

## 摘要

本工程分为读取数据、数据预处理、参数的选择与模型的训练，性能测试四部分。

读取数据对应于” processCsvfile.h”与” processCsvfile.c”文件，主要包括数据读取，存放，数据大小等的统计以及对错误的处理。

数据的预处理主要包括对文本属性种类的统计，文本数据的编码、数字数据的归一化、得到特征向量四个步骤

参数选择与模型的训练主要包括随机数的生成、分层抽样、交叉验证选择最佳参数，模型的训练步骤

最后是测试，得到在测试集上的准确率为 76.37%。

### 一、读取数据

此步骤对应于” processCsvfile.h”与” processCsvfile.c”文件，主要包括数据读取，存放，数据大小等的统计以及对错误的处理。

- ❖ 创建结构体变量 attributes（文本型数据定义为 char 型，数字型数字定义为 double 型）将读进来的数据按属性保存从文件读进来的数据，读取之前我们假设属性已知，结构体里面的数组为静态数组，因此数据最大函数为 33000（#define MAX\_DATA\_SIZE 33000），属性的最大长度为 31 个字符（#define MAX\_char\_SIZE 32）；
- ❖ 由于 csv 文件以逗号‘,’作为分隔符，因此可按逗号来分割一行的数据，实现数据按属性存放；
- ❖ 数据的行数和列数会保存在 trainNumRow、trainNumCol，我们的处理过程仅用到了行数；
- ❖ 同时为了便于查看结果的正确性我们阿勇 writeCsvData(char\* csvFilePath) 将我们读取的数据写入另一个文本中；
- ❖ 对于含有“？”的数据，我们将其删除。



共有数据32561个，其中出错的数据有 2399个

图一、运行结果

从运行结果来看，census.csv 文件共有 32561 行数据，其中出错的有 2399 个，

### 二、数据预处理

数据的预处理主要包括对文本属性种类的统计，文本数据的编码、数字数据的归一化、得到特征向量四个步骤

- ❖ 统计文本每个属性里的种类数量存放在 trainAttriTextData 结构体中，此结构体里的成员变量是静态分配内存。范围类似于前面所说的结构体变量 attributes；
- ❖ 为了后续编码方便，我们还会统计一下各个属性中各类别收入大于 50K 的比

率，存放在 trainover50Rate 结构体中。此结构体是静态的，最大存储类似于前面所说的结构体变量 attributes;

```
workclass有7种，分别是：
State-gov          label over50k的占 0.0114
Self-emp-not-inc   label over50k的占 0.0237
Private            label over50k的占 0.1617
Federal-gov        label over50k的占 0.0121
Local-gov          label over50k的占 0.0202
Self-emp-inc       label over50k的占 0.0199
Without-pay        label over50k的占 0.0000

education有16种，分别是：
Bachelors          label over50k的占 0.0705
HS-grad            label over50k的占 0.0536
11th               label over50k的占 0.0020
Masters            label over50k的占 0.0304
9th                label over50k的占 0.0008
Some-college       label over50k的占 0.0443
Assoc-acdm         label over50k的占 0.0085
7th-8th            label over50k的占 0.0012
Doctorate          label over50k的占 0.0093
Assoc-voc          label over50k的占 0.0114
Prof-school        label over50k的占 0.0135
5th-6th            label over50k的占 0.0004
10th               label over50k的占 0.0020
Preschool          label over50k的占 0.0000
12th               label over50k的占 0.0010
1st-4th            label over50k的占 0.0002

maritalStatus有7种，
occupation有14种
relationship有6种
race有5种
race有2种
nativeCountry有41种
```

图 2.1 属性描述

可以看到 workclass、education 等属性的种类个数，以及每个种类中收入大于 50K 的比率。

- ❖ 对文本数据进行 one-hot encode 和赫夫曼-onehot 编码，要求文本数据编码后的长度不超过 3 维，只有 sex 属性用 one-hot 编码(male 为[1 0], female 是[0 1])，其余均用赫夫曼-onehot 编码，编码结果保存在 trainDataEncode 结构体中，此结构体里的成员变量是静态分配内存。范围类似于前面所说的结构体变量 attributes;
- ❖ 所谓赫夫曼-onehot 编码，是我们借助霍夫曼给权重大的字符短的编码的特性，将编码长度差不多，也就是收入超过 50k 比例大致相同的数据归为一类，共分为三类，编码为[1 0 0]、[0 1 0]、[0 0 1]中一种。（具体实现时霍夫曼编码采用的是树形结构，因此定义了霍夫曼树结构 weight 结点的权重，parent,lchild,rchild 分别是父母结点和左右孩子节点）

```
sex的one-hot编码为: (以前八个为例)
Male: 1 0
Male: 1 0
Male: 1 0
Male: 1 0
Female: 0 1
Female: 0 1
Female: 0 1
Male: 1 0

workclass的等长霍夫曼-onehot编码: (以前8个为例)
0 0 1
0 1 0
1 0 0
1 0 0
1 0 0
1 0 0
1 0 0
1 0 0
0 1 0
```

图 2.2 编码举例

只有 sex 属性用 one-hot 编码，其余均用赫夫曼-onehot 编码，编码结果如上图

- ❖ 编码完成后还需保存文本与数字编码之间的对应转换，以便后续再 test 集上使用，使用结构体 Text2Num 保存，此结构体里的成员变量是静态分配内存。范围类似于前面所说的结构体变量 attributes;
- ❖ 对数字数据进行归一化处理，求出最大最小值，用公式  $x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$  使其取值区间在 [0, 1];

```
workclass的编码对应关系为:
State-gov: 0 0 1
Self-emp-not-inc: 0 1 0
Private: 1 0 0
Federal-gov: 0 0 1
Local-gov: 0 1 0
Self-emp-inc: 0 1 0
Without-pay: 0 0 1

race的编码对应关系为:
White: 1 0 0
Black: 0 1 0
Asian-Pac-Islander: 0 1 0
Amer-Indian-Eskimo: 0 0 1
Other: 0 0 1

对age进行归一化后的数据为: (以前八个为例)
0.3014 0.4521 0.2877 0.4932 0.1507 0.2740 0.4384 0.4795
```

图 2.3 文本编码与数字的归一化

可以看到属性的各个种类的编码以及数字归一化后结果。

- ❖ 得到的所有向量按表头所给属性顺序合并得到 28 维的特征向量;

```
得到的特征向量(28维)为: (以前3个为例)
0.30 0.00 0.00 1.00 0.00 1.00 0.00 0.80 0.00 1.00 0.00 0.00 0.00 1.00 0.00 1.00
0.00 1.00 0.00 0.00 1.00 0.00 0.02 0.00 0.40 1.00 0.00 0.00
0.45 0.00 1.00 0.00 0.00 1.00 0.00 0.80 1.00 0.00 0.00 0.00 1.00 0.00 1.00 0.00
0.00 1.00 0.00 0.00 1.00 0.00 0.00 0.00 0.12 1.00 0.00 0.00
0.29 1.00 0.00 0.00 0.00 1.00 0.00 0.53 0.00 1.00 0.00 0.00 0.00 1.00 0.00 1.00
0.00 1.00 0.00 0.00 1.00 0.00 0.00 0.00 0.40 1.00 0.00 0.00
```

图 2.4 得到的特征向量

### 三、参数选择与模型的训练

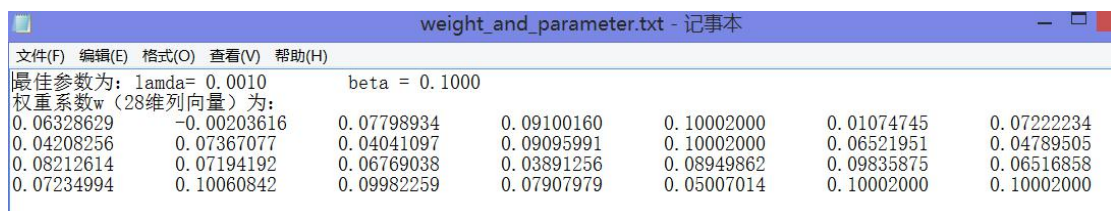
参数选择与模型的训练主要包括随机数的生成、分层抽样、交叉验证选择最佳参数，模型的训练步骤

- ❖ 用洗牌算法生成范围为 $[0, n]$ 的不重复随机数，分别从正负样本中抽取 70% 的数据作为训练集，剩余的 30% 为验证集，以此来分层抽样。)所谓洗牌算法，类似于洗牌的过程，例如要生成 $[0, n-1]$   $n$  个不重复的随机数，我们可以先顺序生成 $[0, 1, 2, 3, 4, 5, \dots, n-1]$ ，然后随机抽取一张和第一张交换，在抽取一张和第 2 张交换，如此反复进行多次，便可以生成范围为 $[0, n]$ 的不重复随机数。)
- ❖ 为保证分层抽象后的比例，我们将数据集分为正负两个集合，存放在 `postivefeatureVector` 和 `negativefeatureVector` 中
- ❖ 选择不同的参数在训练集上训练模型，线性分类器，损失函数，以及  $w$  的更新方式如文档 `assignment.pdf` 里的第三部分所示。
- ❖ 选择最佳参数，将全部的数据用于模型的训练，用  
“`weight_and_parameter.txt`”。保存最优参数与训练出来的权重

```
最佳参数为: lamda= 0.0010      beta = 0.1000
正确率为 0.6786
权重系数w, 最佳参数β和λ已保存至weight_and_parameter.txt文档中
```

图 3.1 最佳参数与模型的训练

如图给出了最佳参数和在验证集上，并将最佳的参数和权重系数保存在 `weight_and_parameter.txt` 文档中。



```
weight_and_parameter.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
最佳参数为: lamda= 0.0010      beta = 0.1000
权重系数w (28维列向量) 为:
0.06328629      -0.00203616      0.07798934      0.09100160      0.10002000      0.01074745      0.07222234
0.04208256      0.07367077      0.04041097      0.09095991      0.10002000      0.06521951      0.04789505
0.08212614      0.07194192      0.06769038      0.03891256      0.08949862      0.09835875      0.06516858
0.07234994      0.10060842      0.09982259      0.07907979      0.05007014      0.10002000      0.10002000
```

图 3.2 `weight_and_parameter.txt` 文档

### 四、测试

如前所述和训练集一样的处理过程，因为可以调用之前的函数，

用 `ReadCsvData(fpTest1)` 来读取 csv 数据;用 `getAttriNum()` 得到各个属性的种类以及种类数;用 `encodeText2num()` 得到文本与数字编码之间的编码转换;用 `normalization()` 对数据进行归一化;用 `getfeatureVector()` 得到特征向量;  
`accuracy = testSetAccuracy()` 计算在测试集上的准确率用来计算测试集的计算准确率输出结果为:

在测试集上的准确率为: 0.76374502