PROJECT

杨帆 2020103661 流行病与卫生统计学

目录

大部分为代码结果，文字部分未超过三页

---

1 predict y values of the last 150 cases

```
## # A tibble: 1 x 8
##   penalty mixture .metric .estimator  mean     n std_err .config
##     <dbl>   <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1     0.1       1 rmse    standard    1.63    10   0.170 Preprocessor1_Model120
```

采用线性回归对前 150 个数据点建模，十折交叉验证通过网格搜索法调参选取参数，然后选取使得 mean squared error 最小的参数，交叉验证 mse 为 1.63，然后在前 150 个数据点训练得到最终的模型，最后用最终训练的模型预测后 150 个数据点的 y。

参数详情：

- penalty: The total amount of regularization in the model. Note that this must be zero for some engines.
- mixture: The mixture amounts of different types of regularization（L_0 与 L_1）.

2 identify the most important variables

2.1 list variables that you think are most important.

我认为 x30，x60，x32 最重要。

代码结果如下：

2.1.1 lasso

变量选择结果

```
##  [1]   1   3   9  10  30  31  32  33  34  45  46  52  60  99 108 170 176 221 252
## [20] 277 284 313 342 360 398 428 472 478 487
```

交叉验证 mse

```
## [1] 2.691511
```

stability test

```
## [1] 21.57
```

2.1.2 SCAD

变量选择结果

```
## [1]  30  32  60 313
```

交叉验证 mse

```
## [1] 3.36536
```

stability test

```
## [1] 13.93
```

2.1.3 MCP

变量选择结果

```
## [1]  30  32  60 313
```

交叉验证 mse

```
## [1] 3.36536
```

stability test

```
## [1] 5.88
```

lasso、SCAD、MCP 三种方法模型选择诊断结果

res	Fmeasure	Gmeasure	VSD	VSD_minus	VSD_plus
res_lasso_ARM	0.1990375	0.3300053	25.7796950	25.7796950	0.0000000
res_lasso_BIC	0.2186130	0.3497498	25.4321452	25.4321452	0.0000000
res_SCAD_ARM	0.8160558	0.8243591	1.2639081	1.0218015	0.2421066
res_SCAD_BIC	0.8930813	0.8972547	0.7930406	0.6125929	0.1804477
res_MCP_ARM	0.8160558	0.8243591	1.2639081	1.0218015	0.2421066
res_MCP_BIC	0.8930813	0.8972547	0.7930406	0.6125929	0.1804477

2.1.4 SOIL importance

（这里展示权重降序排列前十个变量，其余展示在网页版 rmarkdown）

ARM 加权结果

```
## # A tibble: 10 x 2
##    Variables importance
##    <chr>          <dbl>
##  1 x30          1.00
##  2 x60          1.00
##  3 x32          0.777
##  4 x31          0.187
##  5 x313         0.133
##  6 x34          0.0292
##  7 x3           0.0100
##  8 x1           0.00249
##  9 x33          0.00148
## 10 x10          0.00116
```

BIC 加权结果

```
## # A tibble: 10 x 2
##    Variables importance
##    <chr>          <dbl>
##  1 x30           1.
##  2 x60           1.
##  3 x32           1.00
##  4 x313          0.310
##  5 x31           0.309
##  6 x3            0.216
##  7 x1            0.201
##  8 x10           0.201
##  9 x33           0.199
## 10 x34           0.0182
```

AIC 加权结果

```
## # A tibble: 10 x 2
##    Variables importance
##    <chr>          <dbl>
##  1 x30          1.00
##  2 x60          1.00
##  3 x32          1.00
##  4 x31          0.994
##  5 x1           0.993
##  6 x10          0.993
##  7 x3           0.992
##  8 x33          0.991
##  9 x313         0.00667
## 10 x34          0.00234
```

由上可看到 lasso、SCAD、MCP 经过交叉验证后变量选择的结果：lasso 选了 29 个变量，而 SCAD、MCP 选了 4 个变量。

从交叉验证 mse 角度看，lasso 的 mse 是最小的，为 2.69，而 SCAD 和 MCP 相同，为 3.37，lasso 选的变量过多;stability test 检验，采用随机去掉 5% 的样本点来考查三种方法的稳定性，可以看出 lasso 是 21.57，SCAD 是 13.93，MCP 是 5.88，即 lasso、SCAD、SCAD 多选与少选加一起平均 21.57，13.93，5.88 个，也就是说以上三种方法具有较大的不稳定性。

三个方法比较而言，SCAD、MCP 变量选择的 F-measure 和 G-measure（二者是 recall 和 precision 的综合评估指标）大于 0.8，而且 VSD 的值 ARM 加权和 BIC 加权结果分别为 1.26 和 0.79，比 lasso 好很多。从 ARM、BIC 加权结果可以看出，SCAD 和 MCP 多选了一个变量，基本不存在少选的问题。

考虑到 SOIL importance，从三种加权方法结果可以看出，三种加权方法变量重要性值相对较大的都有 x30，x60，x32，x31；SCAD、MCP 变量选择中都有 x30，x60，x32，与 SOIL 结果一致，考虑到 SCAD、MCP 变量选择结果评估中不存在少选变量问题，所以这里选取 x30，x60，x32 作为最重要的变量。

所以，这里我认为最重要的变量是 x30，x60，x32。

2.2 final statistical model with estimated parameters

最后模型为

$$\hat{y} = -10.261 + 2.673x_{30} + 1.832x_{60} + 1.297x_{32}$$

2.3 model averaging methods and compare estimates of the coefficients of the variables

Variables	linear_reg	SAIC	L1_ARM	PMA
(Intercept)	-10.260987	-10.1779578	-10.240232	-9.2503321
x30	2.672682	2.2075420	2.597838	2.0839437
x32	1.297200	0.8334977	1.204109	0.8387390
x60	1.831991	-0.0110243	1.581548	0.7500798

我认为权重选取 L1-ARM 的模型平均结果更可靠。

从上面表格可以看出，L1-ARM 的系数估计与 (b) 中的 final model 系数估计最接近，SAIC、PMA 在 x32、x60 系数估计与 final model 差别过大；ARM 通过加权的结果一般情况下会更好。