

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.edu|128.195.10.249|:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 712565 (696K) [text/plain]
```

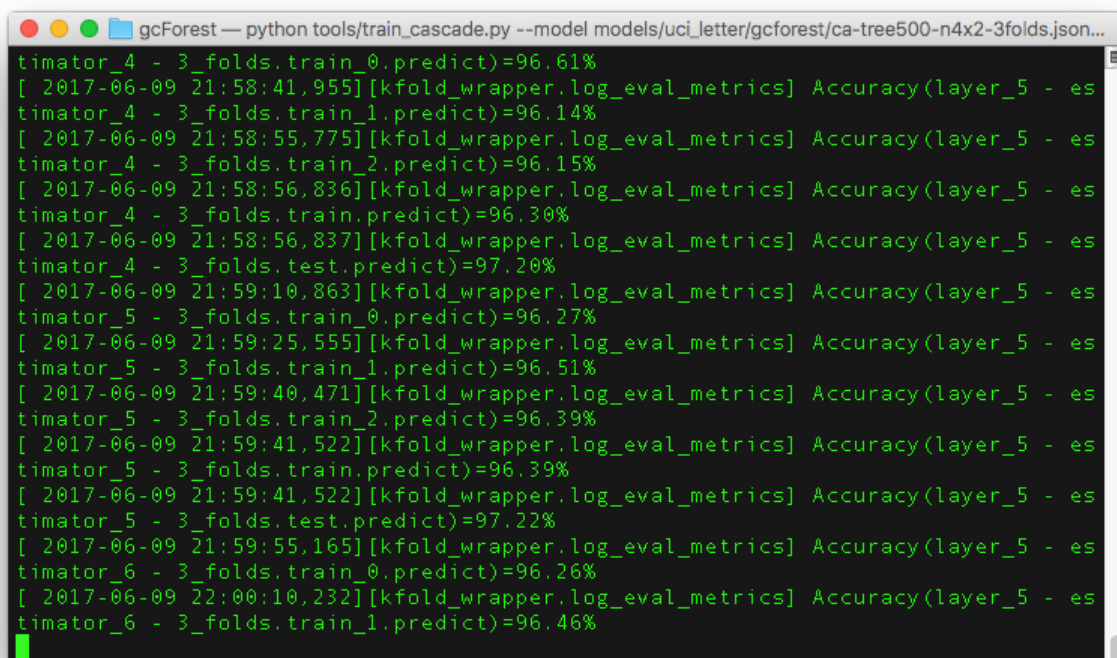
```
Saving to: 'letter-recognition.data'
```

```
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
```



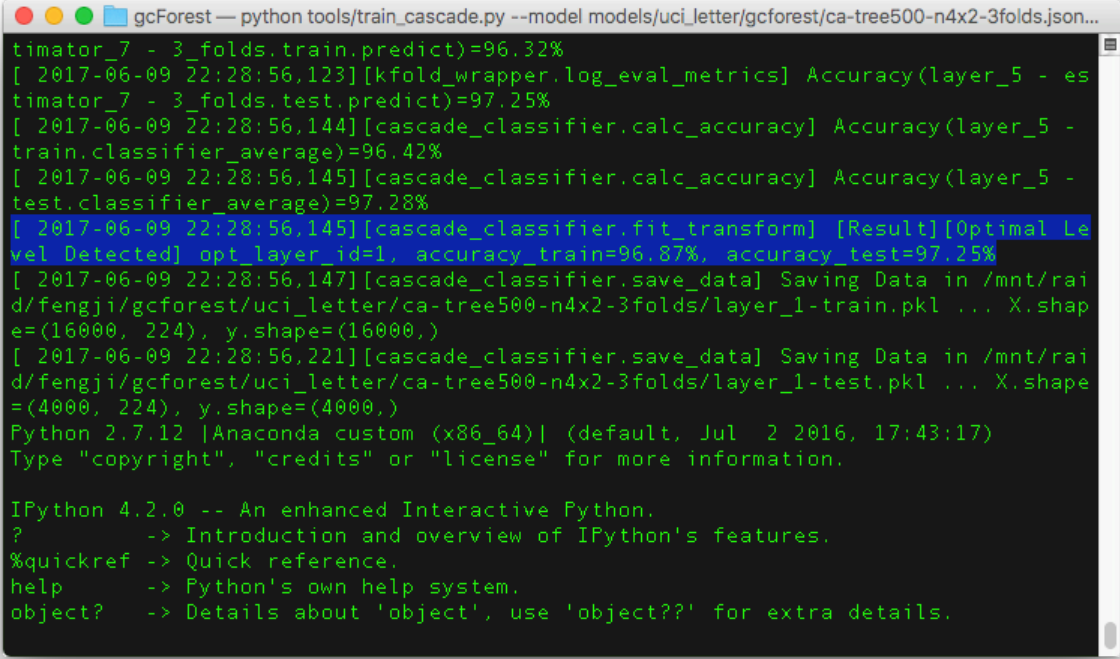
```
timator_4 - 3_folds.train_0.predict)=96.61%
[ 2017-06-09 21:58:41,955][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_4 - 3_folds.train_1.predict)=96.14%
[ 2017-06-09 21:58:55,775][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_4 - 3_folds.train_2.predict)=96.15%
[ 2017-06-09 21:58:56,836][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_4 - 3_folds.train.predict)=96.30%
[ 2017-06-09 21:58:56,837][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_4 - 3_folds.test.predict)=97.20%
[ 2017-06-09 21:59:10,863][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_5 - 3_folds.train_0.predict)=96.27%
[ 2017-06-09 21:59:25,555][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_5 - 3_folds.train_1.predict)=96.51%
[ 2017-06-09 21:59:40,471][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_5 - 3_folds.train_2.predict)=96.39%
[ 2017-06-09 21:59:41,522][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_5 - 3_folds.train.predict)=96.39%
[ 2017-06-09 21:59:41,522][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_5 - 3_folds.test.predict)=97.22%
[ 2017-06-09 21:59:55,165][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_6 - 3_folds.train_0.predict)=96.26%
[ 2017-06-09 22:00:10,232][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_6 - 3_folds.train_1.predict)=96.46%
```

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。。。。



```
gcForest — python tools/train_cascade.py --model models/uci_letter/gcforest/ca-tree500-n4x2-3folds.json...
timator_7 - 3_folds.train.predict)=96.32%
[ 2017-06-09 22:28:56,123][kfold_wrapper.log_eval_metrics] Accuracy(layer_5 - es
timator_7 - 3_folds.test.predict)=97.25%
[ 2017-06-09 22:28:56,144][cascade_classifier.calc_accuracy] Accuracy(layer_5 -
train.classifier_average)=96.42%
[ 2017-06-09 22:28:56,145][cascade_classifier.calc_accuracy] Accuracy(layer_5 -
test.classifier_average)=97.28%
[ 2017-06-09 22:28:56,145][cascade_classifier.fit_transform] [Result][Optimal Le
vel Detected] opt_layer_id=1, accuracy_train=96.87%, accuracy_test=97.25%
[ 2017-06-09 22:28:56,147][cascade_classifier.save_data] Saving Data in /mnt/rai
d/fengji/gcforest/uci_letter/ca-tree500-n4x2-3folds/layer_1-train.pkl ... X.shap
e=(16000, 224), y.shape=(16000,)
[ 2017-06-09 22:28:56,221][cascade_classifier.save_data] Saving Data in /mnt/rai
d/fengji/gcforest/uci_letter/ca-tree500-n4x2-3folds/layer_1-test.pkl ... X.shap
e=(4000, 224), y.shape=(4000,)
Python 2.7.12 |Anaconda custom (x86_64)| (default, Jul  2 2016, 17:43:17)
Type "copyright", "credits" or "license" for more information.

IPython 4.2.0 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.
```

```
[ 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%
```