从用户评论看产品改善

张笑竹 / 201618070114

2019年10月19日

随着互联网的发展，用户评论出现在了生活的方方面面。对于消费者而言，书写用户评论是他们分享经历、抒发感受的途径。而对于商家而言，则可以从用户评论中挖掘有价值的信息来帮助厂商进行产品改善。本案例的数据包括2016年12月之前，某知名电商在其自营平台上销售的手机数据（共207部）以及能爬到的每款手机的全部用户评论数据（共140450条）。

1 任务一：准备工作与分词

首先，设置好工作路径。

```
setwd("C:\\Users\\小竹子\\Desktop\\新建文件夹")
```

然后，找到评论数据集comments_infor.csv。读入该数据，命名为com_data。

```
## 读入comments数据
com_data=read.csv("./comments_infor.csv", header = T, fileEncoding = 'utf-8',
                  stringsAsFactors = F)
```

随后，提取数据中的评论所在列，并命名为comments。

```
## 提取评论所在的列
comments=com_data$评论内容
```

利用分词器worker、停用词文件，对comments进行逐行分词。

```
## 加载R包
library(jiebaRD) #用于jieba分词
library(jiebaR)  #用于jieba分词
## 指定停用词的文件名
stoppath="./stopwords.dat"
## 初始化分词器，可以在分词的时候去停用词
cutter = worker(bylines = TRUE,stop_word=stoppath)
## 进行分词,这步会比较慢
res = cutter[comments]
```

分词后，利用head()查看前6行的分词情况.

```
head(res)
```

```
## [[1]]
## [1] "老爸" "喜欢" "支持" "支持" "支持" "下次"
##
## [[2]]
## [1] "运行" "挺快" "屏幕" "舒服"
##
## [[3]]
## [1] "不错"   "帮别人" "还好"
##
## [[4]]
## [1] "物流"   "很快"   "好不好"
##
## [[5]]
## [1] "不错" "感觉" "暂时" "发现" "缺点"
##
## [[6]]
## [1] "感觉" "不错" "音质" "划算"
```

2 任务二：挑选前50个热评词

下面，从分词后的全部文本中进行高频词提取。利用do.call将全部分词后的文本整合成一列。

```
## 将text从list转换为matrix格式
text = lapply(res,as.matrix)
## 将每行文本的分词结果逐一排列起来
text = as.data.frame(do.call(rbind, text), stringsAsFactors = F)
```

对整合后的数据使用table函数进行词频统计。

```
## 进行词频统计
freq = as.data.frame(table(text),stringsAsFactors = F)
## 按词频个数降序排列
freq = freq[order(-freq[,2]),]
```

挑选前50个高频词，并输出。

```
top50 = freq[1:50,]
top50
```

```
##         text  Freq
## 5372    不错 38064
## 30090   喜欢 11098
## 13041   好好  8698
## 11256   感觉  8118
## 19844   满意  7931
## 22399   屏幕  7688
## 9019    电池  6893
## 30108   系统  6184
## 13543   很快  5753
## 26825   速度  5677
## 17522   客服  5582
## 25696   收到  5357
## 35579   支持  5291
## 29955   物流  5279
## 35845   质量  5156
## 17958   垃圾  5072
## 17802   快递  4822
## 26687     送  4649
## 11419     高  4615
## 19196   流畅  4602
## 28683   外观  4510
## 21799   拍照  4277
## 12744   还行  4266
## 14049   华为  4242
## 15497   价格  4175
## 31400 性价比  4151
## 6273      差  4045
## 25775   手感  3992
## 13197   好用  3989
## 6903    充电  3928
## 6329    差评  3681
## 13136   好评  3678
## 22296   评价  3643
## 5473    不好  3620
## 12080   购买  3513
## 35725   值得  3492
## 15177   几天  3293
## 9978    耳机  3227
## 34218   运行  3202
## 21981   朋友  3171
## 11981   功能  3121
## 34104     月  3076
## 10191   发热  3026
## 30025   希望  3020
## 16865     卡  2958
## 13099   好看  2843
## 26723   送货  2678
## 14177     换  2676
## 8873  第一次  2639
## 30627     想  2631
```

3 任务三：手机型号的热评词频

top50高频词中出现了很多和手机性能或者平台服务相关的词。这里，仅以“屏幕”，“电池”，“客服”，“物流”4个词为例来进行后续分析。

首先，在top50高频词中找出这几个词的标号，然后把它们命名为key_words。

```
## 找出前50个词中有明确含义的属性词，其标号分别为
sele_num=c(6,7,11,14)
## 找出key_words
key_words=top50[sele_num,1]
key_words
```

```
## [1] "屏幕" "电池" "客服" "物流"
```

然后，考虑这4个词对手机好评率的影响。

①构造每个热评词是否出现的0-1矩阵，以每一条评论为代表行，以每一个热评词代表列。例如：

$$x_{ij}=\begin{cases}0, & \text{第}i\text{条评论没有出现第}j\text{个热评词}\\ 1, & \text{第}i\text{条评论出现了第}j\text{个热评词}\end{cases}$$

```
##计算所有评论总数
N=length(comments)
##计算key_words的个数
K=length(key_words)
##初始化结果矩阵，判断每个热评词是否在每条评论中出现，如果出现记为1，不出现记为0
key_mat=matrix(0,nrow=N,ncol=K)
##对每个词进行循环
for(k in 1:K)
{
  theword=key_words[k]    #找到该热评词
  num=grep(theword,comments)   #判断出现该热评词的评论标号
  key_mat[num,k]=1   #记录那些评论包括该热评词
}
```

②构造每个类型手机的热评词词频矩阵，以每一种手机型号代表行，以每一个热评此代表列。例如：

$$x_{ij}:=\text{对第}j\text{个热评词，第}i\text{种手机型号的比例.}$$

```
##提取评论数据集com_data中的手机编号信息
phoneid=com_data$手机编号
##统计每部手机的评论总数
NO_comments=table(phoneid)
##计算手机ID的个数
I=length(unique(phoneid))
##初始化结果矩阵，每行为一部手机，每列为这部手机出现该热评词的频率
freq_mat=matrix(0,nrow=I,ncol=K)
##为每个热评词计算频率
for(k in 1:K)
{
  freq_mat[,k]=tapply(key_mat[,k],phoneid,sum)/NO_comments   #统计每部手机中，第k个热评词的频率
}
```

③输出上面构造的“每个类型手机的热评词词频矩阵”。

```
##找出手机的id
iid=unique(phoneid)
##添加手机ID，便于下面和手机的其他数据进行合并
com_reg=data.frame(iid,freq_mat)
##为com_reg进行命名
colnames(com_reg)=c("手机编号",key_words)
head(com_reg)
```

```
##    手机编号        屏幕        电池        客服        物流
## 1  2876449 0.08021978 0.07912088 0.01868132 0.01648352
## 2  3097849 0.07107843 0.10294118 0.02818627 0.01225490
## 3  3235724 0.07634409 0.09032258 0.02688172 0.01827957
## 4  3398125 0.03150685 0.04726027 0.03082192 0.03493151
## 5  3783186 0.07142857 0.02952381 0.03047619 0.02285714
## 6  3828650 0.06099291 0.06879433 0.04184397 0.02978723
```

4 任务四：“好评率”对“热评词”的回归

读入手机数据集phone_infor.csv，读入并将其命名为phone_infor。使用merge函数将上步得到的com_reg和这个数据进行整合。

然后，计算“好评率=好评数/总评论数”作为因变量，建立线性回归模型探索“屏幕”，“电池”，“客服”，“物流”对手机的好评率的影响。此外，在建立回归模型时还放入了价格、品牌、屏幕尺寸、指纹识别、GPS定位、促销信息等其他解释性变量。

```
## 读入手机数据
phone_infor=read.csv("./phone_infor.csv", header=T, fileEncoding = 'utf-8',
                     stringsAsFactors = F)
## 使用merge将phone_infor和com_reg按照"手机编号"整合到一起
reg_data=merge(phone_infor, com_reg, by = "手机编号", all.x = TRUE)
## 计算好评率
good_freq=reg_data$好评数/reg_data$总评论数
## 建立回归模型
model=lm(good_freq~价格+品牌+屏幕尺寸+指纹识别+GPS定位+促销信息
         +屏幕+电池+客服+物流
         ,data=reg_data)
## 查看回归结果
summary(model)
```

```
##
## Call:
## lm(formula = good_freq ~ 价格 + 品牌 + 屏幕尺寸 + 指纹识别 +
##     GPS定位 + 促销信息 + 屏幕 + 电池 + 客服 + 物流, data = reg_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.108672 -0.009230  0.000221  0.009342  0.059870
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           9.182e-01  2.906e-02  31.599   <2e-16 ***
## 价格                  3.340e-06  1.601e-06   2.086   0.0383 *
## 品牌OPPO              5.286e-03  1.391e-02   0.380   0.7044
## 品牌vivo              1.200e-02  1.345e-02   0.892   0.3734
## 品牌华为（HUAWEI）    1.469e-02  1.079e-02   1.361   0.1752
## 品牌乐视（Letv）     -3.164e-02  1.543e-02  -2.050   0.0417 *
## 品牌努比亚（nubia）  -4.965e-03  1.397e-02  -0.355   0.7228
## 品牌苹果（Apple）    -1.019e-02  1.527e-02  -0.667   0.5054
## 品牌其他             -1.303e-02  1.039e-02  -1.255   0.2111
## 品牌三星（SAMSUNG）  -1.327e-02  1.342e-02  -0.988   0.3242
## 品牌小米（MI）       -1.152e-04  1.270e-02  -0.009   0.9928
## 屏幕尺寸              7.548e-03  5.106e-03   1.478   0.1411
## 指纹识别其他         -1.267e-02  7.415e-03  -1.708   0.0892 .
## 指纹识别支持         -9.720e-04  6.641e-03  -0.146   0.8838
## GPS定位支持          -2.513e-03  8.564e-03  -0.293   0.7695
## 促销信息有           -5.745e-03  3.662e-03  -1.569   0.1184
## 屏幕                  5.158e-02  5.584e-02   0.924   0.3568
## 电池                 -9.103e-02  4.126e-02  -2.207   0.0286 *
## 客服                  5.995e-02  7.827e-02   0.766   0.4447
## 物流                 -1.483e-02  3.856e-02  -0.385   0.7010
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02361 on 187 degrees of freedom
## Multiple R-squared:  0.3017, Adjusted R-squared:  0.2307
## F-statistic: 4.252 on 19 and 187 DF,  p-value: 8.923e-08
```

通过上述结果，不难发现，“价格”、“品牌”和热评词“电池”这三个变量是显著的。现在，只考虑热评词“电池”的效应。由于其参数的估计值为$\hat{\beta}_k = -9.103 \times 10^{-2}$，说明“电池”热评频率每上升1%，好评率就会平均下降-9.103×10^{-2}. 可以推测，大部分关于电池的热评，都由于电池质量不佳而降低了对产品的满意度。

5 任务五：电池背后的具体“关注点”

上述回归结果显示，“电池”是显著为负的，说明电池是手机的一个减分项。那么，顾客对电池的不满大多出自哪里？人们在提到电池的时候都说了些什么？

①提取出包含“电池”的评论

```
## 统计句子，我们按照“，。！？”来表明一个句子
sentence=strsplit(comments,", |。|! |? ")
sentence2=unlist(sentence)
## 判断每个句子中是否包含“电池”
aa=lapply(sentence2,function(x){grep("电池",x)})
## 如果句子不包含电池，aa在相应位置的结果是integer(0)，如果可以通过计算aa每个位置的长度来判断是否出现了integer(0)
bb=lapply(aa,length)
## 将bb由list转换为向量
bb=unlist(bb)
## 提取出bb中大于0的评论编号，就是
com_dianchi=sentence2[bb>0]
head(com_dianchi)
```

```
## [1] "希望电池持久"                    "就是电池"
## [3] "唯一的缺点就是电池太不耐用了" "微中不足之处就是电池用得太快"
## [5] "除了电池不行"                    "电池可用一天"
```

②挑选“电池”评论中的前50个“热评词”

```
## 对只包含电池的短句进行分词
res=cutter[com_dianchi]
## 将分词后的结果从list转换为matrix格式
res=lapply(res,as.matrix)
## 将每行文本的分词结果逐一排列起来
res = as.data.frame(do.call(rbind, res), stringsAsFactors = F)
## 计算每个词出现的词频
freq = as.data.frame(table(res),stringsAsFactors = F)
## 按词频个数降序排列
freq = freq[order(-freq[,2]),]
## 挑选前50个高频词
top50_dianchi = freq[1:50,]
top50_dianchi
```

```
##           res Freq
## 692      电池 6893
## 1586     耐用 1851
## 381      不错  311
## 2363     续航  292
## 1114       换  215
## 632      待机  201
## 1012     毫安  191
## 2319     小时  168
## 394      不行  164
## 1982       送  158
## 556      充电  156
## 912      给力  142
## 296      半天  136
## 2256     系统  127
## 878      感觉  125
## 383      不到  113
## 395      不好  107
## 1873     时间  106
## 2430     一块  106
## 805      发热  101
## 1509     没电   94
## 1678     屏幕   94
## 1361     垃圾   93
## 945      够用   91
## 691        电   90
## 1041     耗电   86
## 683        点   80
## 920      更换   80
## 1607     能力   80
## 435      不太   79
## 2197     唯一   79
## 2657     只能   79
## 1496     满意   78
## 2679       中   77
## 997      还行   76
## 2407     一点   76
## 1319     客服   74
## 2729     自带   74
## 1777     软件   73
## 1312     可换   72
## 1750     缺点   71
## 1991     速度   71
## 2599       长   71
## 81       5000   70
## 693  电池电量   70
## 1430     两天   70
## 2297     想象   68
## 1063     很快   67
## 2682     中度   67
## 783      耳机   66
```

6 任务六：“关注点”分析

然而，我们不满足于仅仅得到关注点，还希望探究用户对各个关注点的态度。比如，我们想知道大家对“耐用”，“续航”，“待机”，“充电”这四个具体关注点是满意还是不满意。完成这一步就需要借助用户评论中的评分数据了。

①提取“耐用”，“续航”，“待机”，“充电”4个关注点的数据框。

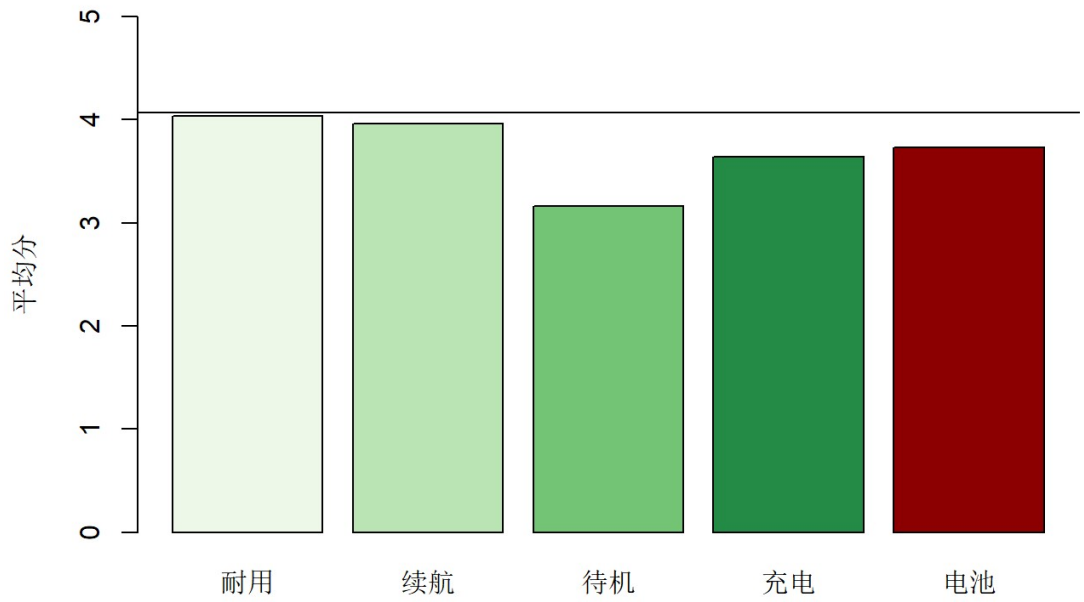
```
ratings=com_data$评论得分
tnum=grep("电池",comments)
comments2=comments[tnum]
ratings2=ratings[tnum]
## 统计在出现电池的评论中，出现某个关注点的评论标号
num1=grep("耐用",comments2)
num2=grep("续航",comments2)
num3=grep("待机",comments2)
num4=grep("充电",comments2)
## 得到这些评论的相应得分
rat_NY=ratings2[num1]
rat_XH=ratings2[num2]
rat_DJ=ratings2[num3]
rat_CD=ratings2[num4]
```

②计算4个关注点，包含“电池”用户，以及全行业的评论平均分。

```
##计算每个关注点的得分的平均分
rr1=mean(rat_NY)
rr2=mean(rat_XH)
rr3=mean(rat_DJ)
rr4=mean(rat_CD)
##计算所有包含电池的用户评论的平均分
rr=mean(ratings2)
##计算行业标准，即所有评论的平均分
themean=mean(ratings)
```

③做出条形图，进行Expository Data Visualization.

```
##将所有平均分放入一个向量并命名
ruse=c(rr1,rr2,rr3,rr4,rr)
names(ruse)=c("耐用","续航","待机","充电","电池")
####作图进行对比
library(RColorBrewer) #用于画图时调用更多颜色
barplot(ruse,col=c(brewer.pal(4, 'Greens'),"dark red"),ylim=c(0,5),ylab="平均分")
abline(h=themean)
```



在上图中，黑色的水平线表示行业均值，最右边红色的柱子显示了电池的整体分数，它明显低于行业均值，说明用户普遍对电池不满，这也是回归结果中电池表现负显著的原因。细看各个关注点，耐用和续航都高于电池的得分，并且接近行业标准；充电的得分和电池的平均分大致持平；相反，待机得分明显偏低。这说明待机正是用户对电池不满意的原因所在，也是厂商可以进一步改进的方向。