機械学習
レポート課題

管理工学科
篠沢佳久

レポート課題

- 問題①～④について回答し，レポートとしてまとめなさい．
  - 問題①（線形分離不可能問題）
  - 問題②（単純ベイズ決定則）
  - 問題③（画像のクラス分類）
  - 問題④（マンションの賃貸予測）
  - 最後に講義の感想，要望を書いて下さい

- 問題のデータ，サンプルプログラムはreport.zipにまとめてあります．

*問題①②は表計算でも解けます．問題③④はscikit-learnを用いないと解けません

レポートを書く際の注意

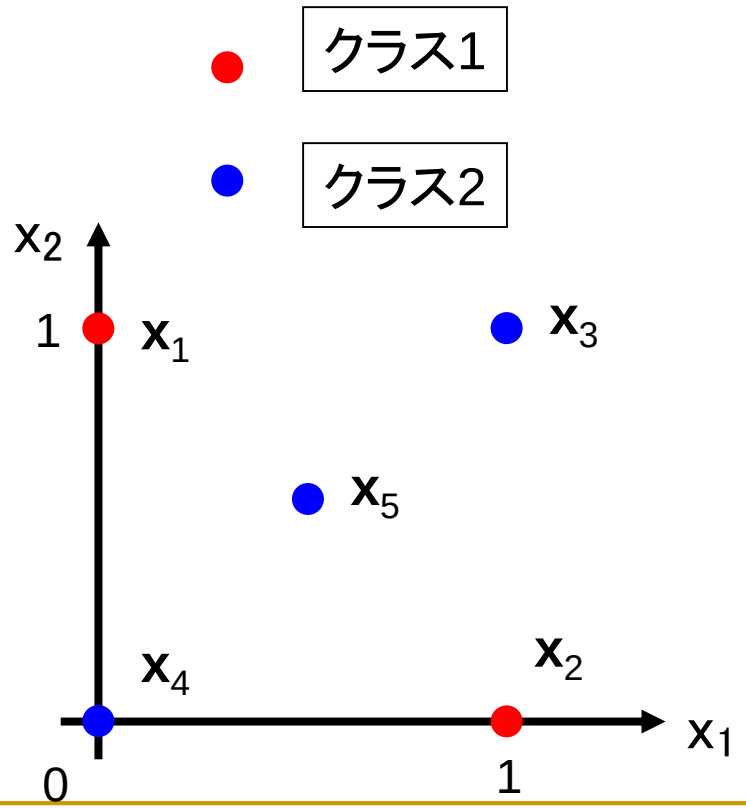
- プログラミングで回答した場合，これまでのプログラミングの課題と同様です
- すなわち，
  - プログラムの説明を簡潔に書く
  - 実行結果を掲載する

- プログラミングの課題でない場合も，どのような過程で求めることができるのか，採点者が理解でき，また再現できるように説明をして下さい

問題①（線形分離不可能問題）

- 下記の線形分離不可能な問題について，クラス1とクラス2を分類できる識別関数を求めなさい

	x_1	x_2	
x_1	0	1	クラス1
x_2	1	0	クラス1
x_3	1	1	クラス2
x_4	0	0	クラス2
x_5	0.5	0.5	クラス2



問題①（線形分離不可能問題）

- 求めた識別関数の数式あるいは条件式を回答して下さい.
  - scikit-learnで解いた場合，識別関数の数式も表示して下さいということです.

- 例えば*,

$$y = \frac{1}{e^{-(0.45x_1 + 0.25x_2 - 0.39)}}$$

$y > 0 \rightarrow$ クラス1

$y < 0 \rightarrow$ クラス2

*本当に例えばで，このような式にはなりません

問題②（単純ベイズ決定則）

- 生成モデルの実習問題（「頭痛」の日の判定）
- 特徴
  - 天気・・・晴れ，曇り，雨
  - 気温・・・暑い，適温，寒い
  - 湿度・・・高い，適当，低い
  - 睡眠・・・寝過ぎ，普通，寝不足 ← 追加
  - 講義・・・yes，no
- 「頭痛」
  - 頭痛はyes，no

問題②(1)(単純ベイズ決定則)

- prob-2-1.xlsx を対象として，最も頭痛が起きる場合の特徴(a_1,a_2,a_3,a_4,a_5)，最も頭痛が起きない場合の特徴(b_1,b_2,b_3,b_4,b_5)を単純ベイズ決定則を用いて予測しなさい.

- 事後確率が最大となる特徴を求めて下さい.
- p(頭痛=yes|天気=a_1，気温=a_2，湿度=a_3，睡眠=a_4，講義=a_5)
- p(頭痛=no|天気=b_1，気温=b_2，湿度=b_3，睡眠=b_4，講義=b_5)

学習データ（25個）

prob-2-1.xlsx

	天気	気温	湿度	睡眠	講義	頭痛
1	晴れ	寒い	低い	寝過ぎ	yes	no
2	曇り	暑い	高い	普通	no	yes
3	晴れ	暑い	低い	寝過ぎ	no	no
4	雨	適温	高い	寝不足	yes	no
5	雨	寒い	高い	普通	no	no
6	晴れ	適温	適当	普通	yes	yes
7	曇り	暑い	低い	寝不足	yes	no
8	雨	寒い	高い	寝過ぎ	no	no
9	曇り	適温	低い	普通	yes	yes
10	曇り	適温	適当	寝不足	no	no
11	晴れ	暑い	適当	普通	yes	yes
12	晴れ	適温	高い	普通	no	no
13	雨	暑い	低い	寝過ぎ	no	no
14	雨	暑い	適当	寝過ぎ	yes	yes
15	曇り	適温	低い	普通	yes	no
16	晴れ	暑い	適当	寝不足	no	yes
17	晴れ	寒い	高い	寝過ぎ	yes	yes
18	曇り	適温	高い	普通	yes	no
19	曇り	適温	適当	普通	no	no
20	雨	寒い	適当	普通	no	yes
21	晴れ	寒い	低い	普通	no	no
22	雨	適温	高い	寝不足	yes	yes
23	曇り	暑い	適当	普通	yes	no
24	晴れ	暑い	高い	寝過ぎ	no	yes
25	晴れ	適温	適当	寝過ぎ	no	yes

問題②(2)(単純ベイズ決定則)

- prob-2-2.xlsx のデータにおいては，データが集まらず条件付き確率 p(気温=寒い|頭痛=yes) の値が0となってしまいます(調べてみて下さい).

- このような場合，事後確率 p(頭痛=yes|天気=晴れ，気温=寒い，湿度=適当，睡眠=寝過ぎ，講義=yes) は0と計算されてしまいます.

- データが集まらず，条件付き確率が0となってしまう場合，どのように事後確率を計算し，ベイズ決定則を行なったらよいか考えなさい.

学習データ(25個)

prob-2-2.xlsx

	天気	気温	湿度	睡眠	講義	頭痛
1	晴れ	寒い	低い	寝過ぎ	yes	no
2	曇り	暑い	高い	普通	no	yes
3	晴れ	暑い	低い	寝過ぎ	no	no
4	雨	適温	高い	寝不足	yes	no
5	雨	寒い	高い	普通	no	no
6	晴れ	適温	適当	普通	yes	yes
7	曇り	暑い	低い	寝不足	yes	no
8	雨	寒い	高い	寝過ぎ	no	no
9	曇り	適温	低い	普通	yes	yes
10	曇り	適温	適当	寝不足	no	no
11	晴れ	暑い	適当	普通	yes	yes
12	晴れ	適温	高い	普通	no	no
13	雨	暑い	低い	寝過ぎ	no	no
14	雨	暑い	適当	寝過ぎ	yes	yes
15	曇り	適温	低い	普通	yes	no
16	晴れ	暑い	適当	寝不足	no	yes
17	晴れ	適温	高い	寝過ぎ	yes	yes
18	曇り	適温	高い	普通	yes	no
19	曇り	適温	適当	普通	no	no
20	雨	暑い	適当	普通	no	yes
21	晴れ	寒い	低い	普通	no	no
22	雨	適温	高い	寝不足	yes	yes
23	曇り	暑い	適当	普通	yes	no
24	晴れ	暑い	高い	寝過ぎ	no	yes
25	晴れ	適温	適当	寝過ぎ	no	yes

問題②(2)(単純ベイズ決定則)

- (補足説明)
- サンプリングを行ない，条件付き確率が0となったのだから，0で計算した方がよいという考えもあります．
- ただし，サンプリングは有限回しか行えません．

- コインをある条件下で5回投げて，表が5回，裏が0回出た場合，裏が出る条件付き確率を0として，事後確率を計算しなければなりません．

- しかし，コインの表裏がでる確率は普通に考えれば0.5です．このような場合，どうしたらよいでしょうか，という問いです．

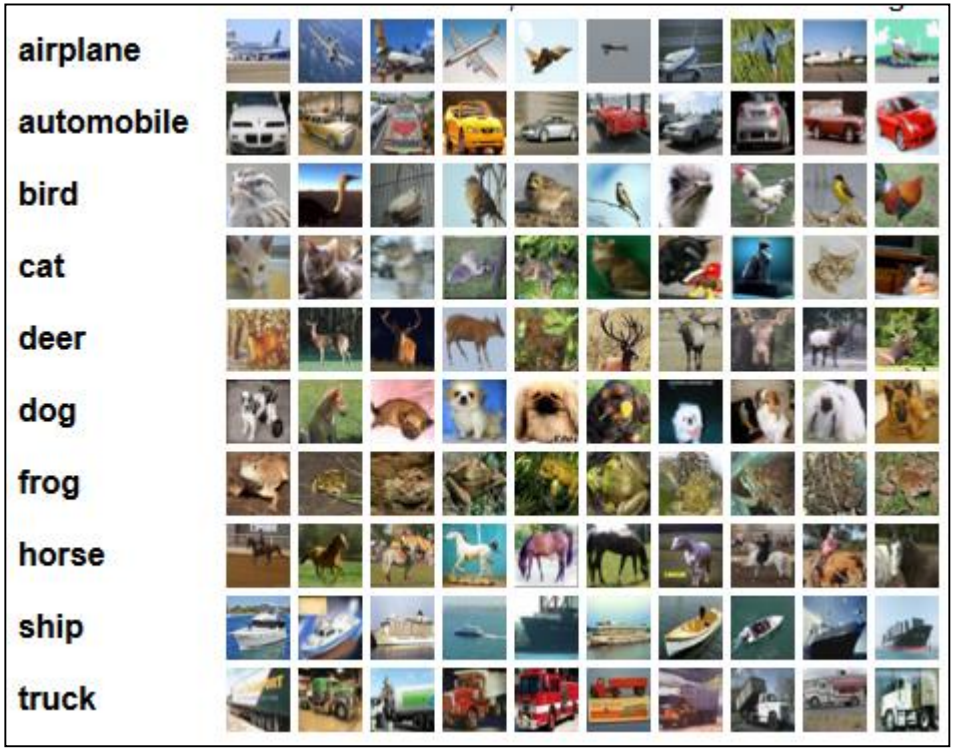
問題③（画像のクラス分類）

- cifar.py
  - cifar-10/train/ を学習データ，cifar-10/test/ をテストデータとして配列に格納
  - train_data　：学習データ
  - train_label：学習データのラベル
  - test_data　：テストデータ
  - test_label　：テストデータのラベル

- cifar-10/train/
  - 学習データ（10クラス×200枚）
- cifar-10/test/
  - テストデータ（10クラス×200枚）

CIFAR-10①

- https://www.cs.toronto.edu/~kriz/cifar.html



クラス数　10クラス
学習データ　50,000枚
テストデータ 10,000枚
画像の大きさ　32×32

CIFAR-10②

- cifar-10/train/ ・・・ 学習データ
- cifar-10/test/ ・・・ テストデータ
- 各クラスごとに200枚
- 学習データ，テストデータとも2,000枚
- 大きさ 32×32（RGB画像），値は0～255

二次（三次?）配布はしないで下さい

train/airplane/

問題③（画像のクラス分類）

- 講義で学んだ手法を用いて，
  - cifar-10/train/以下の画像（2,000枚）を学習
  - cifar-10/train/以下の画像（2,000枚）を認識

  しなさい．

- さまざまな手法（およびそのパラメータ）を用いて Accuracy を求め，最も高い Accuracy の手法を調べて下さい．

- 調べた過程をレポートとしてまとめなさい．

問題④（マンションの賃貸予測）

- マンションの賃貸情報*
  - 学習データ（prob-4-train_data.csv） 500件
  - テストデータ（prob-4-test_data.csv） 500件

物件番号	ID番号（整数値）
路線	カテゴリー変数（整数値）
駅	カテゴリー変数（整数値）
徒歩距離	実数値（メートル）
コンビニ距離	実数値（メートル）
面積	実数値（平方メートル）
部屋数	整数値
築年数	整数値6桁（年，月）
階数	整数値
家賃	整数値（円）
賃借中	賃借中→0 空部屋→1

*実データの分布から生成した人工データです

問題④（マンションの賃貸予測）

- 設定した家賃で物件を借りる人がいるかどうかを予測できるモデルを考えます.
- 学習データ（prob-4-train_data.csv，500件）を対象に講義で学んだ手法を用いて，予測モデルを作成して下さい.
- テストデータ（prob-4-test_data.csv，500件）を対象に予測精度を調べます.

問題④（マンションの賃貸予測）

- さまざまな手法（およびそのパラメータ）により予測モデルを作成し，予測精度を求め，最も高い予測精度の手法を調べて下さい．
- また用いる特徴についても考えて下さい（不必要な特徴もあるかもしれません）．
- 調べた過程をレポートとしてまとめなさい．

提出方法

- 提出期限　1/20（月）13時まで
- 提出方法　keio.jp
- レポート本体，その他のファイルを全て一緒に提出して下さい
  - 配布したデータは提出しなくてよいです

補足説明

cifar.py

cifar.py

```
import sys
import os
import numpy as np
import random
from PIL import Image
import matplotlib.pyplot as plt

# クラス数
class_num = 10

# 画像の大きさ
size = 32
feature = size * size * 3

# データ数
DATA = 200
```

必要なパッケージをインポート

画像処理（読み込み，大きさの変更）に必要

画像を表示するために必要

画像の縦，横の長さ

画像の特徴数（画素数）
縦の長さ×横の長さ×RGB
→ feature×feature×3

```
# 学習データ
train_data = np.zeros( (class_num*DATA,feature) , dtype=np.float32 )
train_label = np.zeros( class_num*DATA, dtype=np.int32 )
```

学習データ train_data
（クラス数×データ数，特徴数）

学習ラベル train_label
（クラス数×データ数）
整数値

```
# テストデータ
test_data = np.zeros( (class_num*DATA,feature) , dtype=np.float32 )
test_label = np.zeros( class_num*DATA, dtype=np.int32 )
```

テストデータ test_data
（クラス数×データ数，特徴数）

テストラベル test_label
（クラス数×データ数）
整数値

```
# フォルダー名
dir = [ "train" , "test" ]
dir1 = [ "airplane" , "automobile" , "bird" , "cat" , "deer" , "dog" , "frog" , "horse" ,
"ship" , "truck" ]
```

学習データ，テストデータのフォルダー名

10クラスのフォルダー名

```
# データの読み込み
def Read_data():
    # 学習データの読み込み
    for i in range(class_num):
        for j in range(0,DATA):
            # RGB画像で読み込み
            file = "cifar-10/" + dir[ 0 ] + "/" + dir1[i] + "/" + str(j) + ".png"
            work_img = Image.open(file).convert('RGB')

            # 大きさの変換
            resize_img = work_img.resize((size, size))

            # 配列に格納
            train_data[ i * DATA + j ] =
              np.asarray(resize_img).astype(np.float64).flatten()
            train_label[ i * DATA + j ] = i
```

画像ファイル名の作成

RGB画像として読み込む

(size,size)に大きさを変更

numpyに変換→ベクトルに変更

学習データのラベル

データの格納

学習データの場合

train_label[0] ～ train_label[199]

- train_data[0] ～ train_data[199] → airplane（ラベルは0）
- train_data[200] ～ train_data[399] → automobile（1）
- train_data[400] ～ train_data[599] → bird（2）
- train_data[600] ～ train_data[799] → cat（3）
- train_data[800] ～ train_data[999] → deer（4）
- train_data[1000] ～ train_data[1099] → dog（5）
- train_data[1200] ～ train_data[1399] → frog（6）
- train_data[1400] ～ train_data[1599] → horse（7）
- train_data[1600] ～ train_data[1799] → ship（8）
- train_data[1800] ～ train_data[1999] → truck（9）

train_label[1800] ～ train_label[1999]

*学習データのラベル（train_label），テストデータ（test_data），テストデータのラベル（test_label）も同様の要素番号に格納されます

```
    # テストデータの読み込み
    for i in range(class_num):
        for j in range(0,DATA):
            # RGB画像で読み込み
            file = "cifar-10/" + dir[ 1 ] + "/" + dir1[i] + "/" + str(j) + ".png"
            work_img = Image.open(file).convert('RGB')

            # 大きさの変換
            resize_img = work_img.resize((size, size))

            # 配列に格納
            test_data[ i * DATA + j ] =
              np.asarray(resize_img).astype(np.float64).flatten()
            test_label[ i * DATA + j ] = i

# データの読み込み
Read_data()
```

画像ファイル名の作成

RGB画像として読み込む

(size,size)に大きさを変更

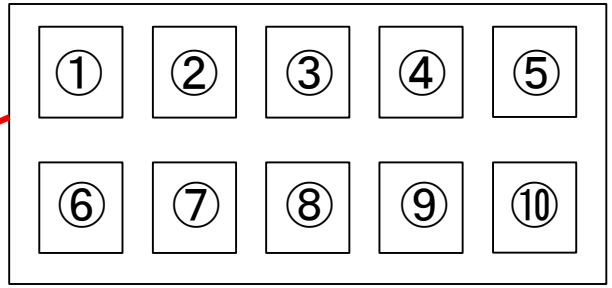
numpyに変換→ベクトルに変更

学習データのラベル

```
# 画像の表示
plt.figure()
rnd = random.randint(0,DATA-1)
count = 1
for i in range(10):
    plt.subplot(2,5,count)
    plt.imshow(np.reshape(train_data[i*DATA+rnd]/255,(size,size,3)))
    plt.xticks(color="None")
    plt.yticks(color="None")
    plt.tick_params(length=0)
    plt.title( dir1[i] )
    count += 1
plt.show()
plt.close()
```

乱数で画像を選択

縦2枚，横5枚で表示



train_data[]：（size×size×3）のベクトル
→ 画像（size，size，3）に変換

画像の表示

cifar.pyの実行

画像の表示