# Introduction to Machine Learning
# Homework 1

151250048, 郭浩滨, 151250048@smail.nju.edu.cn

May 20, 2019

## 1 [20pts] Naive Bayes Classifier

We learned about the naive Bayes classifier using the "property conditional independence hypothesis". Now we have a data set as shown in the following table:

Table 1: Dataset

| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $y$ |
|---|---|---|---|---|---|
| Instance1 | 1 | 1 | 1 | 0 | 1 |
| Instance2 | 1 | 1 | 0 | 0 | 0 |
| Instance3 | 0 | 0 | 1 | 1 | 0 |
| Instance4 | 1 | 0 | 1 | 1 | 1 |
| Instance5 | 0 | 0 | 1 | 1 | 1 |

(1) [10pts] Calculate: $\Pr\{y = 1|\mathbf{x} = (1, 1, 0, 1)\}$ and $\Pr\{y = 0|\mathbf{x} = (1, 1, 0, 1)\}$.

(2) [10pts] After using Laplacian Correction, recalculate the value in the previous question.

(1) 根据贝叶斯定理，有

$$P(y|x) = \frac{P(y)P(x|y)}{P(x)}$$

由题意可知

$$\begin{aligned}P(x=(1,1,0,1)|y=1) &= P(x_1=1|y=1)P(x_2=1|y=1)P(x_3=0|y=1)P(x_4=1|y=1)\\ &= \frac{2}{3}*\frac{1}{3}*\frac{0}{3}*\frac{2}{3}\\ &= 0\end{aligned}$$

$$\begin{aligned}P(x=(1,1,0,1)|y=0) &= P(x_1=1|y=0)P(x_2=1|y=0)P(x_3=0|y=0)P(x_4=1|y=0)\\ &= \frac{1}{2}*\frac{1}{2}*\frac{1}{2}*\frac{1}{2}\\ &= \frac{1}{16}\end{aligned}$$

故

$$\begin{aligned}P(x=(1,1,0,1)) &= P(x=(1,1,0,1)|y=1)P(y=1)+P(x=(1,1,0,1)|y=0)P(y=0)\\ &= \frac{3}{5}*0+\frac{2}{5}*\frac{1}{16}\\ &= \frac{1}{40}\end{aligned}$$

因此

$$\Pr\{y=1|x=(1,1,0,1)\} = \frac{P(y=1)P(x=(1,1,0,1)|y=1)}{P(x=(1,1,0,1))} = \frac{\frac{3}{5}*0}{\frac{1}{40}} = 0$$

$$\Pr\{y=0|x=(1,1,0,1)\} = \frac{P(y=0)P(x=(1,1,0,1)|y=0)}{P(x=(1,1,0,1))} = \frac{\frac{2}{5}*\frac{1}{16}}{\frac{1}{40}} = 1$$

(2) 运用拉普拉斯修正后可得

$$\begin{aligned}P(y=1) &= \frac{3+1}{5+2} = \frac{4}{7}\\ P(y=0) &= \frac{2+1}{5+2} = \frac{3}{7}\\ P(x=(1,1,0,1)|y=1) &= \frac{2+1}{3+2}*\frac{1+1}{3+2}*\frac{0+1}{3+2}*\frac{2+1}{3+2} = \frac{18}{625}\\ P(x=(1,1,0,1)|y=0) &= \frac{1+1}{2+1}*\frac{1+1}{2+1}*\frac{1+1}{2+1}*\frac{1+1}{2+1} = \frac{1}{16}\\ P(x=(1,1,0,1)) &= \frac{4}{7}*\frac{18}{625}+\frac{3}{7}*\frac{1}{16} = \frac{21189}{490000}\end{aligned}$$

故

$$\Pr\{y=1|\mathbf{x}=(1,1,0,1)\} = \frac{\frac{4}{7}*\frac{18}{625}}{\frac{21189}{490000}} \approx 0.38$$

$$\Pr\{y=0|\mathbf{x}=(1,1,0,1)\} = \frac{\frac{3}{7}*\frac{1}{16}}{\frac{21189}{490000}} \approx 0.62$$

## 2 [20pts] Bayes Optimal Classifier

For a binary classification task, when data in the two classes satisfies Gauss distribution and have the same variance, please prove that LDA can produce the bayes optimal classifier.

根据贝叶斯判定准则，我们需要最小化总体风险，即找出贝叶斯最优分类器 $h^*$，满足

$$h^*(x) = \arg\min_{c\in\gamma} R(c|x) = \arg\max_{c\in\gamma} P(c|x) = \arg\max_{c\in\gamma} P(c)P(x|c)$$

由于两类数据均满足高斯分布且协方差相等（设为 $\Sigma$ ），则可得

$$P(x|c) = \frac{1}{\sqrt{2\pi^n\Sigma}} \exp -\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)$$

因此

$$\begin{aligned}
h^*(x) &= \arg\max_{c\in\gamma}\ P(c)P(x|c) \\
&= \arg\max_{c\in\gamma}\ \log P(c)P(x|c) \\
&= \arg\max_{c\in\gamma}\ \log(\frac{1}{\sqrt{2\pi^n\Sigma}} \exp -\frac{1}{2}(x-\mu_c)^T\Sigma^{-1}(x-\mu_c)) + log(P(c)) \\
&= \arg\max_{c\in\gamma}\ -\frac{1}{2}(x-\mu_c)^T\Sigma^{-1}(x-\mu_c) + \log P(c) \\
&= \arg\max_{c\in\gamma}\ x^T\Sigma^{-1}\mu_c - \frac{1}{2}\mu_c^T\Sigma^{-1}\mu_c + \log P(c)
\end{aligned}$$

对于二分类任务，设两类数据的中心分别为 $\mu_1$ 和 $\mu_0$，即可得贝叶斯分类器的决策边界为

$$\begin{aligned}
g(x) &= x^T\Sigma^{-1}\mu_1 - x^T\Sigma^{-1}\mu_0 - \frac{1}{2}(\mu_1^T\Sigma^{-1}\mu_1 - \mu_0^T\Sigma^{-1}\mu_0) + \log P(1) - \log P(0) \\
&= x^T\Sigma^{-1}(\mu_1-\mu_0) - \frac{1}{2}(\mu_1+\mu_0)\Sigma^{-1}(\mu_1-\mu_0) \quad // \text{同先验，故} P(0) = P(1)
\end{aligned}$$

根据线性判别分析的结果可得

$$w = S_w^{-1}(\mu_1-\mu_0) = \frac{1}{2}\Sigma^{-1}(\mu_1-\mu_0)$$

可得两类数据的中心在投影面连线的中心为

$$\frac{1}{2}(\mu_1+\mu_0)^T w = \frac{1}{4}(\mu_1+\mu_0)\Sigma^{-1}(\mu_1-\mu_0)$$

数据在投影面上的投影相对连线中心位置的正负即为决策边界，即

$$\begin{aligned} g(x) &= x^T w - \frac{1}{4}(\mu_1+\mu_0)\Sigma^{-1}(\mu_1-\mu_0) \\ &= \frac{1}{2}x^T\Sigma^{-1}(\mu_1-\mu_0) - \frac{1}{4}(\mu_1+\mu_0)\Sigma^{-1}(\mu_1-\mu_0) \end{aligned}$$

由于我们判断的是决策边界取值的正负，故等价于

$$g(x) = x^T\Sigma^{-1}(\mu_1-\mu_0) - \frac{1}{2}(\mu_1+\mu_0)\Sigma^{-1}(\mu_1-\mu_0)$$

因此二分类任务中，若两类数据满足高斯分布且协方差相同时，线性判别分析产生贝叶斯最优分类器。

# 3 [60pts] Ensemble Methods in Practice

Due to their outstanding performance and robustness, ensemble methods are very popular in machine community. In this experiment we will practice ensemble learning methods based on two classic ideas: Boosting and Bagging.

In this experiment, we use an UCI dataset Adult. You can refer to the link[1] to see the data description and download the dataset.

Adult is an class imbalanced dataset, so we select AUC as the performance measure. You can adopt sklearn to calculate AUC.

(1) [10pts] You need finish the code in Python, and only have two files: AdaBoost.py, RandomForestMain.py. (The training and testing process are implemented in one file for each algorithm.)

(2) [40pts] The is experiment requires to finish the following methods:

[1] http://archive.ics.uci.edu/ml/datasets/Adult

- Implement AdaBoost algorithm according to the Fig(8.3), and adopt decision tree as the base learner (For the base learner, you can import sklearn.)
- Implement Random Forest algorithm. Please give a pseudo-code in the experiment report.
- According to the AdaBoost and random forest, analysis the effect of the number of base learners on the performance. Specifically, given the number of base learners, use 5-fold cross validation to obtain the AUC. The range of the number of base learners is decided by yourself.
- Select the best number of base classifiers for AdaBoost and random forests, and obtain the AUC in the test set.

(3) [10pts] In the experimental report, you need to present the detail experimental process. The experimental report needs to be hierarchical and organized, so that the reader can understand the purpose, process and result of the experiment.

实验报告.

## 3.1 实验目的与简介

本次实验使用 UCI 数据集 Adult 和集成学习方法处理收入区间预测问题，通过 AdaBoost 与 Random Forest 算法来构建集成学习的基本框架，以达到加深对这两种算法的理解与集成学习构建过程的基本认知。

本次实验使用 Python3 语言编写实验代码，并使用 sklearn 框架实现基分类器（决策树）以及数据的预处理和实验结果指标的计算等。

## 3.2 数据加载及预处理

首先我们需要加载数据集并对其进行预处理，通过 pandas 包提供的 read_csv 方法，将数据集加载到 DataFrame 中。其次，由于数据集中的特征是一些表示类别的字符串，如 workclass 中有 Self-emp-not-inc、Local-gov 等，可以通过 sklearn 中的 LabelEncoder 类将它们转换为相应的标签，如 Self-emp-not-inc 对应 0，Local-gov 对应 1。

在表示收入的标签中，我们分别将 >50k 和 <=50k 转换为 1 和 -1，表示正负类。

另外，数据集中有用“?”表示的未知数据，可以简单将所有包含未知特征的数据行都去除（大约 4000 多条数据），也可以包含进来，将其作为一个新的特征类型。经过比较验证之后采用第二种方法。

## 3.3 基分类器构建

对于本次实验的基分类器，即决策树分类器，我们可以用 sklearn 的 DecisionTreeClassifier 进行实现。值得注意的是，在 AdaBoost 中训练基分类器时，我们应将样本权重传入到 DecisionTreeClassifier 的 fit 方法中；而在进行 Random Forest 实验时，由于需要先对节点当前的可选属性随机采样，故在构

造 DecisionTreeClassifier 时应传入 max_features 参数（本次实验我们选择了 sqrt 方法对属性进行随机采样）。

另外一点是决策树的最大深度也会对集成分类器（特别是 Random Forest）的预测性能产生影响，故我们在构造 Random Forest 时考虑将决策树最大深度纳入超参数的分析中。

## 3.4 Adaboost

我按照《机器学习》中图 8.3 实现了一个 AdaBoost 分类器（见 AdaBoost.py 中的 AdaBoostClassifier 类），定义了其训练、预测和性能评估等方法。

**超参数选择**

在 AdaBoost 部分我就基分类器的数量对预测性能的影响进行了分析，经过五折交叉验证，可以得到不同基分类器数量对验证性能（以 AUC 表示）的影响如下：

| 基分类器数量 | 10 | 20 | 40 | 60 | 80 | 100 | 200 |
|---|---|---|---|---|---|---|---|
| AUC | 78.0% | 79.0% | 79.3% | 79.3% | 79.4% | 79.2% | 79.0% |

选择验证阶段性能最好的基分类器数量 80 进行测试，最终的测试结果 AUC 为 78.7%。

## 3.5 Random Forest

在 Random Forest 中我们采用 sklearn 的 DecisionTreeClassifier 实现基分类器，即决策树。然后每次使用 bootstrap 方法获得新的样本，训练出一个基分类器，最后使用多个基分类器的结果进行预测，伪代码如下：

```
# train
hypotheses = []
```

```
for t = 1,2,...,T do
  indices = random_choice([1,2,...N], replace=True)
  X_t = X[indices]
  y_t = y[indices]
  h_t = DecisionTree(max_depth=20, max_features='sqrt')
  h_t.train(X_t, y_t)
  hypotheses.append(t)
# test
y_pred = (0, 0, ..., 0)
for h in hypotheses:
  y_pred += h.predict(X_test)
y_pred = np.sign(y_pred)
y_pred[y_pred==0] = random_choice([-1, 1])
```

**超参数选择**

在这个部分我们选择基分类器数量和基分类器决策树的最大深度作为超参数，通过五折交叉验证研究它们对验证性能的影响。

控制决策树的最大深度为 20，可以得到基分类器数量与 Random Forest 的预测性能的关系：

| 基分类器数量 | 10 | 20 | 50 | 60 | 100 | 200 | 300 |
|---|---|---|---|---|---|---|---|
| AUC | 77.2% | 77.8% | 78.2% | 78.0% | 77.9% | 78.1% | 78.0% |

选择基分类器数量为 50，得到决策树最大深度对预测性能的影响：

| 决策树最大深度 | 10 | 15 | 20 | 25 | 30 |
|---|---|---|---|---|---|
| AUC | 75.0% | 77.6% | 78.2% | 77.8% | 77.6% |

最终选择基分类器数量为 50，决策树最大深度为 20 进行测试，得到测试阶段 AUC 为 77.6%。