机器学习第一次作业报告

1501214415 位冰镇
1501210408 朱铭健
1300010680 朱垣金

2016 年 10 月 29 日

1. 项目介绍

1) . 所选用的机器学习算法:

a 决策树

b 支持向量机

2) . 数据集的选取:

UCI 网站数据集: Car Evaluation

链接: http://archive.ics.uci.edu/ml/datasets/Car+Evaluation

2. 数据预处理

我们小组经过筛选比较, 选定了 Car Evaluation 作为数据集. 该数据集数据量适中, 特征数目和特征值类型, 类别都比较多. 适合进行机器学习的训练和测试. 数据的具体特征如下:

- 类别 (4 类):

  unacc、acc、good、vgood

- 特征 (5 种) :

  buying: vhigh、high、med、low

  maint: vhigh、high、med、low

  doors: 2、3、4、5more

  person: 2、4

  lugboot: small、med、big

  safety: low、med、high

可见各维特征均为离散型变量, 不过取值类型不统一, 需要对数据进行预处理. 我们通过代码分析了原始数据类别的分布情况

```
数据类别分布情况如下:
类别        数目        百分比%
------------------------------
unacc       1210        70.023
acc         384         22.222
good        69          3.993
vgood       65          3.762
总计        1728        1.000

Process finished with exit code 0
```

为了方便起见，我们对数据进行了数值化. 具体来讲，就是将每一个特征非数值的分量赋予一个合理的数值，以便进一步的处理。当然，赋予什么值也在我们的考虑之中，考虑到决策树对于数值的绝对值并不敏感 (因为对于决策数来讲, 特征之间的距离并不会影响预测的效果)，所以我们暂时对特征属性由低到高赋予 1 2 3 4... 这样的数值。而对于 svm 来说，特征的数值会很大程度影响学习的效果，但是我们依然可以通过对变量权重的改变来控制着一点，所以这么赋值也是可以接受的，关于这一点，我们将在下文讨论。

3. 算法实现

3.1 决策树模型

3.1.1 问题分析与程序实现

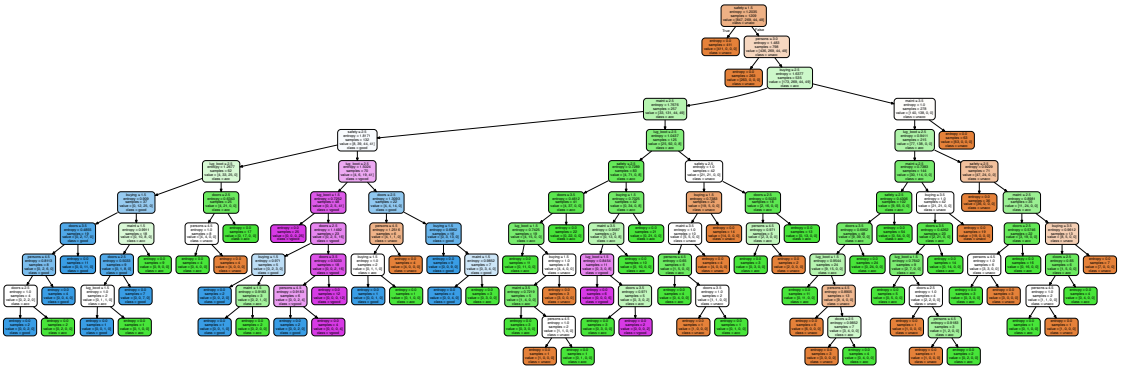
调用 python 中的 package sklearn.tree 中的函数来帮助完成决策树模型的实现。$DecisionTreeClassifier()$ 函数可以创建一个决策树分类器实例 clf_tree, 之后便可以调用方法 $clf.fit()$、$clf.score()$ 对数据进行训练和检验。这里，我们是随机选取了 1000 个样本进行训练，余下 728 个样本进行验证。多次随机抽样、训练并测试的结果如下:

```
决策树开始训练!
决策树训练结束!

开始测试!
测试结束!

tree时间20.470619201660156 准确率0.9672447013487476
决策树开始训练!
决策树训练结束!

开始测试!
测试结束!

tree时间19.240379333496094 准确率0.9672447013487476
决策树开始训练!
决策树训练结束!

开始测试!
测试结束!

tree时间22.480487823486328 准确率0.9672447013487476
```

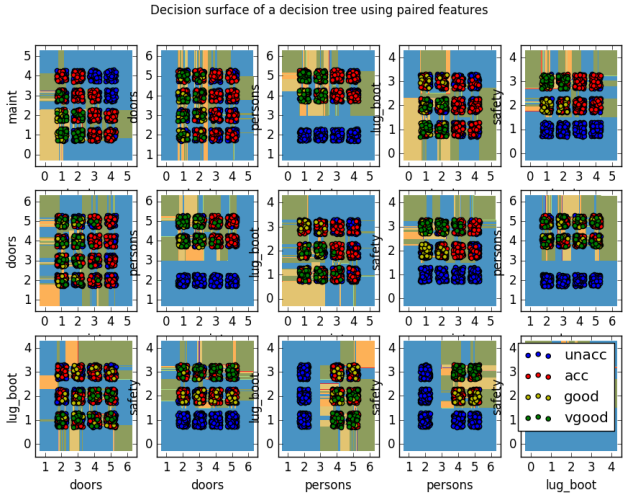
我们发现每一次分类器的准确率都高达 0.95 这说明决策树模型能很好地对此数据及进行分类。

3.1.2 数据可视化

我们画出了决策树的可视图 (因为空间有限，更细致的观察请查看文件 car.pdf)



为了更加直观的对分类效果进行可视化，我们想到将数据投影到 2 维，观测分类效果。



因为是数据在二维上的投影，所以很明显会损失一部分的准确率:

```
0:buying   & maint     准确率    0.628131021194605
1:buying   & doors     准确率    0.4913294797687861
2:buying   & persons   准确率    0.628131021194605
3:buying   & lug_boot  准确率    0.5009633911368016
4:buying   & safety    准确率    0.6705202312138728
5:maint    & doors     准确率    0.371868978805395
6:maint    & persons   准确率    0.5703275529865125
7:maint    & lug_boot  准确率    0.5645472061657033
8:maint    & safety    准确率    0.6551059730250481
9:doors    & persons   准确率    0.5645472061657033
10:doors   & lug_boot  准确率    0.5414258188824663
11:doors   & safety    准确率    0.6184971098265896
12:persons & lug_boot  准确率    0.6242774566473989
13:persons & safety    准确率    0.720616570327553
14:lug_boot & safety   准确率    0.5934489402697495
```

3.2 svm 模型

3.2.1 程序实现

调用 sklearn.svm 包中的 SVC 函数。SVC() 可以创建一个分类器实例，紧接着可以调用 fit、score 方法进行训练和检验，我们随机选取 1000 个样本进行训练，余下的 728 个样本进行测试。不断改变训练时对于不同特征的权重，可以得分类器的分类结果。

```
SVM开始训练!
SVM训练结束!

开始测试!
测试结束!

svm 时间486.1021041870117 准确率0.9595375722543352
```

为了比较分类的性能，我们可以改变 SVC() 的参数。首先考虑到 SVM 模型的优化问题:

$$\min_{\xi,\omega,b}\left\{\frac{1}{2}\omega^T\omega + C\sum_{i=1}^{m}\xi_i\right\}$$

$$st.y_i(\omega^T\phi(x_i)+b) \geq 1-\xi_i, \xi_i \geq 0$$

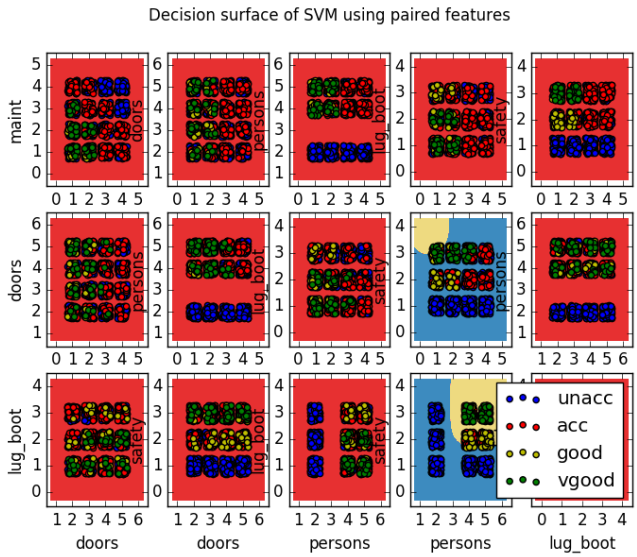
我们可以修改核函数 $\phi(x_i)$ 以及惩罚项系数 C, 来做出不同的分类器。

C	Kernel	runtime(ms)	accuracy	degree
1	rbf	273	0.96	
3	rbf	274	0.98	
10	rbf	370	0.98	
100	rbf	276	0.99	
3	linear	875	0.7938	
10	linear	875	0.79	
3	poly	18205	0.92	3
3	poly	溢出	-	5

通过改变核函数 $\phi(x_i)$ 和 C, 做出了不同的分类器。发现最影响分类效果的应该是核函数 $\phi(x_i)$, 而这之中效果最好的是高斯核函数，而线性核函数表现的效果是最差的。而采用多项式核函数的时候，时间复杂度有了明显的提高，甚至到了 $degree = 5$ 的情况下溢出了时间上限制，在准确率上只有微小的提升。考虑到我组选择的数据集比较密集，试想如果处理相对分散的数据集，多项式核函数处理下可能会引发过拟合的情况。而改变惩罚项系数 C，确实会对分

类的效果有所提升，但是依然不是太明显，这不排除数据集比较特殊的因素，并不能作为普遍结论。而且 C 的值太大是有一定风险的，因为倘若存在一两个离群点，可能会使得这两个点的影响力度加大从而产生过拟合。

3.2.2 数据可视化



相应的结果是：

```
 0:buying   & maint     准确率    0.6994219653179191
 1:buying   & doors     准确率    0.6994219653179191
 2:buying   & persons   准确率    0.6994219653179191
 3:buying   & lug_boot  准确率    0.6994219653179191
 4:buying   & safety    准确率    0.6994219653179191
 5:maint    & doors     准确率    0.6994219653179191
 6:maint    & persons   准确率    0.6994219653179191
 7:maint    & lug_boot  准确率    0.6994219653179191
 8:maint    & safety    准确率    0.6994219653179191
 9:doors    & persons   准确率    0.6994219653179191
10:doors    & lug_boot  准确率    0.6994219653179191
11:doors    & safety    准确率    0.6994219653179191
12:persons  & lug_boot  准确率    0.6994219653179191
13:persons  & safety    准确率    0.7726396917148363
14:lug_boot & safety    准确率    0.6994219653179191
```

4. 结果分析

在我们选定的数据集 car evaluation 上，我们通过比较发现决策树模型的分类效果从时间、准确率综合考虑上，要优于 svm 模型。我们认为这是因为，car evluations 数据集的每一个特征下的属性种类比较少 (最多有 5 个)，而这种相对集中而规整的数据使用决策树是有优势的。

同时，结合对数据结构的观察，我们发现数据测采集是比较不平衡的 (样本中 unacc 类别的样本占据总体的 70%), 这是过抽样的，也可能对训练的效果产生影响。SVM 对于多分类问题需要转化为多个二分类问题，采用一对多或者一对一的分类方法，而决策树模型天然支持多分类问题。