使用 Neural Networks 预测社会地位

郑军威

目录

1 引言

本文使用 CGSS2015 数据，对个人对自己社会地位进行预测。社会地位分为两类：下层、上层。其中自变量影响因为包括：收入、性别、教育年限、户口、业余学习时间、与邻居交往频率（做为社会资本变量之一）、是否参加工会、工作经历。

2 数据处理

其中社会地位 level 是名义变量，0 代表下层，1 代表上层。分布如下：

```
summary(data$level)
```

```
##    0    1
## 6988  709
```

将数据分为 training and test 子数据

```
set.seed(123)
train =  createDataPartition(data$level,p=0.5,list=F)
data_train = data[train,]
data_test = data[-train,]
ytrue = data_test$level
```

3 Logistic Regression

```
fit <- glm(level~.,data_train,family='binomial')
summary(fit)
```

```
##
## Call:
## glm(formula = level ~ ., family = "binomial", data = data_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6497  -0.4689  -0.3598  -0.2477   3.1249
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.59712    0.07076 -36.703  < 2e-16 ***
## ln_income    0.67564    0.08075   8.367  < 2e-16 ***
## gender       0.06697    0.05883   1.138  0.25496
## edu          0.29259    0.06794   4.307 1.66e-05 ***
## hukou       -0.01249    0.06148  -0.203  0.83906
## study        0.16575    0.06366   2.604  0.00922 **
## neighbor    -0.24190    0.06213  -3.893 9.89e-05 ***
## union        0.02063    0.05531   0.373  0.70916
## experience  -0.03354    0.06100  -0.550  0.58242
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2368.5  on 3848  degrees of freedom
## Residual deviance: 2147.5  on 3840  degrees of freedom
## AIC: 2165.5
##
## Number of Fisher Scoring iterations: 6
```

其中错误率是：

```
phat = predict(fit,data_test,type="response")
yhat = as.numeric(phat > 0.5)
View(yhat)
table(ytrue,yhat)
```

```
##      yhat
## ytrue    0    1
##     0 3490    4
##     1  351    3
```

```
1-mean(yhat==ytrue)
```

```
## [1] 0.09225572
```

4 Neural Network

R 代码用 R Markdown 的语法嵌入，即三个反引号开始一段代码 ```{r} 和三个反引号 ``` 结束一段代码：使用命令

得到 Show in New WindowClear OutputExpand/Collapse Output size decay 4 1 1

```
set.seed(100)
require(nnet)
fit = nnet(level ~.,data=data_train,
           size=1,maxit=10000,MaxNWts=10000,decay=1)
```

```
## # weights:  11
## initial  value 2143.788096
## iter  10 value 1103.858342
## iter  20 value 1078.716157
## iter  30 value 1078.595116
## final  value 1078.595052
## converged
```

其中错误率是：

```
# test err
yhat1 = predict(fit,data_test,type="class")
table(ytrue,yhat1)
```

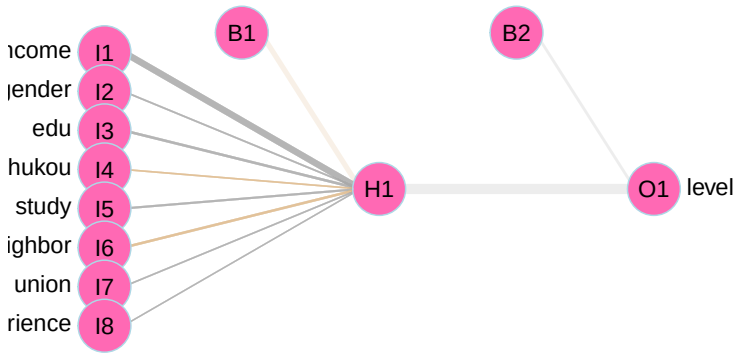
```
##      yhat1
## ytrue    0
##     0 3494
##     1  354
```

```
1-mean(yhat1==ytrue) #misclassification error rate
```

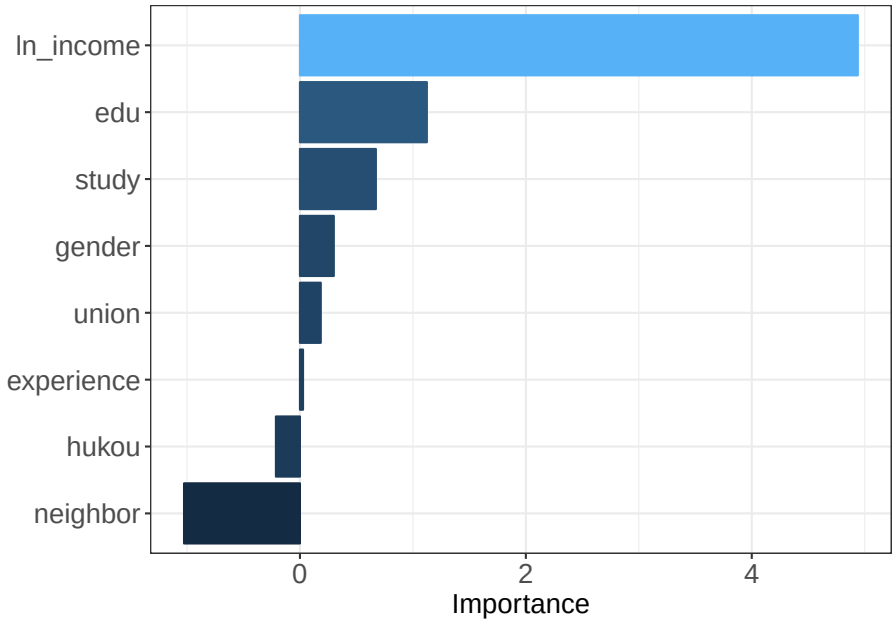
```
## [1] 0.09199584
```

可视化结果：

```
# vidualize network
plotnet(fit,alpha_val=.2,
        circle_col="hotpink",
        pos_col="burlywood",
        neg_col="darkgray")
```

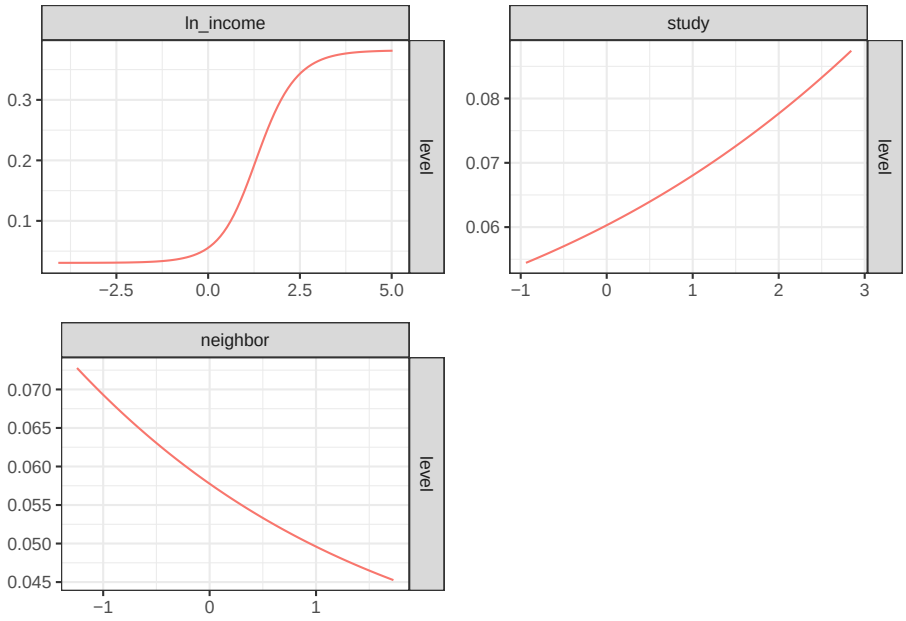



```
# importance plots based on weights
h = olden(fit)
h + coord_flip() + theme(axis.text=element_text(size=14),axis.title=element_text(size=1
```



```
# partial dependence plots of selected vars (other vars fixed at median)
h1 = lekprofile(fit,xsel=c("ln_income"),group_vals=0.5) +
  theme(legend.position="none",axis.title=element_blank())
h2 = lekprofile(fit,xsel=c("study"),group_vals=0.5) +
  theme(legend.position="none",axis.title=element_blank())
h3 = lekprofile(fit,xsel=c("neighbor"),group_vals=0.5) +
  theme(legend.position="none",axis.title=element_blank())

grid.arrange(h1,h2,h3,ncol=2)
```



5 小结

从错误率来看，Neural Network 模型要优于 Logistic 模型。在社会地位中，收入占绝对优势。