第五周作业

1 ESL第310页第2段“The Gini index can be interpreted in two interesting ways...”
请用具体的计算推导过程完善这段话里对基尼指数意义的解释

答：

文中介绍了两个对基尼指数的解释：

（1）如果分类的时候按照每个分类的概率来随机分配，来计算分类错误率：

假设有K类，其中k类对应的概率为 p_{mk} ，那么，错误率为 $1-p_{mk}$ ，那么总体错误率为正好是基尼指数。

$$\sum_{k=1}^{K} p_{mk}(1-p_{mk})$$

我觉得可以解释为：如果按照每个分类的概率来随机分类，哪个特征的分类错误率越低，优先使用哪个特征进行分类。

（2）假设某个样本被分为第k类，和不分为第k类，是一个0-1分布，其中的概率参数就是 p_{mk} ，这个分布的方差就是 $p_{mk}(1-p_{mk})$ 。方差阐述了一个分布的离散程度，方差越小，说明这个分布的值越集中。那么对于K个分类，可以将每个分类对应的分布的方差计算求和，即为基尼指数。每个分类对应的分布方差和越小，对应的分布都相对更集中。
可以解释为：如果将每个分类假设为一个0-1分布，那么哪个特征的每个分类更集中（可能这样更容易分类），就先使用该特征进行分类。

2 这里有gcForest的“官方实现”
https://github.com/kingfengji/gcForest
请部署有关代码并跑通一个demo，抓图实验过程

下载代码：

(anaconda2-4.1.1) localhost:GitHub$ git clone git@github.com:kingfengji/gcForest.git

安装一些包：

(anaconda2-4.1.1) localhost:gcForest$ pip install joblib
(anaconda2-4.1.1) localhost:gcForest$ pip install keras

(anaconda2-4.1.1) localhost:gcForest$ pip install --upgrade sklearn
(anaconda2-4.1.1) localhost:gcForest$ pip install tensorflow

测试

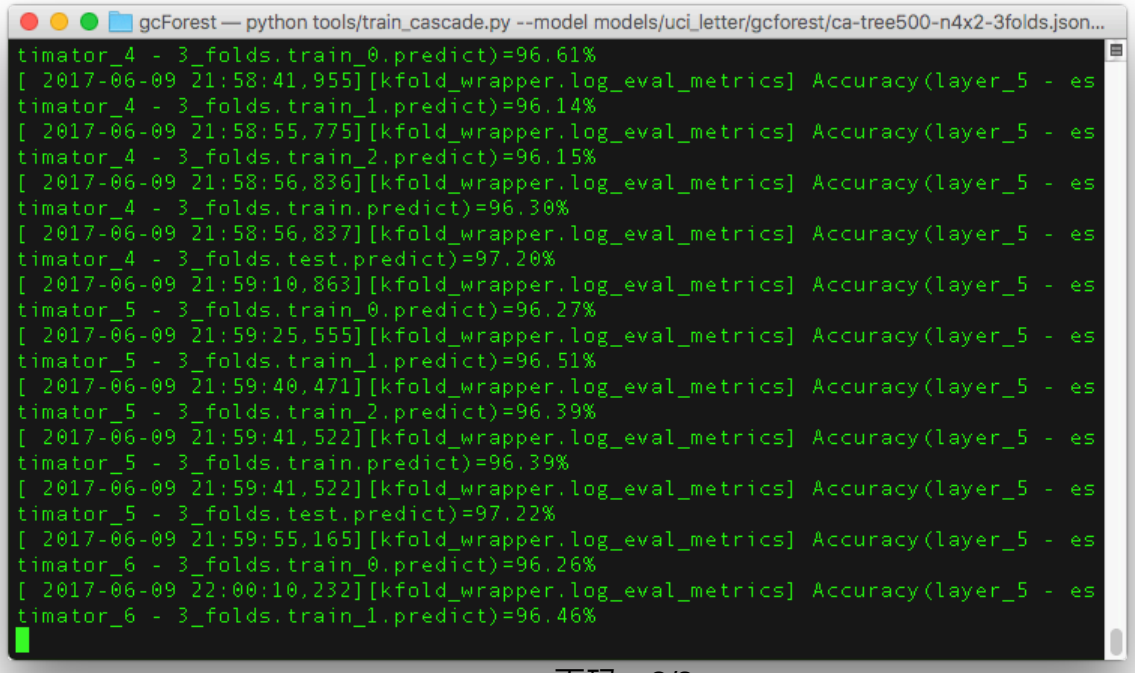
(anaconda2-4.1.1) localhost:gcForest$ cd datasets/uci_letter/
(anaconda2-4.1.1) localhost:uci_letter$ ls
get_data.sh
(anaconda2-4.1.1) localhost:uci_letter$ sh get_data.sh
--2017-06-09 21:15:58-- http://archive.ics.uci.edu/ml/machine-learning-databases/letter-recognition/letter-recognition.data
Resolving archive.ics.uci.edu... 128.195.10.249
Connecting to archive.ics.uci.edul128.195.10.249l:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 712565 (696K) [text/plain]
Saving to: ‘letter-recognition.data’

letter-recognition. 100%[==================>] 695.86K 121KB/s in 7.2s

2017-06-09 21:16:07 (97.1 KB/s) - ‘letter-recognition.data’ saved [712565/712565]

(anaconda2-4.1.1) localhost:uci_letter$ cd ../../

(anaconda2-4.1.1) localhost:gcForest$ python tools/train_cascade.py --model models/uci_letter/gcforest/ca-tree500-n4x2-3folds.json --log_dir logs/gcforest/uci_letter/ca

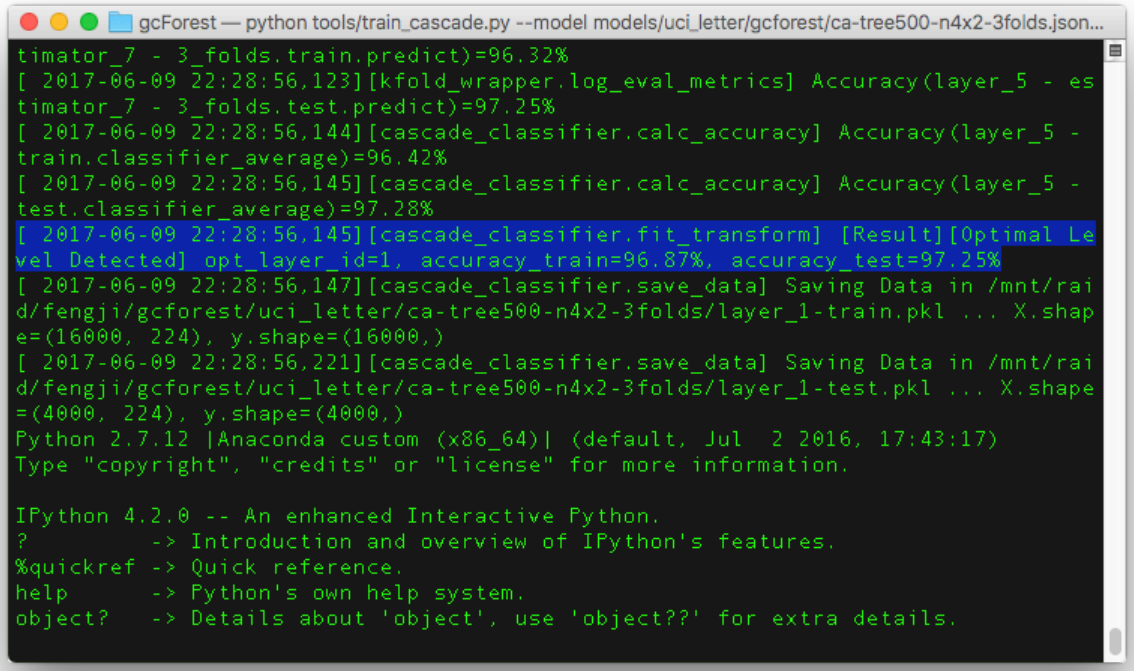


还报了个错：

```
OSError: [Errno 13] Permission denied: '/mnt/raid'
```

修改了下mnt的权限，直接777权限；

重新执行，又是漫长的等待。。。。



[2017-06-09 22:28:56,145][cascade_classifier.fit_transform] [Result][Optimal Level Detected] opt_layer_id=1, accuracy_train=96.87%, accuracy_test=97.25%