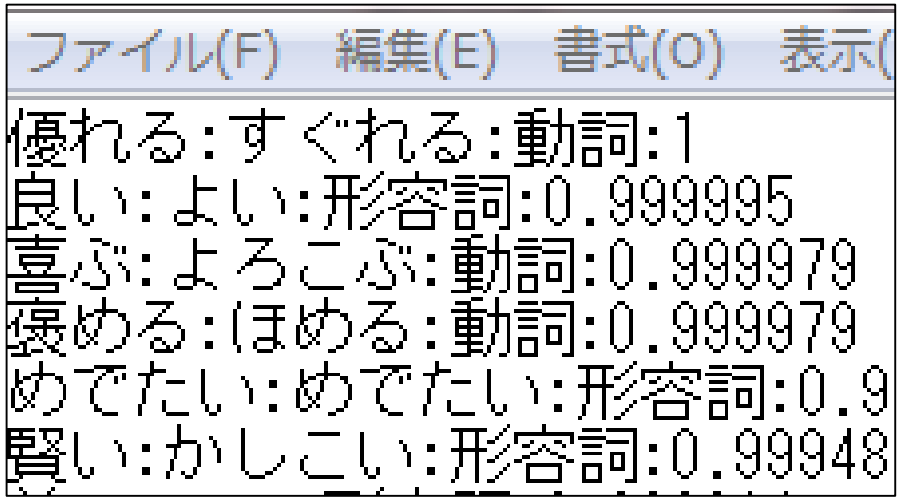
実験の流れ



P群とN群で偶然ではない
差があるのはどの項目？

感情評価の方法

単語感情極性対応表（辞書A）



定量値での評価
+1(P) ～ -1(N)

ツイートを
単語単位に分解

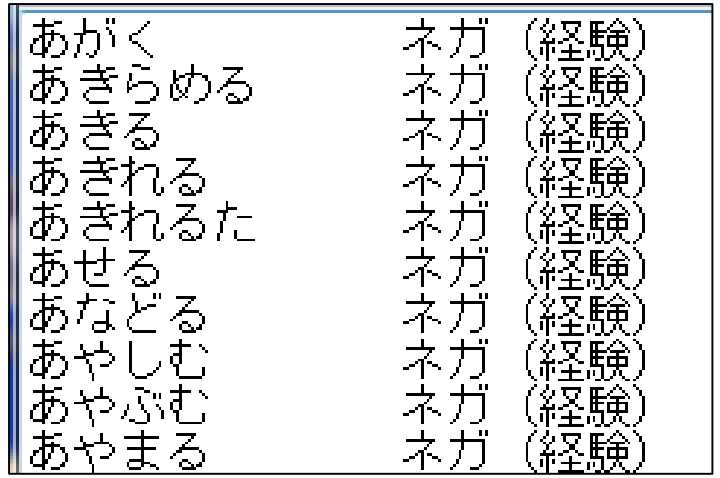
↓

キーワード
マッチング

↓

スコアの合計が
１文章のスコア

日本語評価極性辞書（辞書B）



ラベルでの評価
ポジ：+1
ネガ： -1

例：発表 / は / 緊張 / する
A: -0.3 -0.5 ＝ -0.8
B: -1 -1 ＝ -2

Brunner-Munzel検定の結果

	辞書A	辞書B				
	Case 1	Case 1	Case 2	Case 2	Case 3	Case 3
フォロー人数増減数	n.s.	**	n.s.	*	*	*
フォロワー人数増減数	n.s.	**	n.s.	**	*	*
相互フォロー増減数	n.s.	**	n.s.	*	**	**
感情ツイート数	**	n.s.	**	n.s.	n.s.	n.s.
全ツイート数	n.s	n.s.	n.s.	n.s.	n.s.	n.s.

*p<0.05, **p<0.01, n.s.: not significant

・Case 1 ＝ 感情ツイート15個を無作為に選んで，平均感情値を求めた場合
・Case 2 ＝ 2つの辞書でスコアが付いた感情ツイート数が，9以下のユーザーを除外
・Case 3 ＝ 2つの辞書でスコアが付いた感情ツイート数が，19以下のユーザーを除外

ナイーブベイズ分類

- **単語が独立して生起すると見なす**
- **データの事後確率（条件付き確率）を最大化するカテゴリに分類する**

文章の各単語に対し、それぞれのカテゴリに含まれる確率を計算する。文章内の単語が持つ確率をカテゴリごとに足し合わせ、最大となるカテゴリに分類する。

・文章(D)がカテゴリ(C)に属する確率P(C|D)

$$P(C|D) = \frac{P(D|C)P(C)}{P(D)}$$

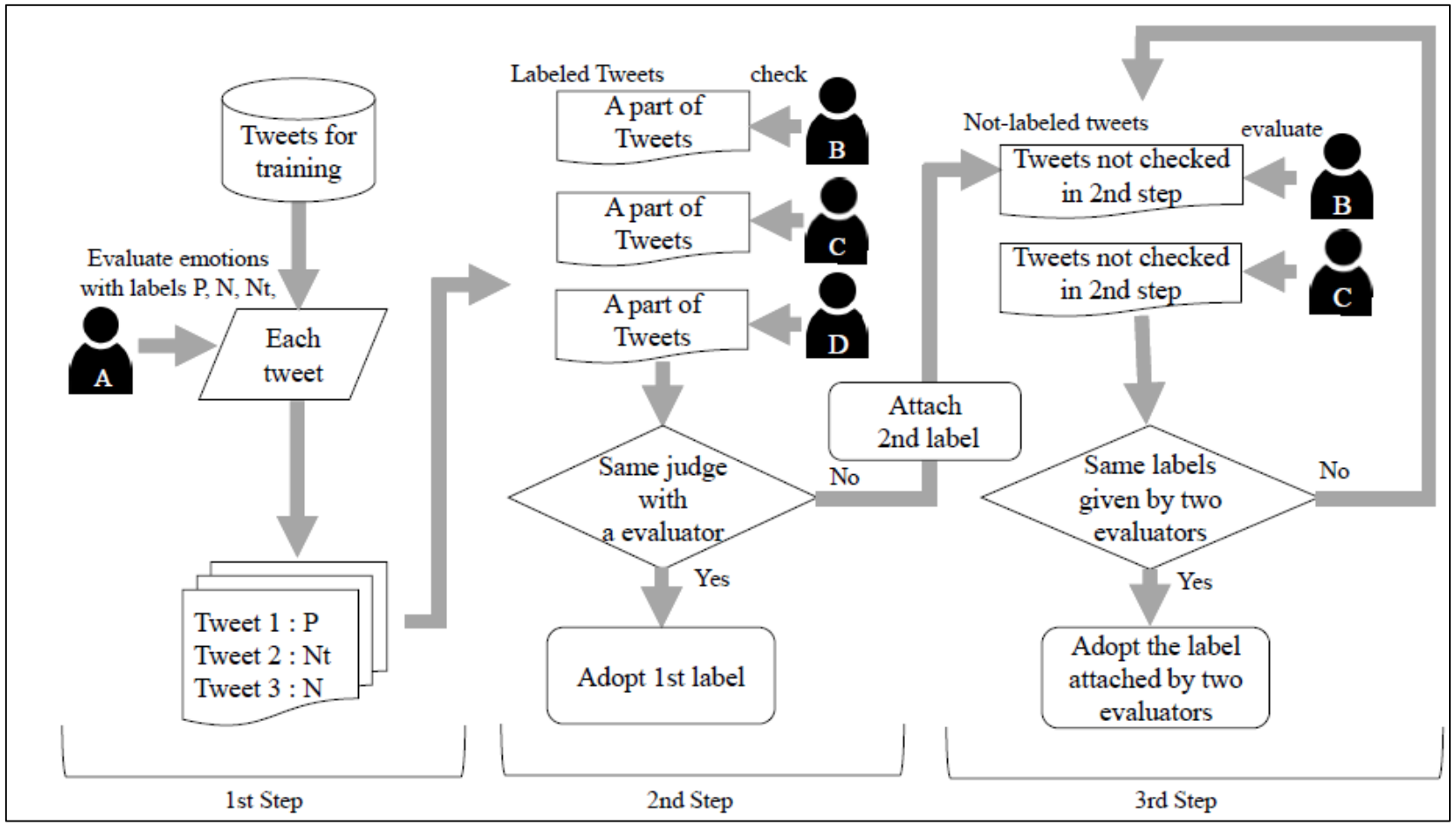
・文章(D)がカテゴリ(C)に属する確率P(C|D)
（文章中の単語群を$W_1, W_2, \cdots W_i$とする）

$$P(D|C) \simeq P(W_1|C)P(W_2|C)\cdots P(W_i|C)$$

・$P(W_i|C)$の計算方法（ラプラススムージング前）

$$P(W_i|C) = \frac{\text{単語 } W_i \text{の数}}{\text{カテゴリ } C \text{ における語彙数}}$$

学習データの作成



分類総数

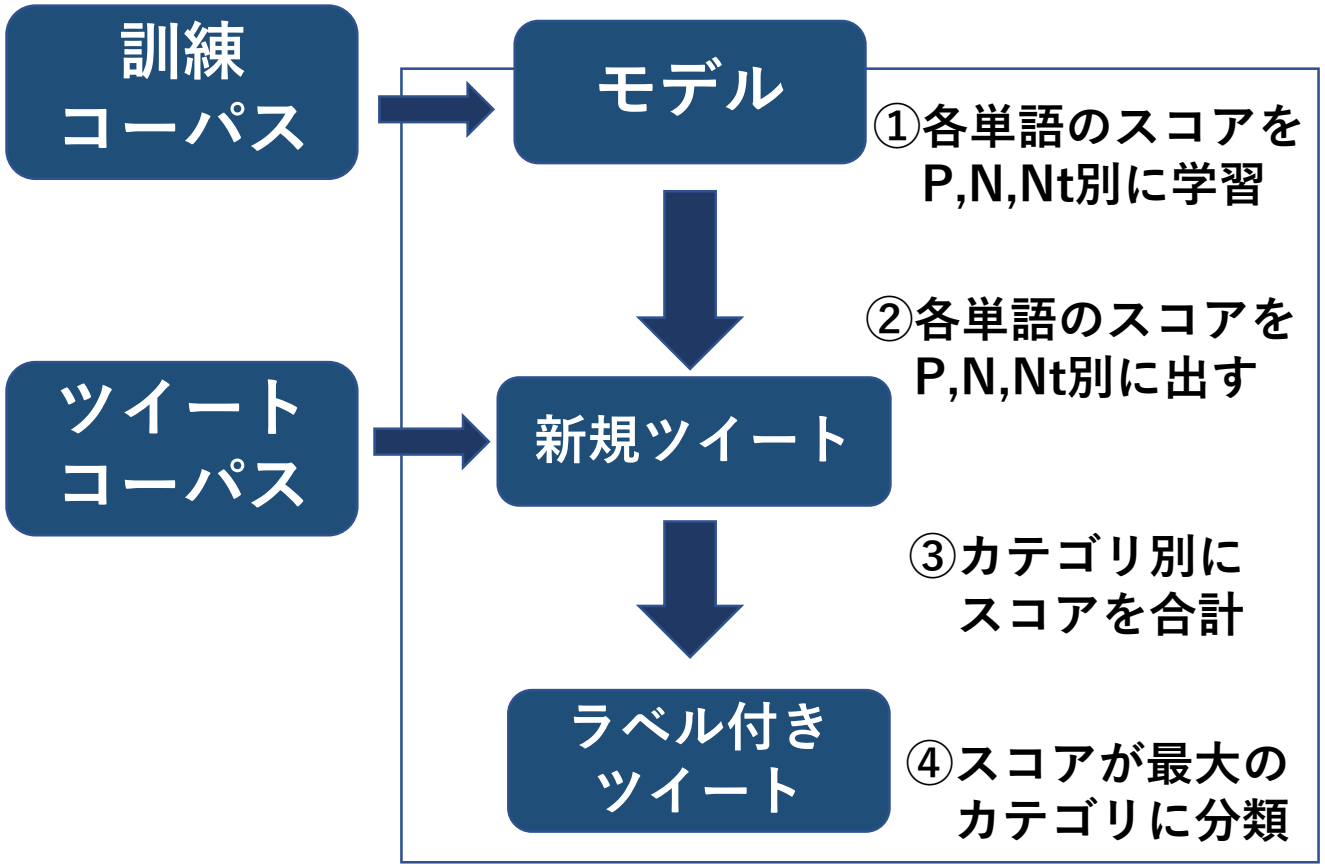
P　: 810
N　: 655
Nt : 39154

Ntを無作為抽出（PとNの平均）

学習データ総数

P　: 810
N　: 655
Nt : 733

分類方法



① 嬉しい ：P=25, N= 0.1, Nt=5
私 ：P=10, N=16, Nt=10

② 私 / は /嬉しい
P 10, 4, 25 ＝ 合計39
N 16, 6, 0.1 ＝ 合計22.1
Nt 10, 8, 5 ＝合計23

③ 私は嬉しい
P=39, N＝22.1, Nt=23

④ 私は嬉しい：P

ナイーブベイズの分類精度

5交差検証（係り受けなし）

		By Naive Bayes Classification			Total	Matching Rate
		P	N	Nt		
By the Evaluator	P	554	85	171	810	68.7%
	N	121	383	151	655	58.5%
	Nt	240	98	395	733	53.9%
	Total	915	566	717	2198	60.6%

5交差検証（係り受けあり）

		By Naive Bayes Classification			Total	Matching Rate
		P	N	Nt		
By the Evaluator	P	540	83	187	810	66.7%
	N	213	261	181	655	39.8%
	Nt	215	106	412	733	56.2%
	Total	968	450	780	2198	55.2%

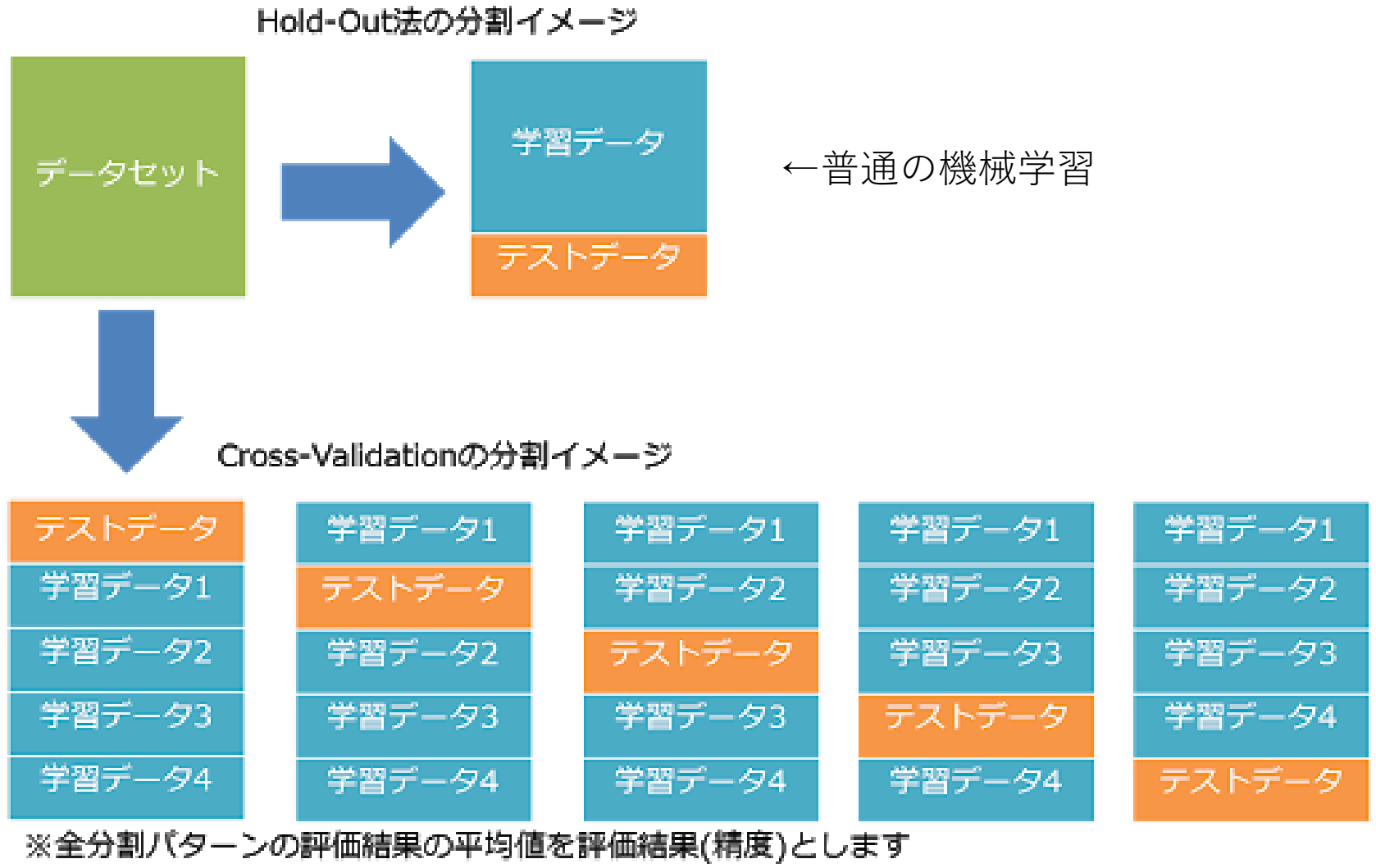
新規150ツイートへの分類（係り受けなし）

		By Naive Bayes Classification			Total	Matching Rate
		P	N	Nt		
By the Evaluator	P	22	1	27	50	44.0%
	N	2	13	35	50	26.0%
	Nt	1	1	48	50	96.0%
	Total	25	15	110	150	55.3%

新規150ツイートへの分類（係り受けあり）

		By Naive Bayes Classification			Total	Matching Rate
		P	N	Nt		
By the Evaluator	P	41	3	6	50	82.0%
	N	12	29	9	50	48.0%
	Nt	13	6	31	50	62.0%
	Total	66	38	46	150	67.3%

5交差検証


Hold-Out法の分割イメージ
データセット
学習データ
テストデータ
←普通の機械学習
Cross-Validationの分割イメージ
テストデータ
学習データ1
学習データ2
学習データ3
学習データ4
学習データ1
テストデータ
学習データ2
学習データ3
学習データ4
学習データ1
学習データ2
テストデータ
学習データ3
学習データ4
学習データ1
学習データ2
学習データ3
テストデータ
学習データ4
学習データ1
学習データ2
学習データ3
学習データ4
テストデータ
←モデルの検証
※全分割パターンの評価結果の平均値を評価結果(精度)とします

係り受け分析

- 意味を強める「とても」「かなり」に対応する単語のスコアを１．５倍に。
- 意味を弱くする「少し」「ちょっと」に対応する単語のスコアを０．５倍に。
- 意味を逆にする「～ない」に対応する単語のスコアを-1倍に。

Yahoo係り受けAPI

ユーザーの感情傾向算出

各単語が持つカテゴリごとのスコアを合計し、ユーザーの感情傾向（CI)を算出

ユーザーが投稿したツイートの各カテゴリ合計スコア

$$TPES = \sum_{i=1}^{n} PES_i$$

ポジティブ

・*n*はツイートの数，PES_iはPカテゴリのスコア

$$TNES = \sum_{i=1}^{n} NES_i$$

ネガティブ

・*n*はツイートの数，NES_iはNカテゴリのスコア

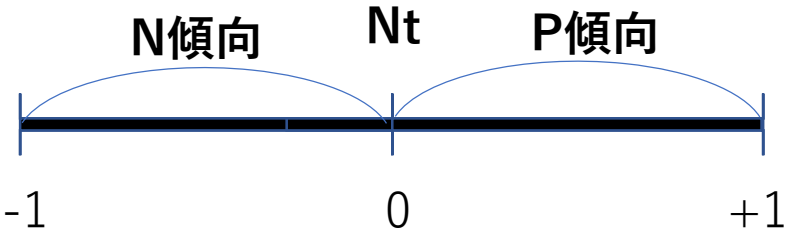
$$TNtES = \sum_{i=1}^{n} NtES_i$$

ニュートラル

・*n*はツイートの数，$NtES_i$はNtカテゴリのスコア

複合的感情傾向指標(Composite Index)

$$CI = \frac{TPES + 0.5TNtES}{TPES + TNES + TNtES}$$



検定結果

- 全ツイート数が10以下のユーザーと，
  感情ツイート数が平均の半分以下のユーザーを除外.
  C/からP/N群を作成してBrunner-Munzel検定を実施した.

	Total Sample (N = 322)			P–Group (N = 81)			N–Group (N = 81)			t–value
	Median	Mean	SD	Median	Mean	SD	Median	Mean	SD	
Followee Fluctuation	2.000	5.886	16.648	5.000	12.840	26.718	1.000	2.520	6.404	-3.707**
Follower Fluctuation	1.000	5.500	17.454	3.000	13.910	31.234	0.000	1.173	4.697	-5.230**
Mutual Follow Fluctuation	0.000	4.0100	14.397	2.000	10.760	26.532	0.000	0.800	3.106	-5.007**
Emotional Tweet Count	72.500	91.690	63.407	80.000	105.200	77.249	73.000	86.010	58.498	$0.924^{n.s.}$
All Tweet Count	142.000	170.200	111.641	126.000	174.900	125.400	154.400	173.600	107.942	$-1.190^{n.s.}$

*p<0.05, **p<0.01, n.s.: not significant

実験におけるt値比較

	感情語辞書A	感情語辞書B	ナイーブベイズ	NB+係り受け
フォロー増減数	-2.055*	-2.022*	-3.218**	-3.707**
フォロワー増減数	-2.418*	-2.566*	-2.826**	-5.230**
相互フォロー増減数	-2.749**	-2.930**	-2.907**	-5.007**
感情ツイート数	-1.432$^{n.s.}$	-1.311$^{n.s.}$	-0.157$^{n.s.}$	0.924$^{n.s.}$
全ツイート数	-1.721$^{n.s.}$	0.528$^{n.s.}$	0.020$^{n.s.}$	-1.190$^{n.s.}$

*p<0.05, **p<0.01, n.s.: not significant

議論と考察

- 感情傾向評価を改善して検証を行い，全実験において
  - ○ フォロー数
  - ○ フォロワー数
  - ○ 相互フォロー数

に有意差が見られた

⇒ Pユーザーは，Nユーザーよりもユーザー関係が増えるだけでなく，相互的にフォローし関係構築を行っている

機械学習を使うときの注意点

- 最終的な予測をするときに使われるだけでなく，何らかの分析の過程におけるツールとして用いられることもある．
- その場合，結果を鵜呑みにして用いるのではなく予備実験や交差検証を行い，どれだけ信頼できるかをテストする．
- どれだけ質の良い（ノイズが少ない）データを大量に集められるかで大きく変わってくる．

レビューシートの提出

- 今日の授業に関するレビューシートを，manabaから提出すること.
- 機械学習で，どのようなことが分類・予測できるかを説明変数と目的変数に分けて提案すること.
  （その他の部分に記載．オプション）

後日不明点があれば，多胡まで.

７号館５階　第９実験室内　第９研究室
tago@net.it-chiba.ac.jp