R 语言编程：基于 tidyverse

第 23 讲　特征工程，探索变量关系

张敬信

2022 年 12 月 6 日

哈尔滨商业大学

自变量通常称为特征，**特征工程**（Feature Engineering），就是发现或构建对因变量有明显影响作用的特征，具体来说是将原始特征转化成更好的表达问题本质的特征的过程，将这些特征运用到预测模型中能提高对不可见数据的模型预测精度。

数据清洗、特征工程属于机器学习中的数据预处理环节，R 机器学习框架 `tidymodels` 下的 `recipes` 包，`mlr3verse` 下的 `mlr3pipelines` 包都能更系统方便地实现，但也更抽象，这里优先不用它们实现。

一. 特征缩放

不同数值型特征的数据量纲可能相差多个数量级，这对很多数据模型会有很大影响，所以有必要做归一化处理，就是将列或行对齐并转化为一致。

1. 标准化

标准化也称为 Z 标准化，将数据变成均值为 0，标准差为 1：

$$z = \frac{x - \mu}{\sigma}$$

其中，μ 为均值，σ 为标准差。Z 值反映了该值偏离均值的标准差的倍数。

```
scale(x)                        # 标准化
scale(x, scale = FALSE)         # 中心化：减去均值
```

注： 中心化后，0 就代表均值，更方便模型解释。

2. 归一化

归一化是将数据线性放缩到 $[0,1]$，一般还同时考虑指标一致化，将正向指标(值越大越好）和负向指标（值越小越好）都变成正向。

正向指标：

$$x_i' = \frac{x_i - \min x_i}{\max x_i - \min x_i}$$

负向指标：

$$x_i' = \frac{\max x_i - x_i}{\max x_i - \min x_i}$$

注：根据需要也可以线性放缩到 $[a,b]$.

- 定义归一化函数:

```
rescale = function(x, type = "pos", a = 0, b = 1) {
  rng = range(x, na.rm = TRUE)
  switch (type,
    "pos" = (b - a) * (x - rng[1]) / (rng[2] - rng[1]) + a,
    "neg" = (b - a) * (rng[2] - x) / (rng[2] - rng[1]) + a)
}
```

```r
iris %>%
  as_tibble() %>%       # 将所有数值列归一化到 [0,100]
  mutate(across(where(is.numeric), rescale, b = 100))
#> # A tibble: 150 x 5
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width Spe
#>          <dbl>       <dbl>        <dbl>       <dbl> <fc
#> 1         22.2        62.5         6.78        4.17 set
#> 2         16.7        41.7         6.78        4.17 set
#> 3         11.1        50           5.08        4.17 set
#> # ... with 147 more rows
```

3. 行规范化

行规范化，常用于文本数据或聚类算法，是保证每行具有单位范数，即每行的向量”长度”相同。想象一下，m 个特征下，每行数据都是 m 维空间中的一个点，做行规范化能让这些点都落在单位球面上（到原点的距离均为 1)。

行规范化，一般采用 L_2 范数：

$$x'_{ij} = \frac{x_{ij}}{\|\mathrm{x_i}\|} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{m} x_{ij}^2}}$$

```r
iris[1:3,-5] %>%
  pmap_dfr(~ c(...) / norm(c(...), "2"))
#> # A tibble: 3 x 4
#>   Sepal.Length Sepal.Width Petal.Length Petal.Width
#>          <dbl>       <dbl>        <dbl>       <dbl>
#> 1        0.804       0.552        0.221      0.0315
#> 2        0.828       0.507        0.237      0.0338
#> 3        0.805       0.548        0.223      0.0343
```

若数据噪声太多的问题，通常就需要做数据平滑。

最简单的数据平滑方法是移动平均，即用一定宽度的小窗口[1]滑过曲线，会把曲线的毛刺尖峰抹掉，能一定程度上去掉噪声还原原本曲线。

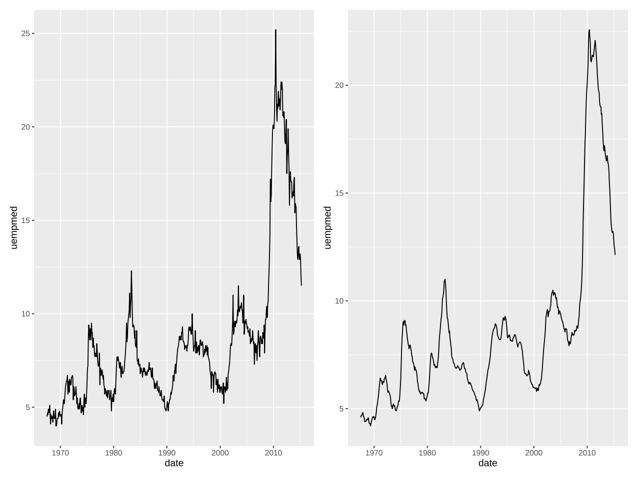
窗口宽度越大，平滑的效果越明显。

```
library(slider)
library(patchwork)
p1 = economics %>%
  ggplot(aes(date, uempmed)) +
  geom_line()
```

[1]比如五点平滑，用前两点/自身/后两点，共五点平均值代替自身因变量值

```r
p2 = economics %>%      # 做五点移动平均
  mutate(uempmed = slide_dbl(uempmed, mean,
                             .before = 2, .after = 2)) %>%
  ggplot(aes(date, uempmed)) +
  geom_line()
```

```r
p1 | p2
```



`lowess()`: 局部加权多项式回归平滑，更多的平滑方法还有指数平滑、滤波、光滑样条等。

1. 非线性特征

对于数值特征 x1, x2, ..., 可以创建更多的多项式项特征: x1^2, X1 x2, x2^2, ..., 这相当于是用自变量的更高阶泰勒公式去逼近因变量。

这里[2]再给出一种 recipes 包的实现，整体上是管道流操作:

- `recipe()`: 准备数据和模型变量
- `step_poly()`: 特征工程步，构建单变量的多项式特征，参数 `degree` 设置多项式次数，默认是生成正交多项式特征，原始特征需要设置 `raw = TRUE`
- `prep()`: 用数据估计特征工程步参数
- `bake()`: 应用到新数据，`new_data = NULL` 表示应用到原数据

---

[2]在多元线性回归实例中，已给出了借助 `I()`, `poly()`, `mpoly()` 生成多项式项，加入回归模型公式.

```r
library(tidymodels)
recipe(hwy ~ displ + cty, data = mpg) %>%
  step_poly(all_predictors(), degree = 2,
            options = list(raw = TRUE)) %>%
  prep() %>%
  bake(new_data = NULL)
#> # A tibble: 234 x 5
#>     hwy displ_poly_1 displ_poly_2 cty_poly_1 cty_poly_2
#>   <int>        <dbl>        <dbl>      <dbl>      <dbl>
#> 1    29          1.8         3.24         18        324
#> 2    29          1.8         3.24         21        441
#> 3    31          2           4            20        400
#> # ... with 231 more rows
```

也可以构建其他非线性特征，以及样条特征、广义加法模型特征等。另外，文本数据有专用的文本特征（词袋、TF-IDF 等）。

2. 正态性变换

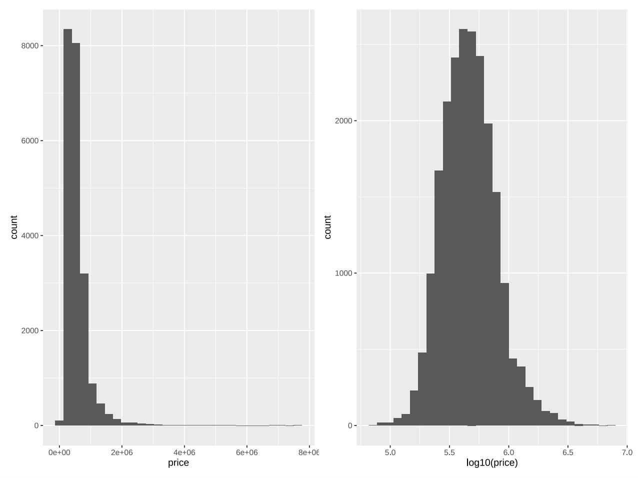
- 对数变换或幂变换

对于方差逐渐变大的异方差的时间序列数据，或右偏分布的数据，可以尝试做对数变换或开根号变换，以稳定方差和变成正态分布：

$$y' = \log_a(y), \qquad y' = \sqrt{y}, \qquad y' = \sqrt[3]{y}$$

- 以 King Country 的房价数据为例，右偏分布做对数变换后变成近似正态分布：

```
df = mlr3data::kc_housing
p1 = ggplot(df, aes(price)) +
  geom_histogram()
p2 = ggplot(df, aes(log10(price))) +
  geom_histogram()
p1 | p2
```



count
8000
6000
4000
2000
0
0e+00
2e+06
4e+06
6e+06
8e+06
price
count
2000
1000
0
5.0
5.5
6.0
6.5
7.0
log10(price)

对数变换特别有用，因为具有可解释性：对数值的变化是原始尺度上的相对(百分比）变化。若使用以 10 为底的对数，则对数刻度上每增加 1 对应原始刻度上的乘以 10。

注意，原始数据若存在零或负，则不能取对数或开根号，解决办法是做平移：$a = \max\{0, -\min\{x_i\} + \varepsilon\}$.

- Box-Cox 变换与 Yeo-Johnson 变换

Box-Cox 变换是更神奇的正态性变换，用最大似然估计选择最优的 λ 值，让非负的非正态数据变成正态数据：

$$y' = \begin{cases} \ln(y), & \lambda = 0 \\ (y^\lambda - 1)/\lambda, & \lambda \neq 0 \end{cases}$$

若数据包含 0 或负数，则 `Box-Cox` 变换不再适用，可以改用同样原理的 `Yeo-Johnson` 变换：

$$y' = \begin{cases} \ln(y+1), & \lambda = 0,\, y \geq 0 \\ \dfrac{(y+1)^{\lambda} - 1}{\lambda} & \lambda \neq 0,\, y \geq 0 \\ -\ln(1-y), & \lambda = 2,\, y < 0 \\ \dfrac{(1-y)^{2-\lambda} - 1}{\lambda - 2}, & \lambda \neq 2,\, y < 0 \end{cases}$$

用 bestNormalize 包中的 boxcox() 和 yeojohnson() 函数，可以实现这两种变换及其逆变换：

```
library(bestNormalize)
x = rgamma(100, 1, 1)
yj_obj = yeojohnson(x)
yj_obj$lambda                  # 最优 lambda
#> [1] -0.795
p = predict(yj_obj)                                    # 变换
x2 = predict(yj_obj, newdata = p, inverse = TRUE) # 逆变换
```

3. 连续变量离散化

在统计和机器学习中，有时需要将连续变量转化为离散变量，称为**连续变量离散化**或分箱，常用于银行风控建模，特别是线性回归或 Logistic 回归模型。

分箱的好处有：

- 使得结果更便于分析和解释。比如，年龄从中年到老年，患高血压比例增加 25%，而年龄每增加一岁，患高血压比例不一定有显著变化；
- 简化模型，将自变量与因变量间非线性的潜在的关系，转化为简单的线性关系。

当然，分箱也可能带来问题：简化的模型关系可能与潜在的模型关系不一致 (甚至发现的是错误的模型关系)、删除数据中的细微差别、切分点可能没有实际意义。

`rbin` 包提供了简单的分箱方法：

- `rbin_manual()`：自定义分箱，手动指定切分点（左闭右开）
- `rbin_equal_length()`：等宽分箱
- `rbin_equal_freq()`：等频分箱
- `rbin_quantiles()`：分位数分箱
- `rbin_winsorize()`：缩尾分箱，不受异常值影响

这些函数返回分箱结果的汇总统计以及 WOE、熵和信息值指标，用 `rbin_create()` 可以进一步创建虚拟变量。

```r
library(rbin)
df = readxl::read_xlsx("data/hyper.xlsx")
bins = df %>%
  rbin_equal_length(hyper, age, bins = 3)
rbin_create(df, age, bins) %>% head(3)
#>   age ageg hyper age_<_49 age_<_58 age_>=_58
#> 1  58    2     1        0        0         1
#> 2  50    2     1        0        1         0
#> 3  56    2     1        0        1         0
```

其他基于模型的分箱方法还有，基于 k-means 聚类、决策树、ROC 曲线、广义可加模型、最大秩统计量等。

另外，分类特征在用于回归建模或机器学习模型之前，经常需要做重新编码：转化虚拟变量（见 4.4.3 节）、效应编码等。

三. 特征降维

有时数据集可能包含过多特征，甚至是冗余特征，可以用降维技术进压缩特征，但通常会降低模型性能。

最常用的**特征降维**方法是主成分分析（PCA），是利用协方差矩阵的特征值分解原理，实现多个特征向少量综合特征（称为主成分）的转化，每个主成分都是多个原始特征的线性组合，且各个主成分之间互不相关，第一主成分是解释数据变异（方差）最大的，然后是次大的，依此类推。

n 个特征，若转化为 n 个主成分，则会保留原始数据的 100% 信息，但这就失去了降维的意义。所以一般是只选择前若干个主成分，一般原则是选择至保留 85% 以上信息的主成分。

用 `recipes` 包实现，关键步骤是特征工程步 `step_pca()`，`threshold` 设置保留信息的阈值，或用 `num_comp` 设置保留主成分个数。

```r
recipe(~ ., data = iris) %>%
  step_normalize(all_numeric()) %>%
  step_pca(all_numeric(), threshold = 0.85) %>%
  prep() %>%
  bake(new_data = NULL)
#> # A tibble: 150 x 3
#>   Species   PC1    PC2
#>   <fct>   <dbl>  <dbl>
#> 1 setosa  -2.26 -0.478
#> 2 setosa  -2.07  0.672
#> 3 setosa  -2.36  0.341
#> # ... with 147 more rows
```

其他特征降维的方法，还有核主成分分析、独立成分分析（ICA）、多维尺度等。

四. 探索变量间的关系

数据中的变量值得去关注，是因为变量自身的变化（取常值的变量毫无价值），以及变量与变量之间的协变化。

描述统计相当于是探索单个变量自身的变化。比如，连续变量可以用均值等汇总统计量、直方图、箱线图探索其分布；离散变量可以用频率表、条形图等。

探索性数据分析另一重要内容就是探索变量间的关系，也叫作探索协变化。

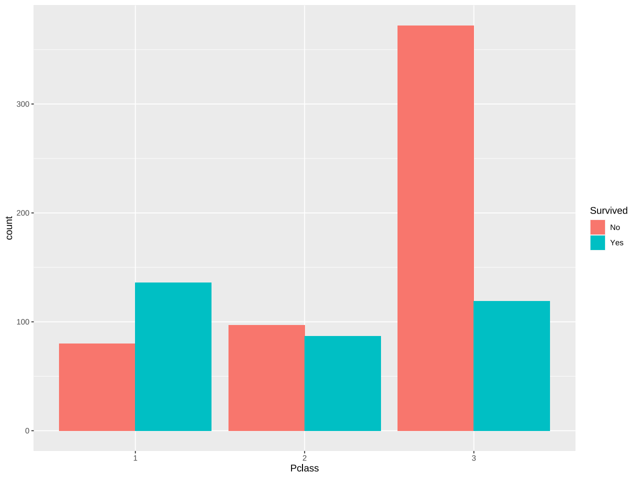
协变化是两个或多个变量的值以一种相关的方式一起变化。识别出协变化的最好的方式，将两个或多个变量的关系可视化，当然也要区分变量是分类变量还是连续变量。

探索两个分类变量的常用方法：

- 可视化：复式条形图、堆叠条形图
- 描述统计量：交叉表
- Cramer’s V 统计量：rstatix::cramer_v()
- 假设检验：检验两个比例的差、卡方独立性检验

```
titanic = read_rds("data/titanic.rds")
```

```r
titanic %>%
  ggplot(aes(Pclass, fill = Survived)) +
  geom_bar(position = "dodge")
```



```r
library(rstatix)
tbl = table(titanic$Pclass, titanic$Survived)
cramer_v(tbl)                    # Cramer'V 检验
#> [1] 0.34
prop_test(tbl)                   # 比例检验
#> # A tibble: 1 x 5
#>       n statistic    df        p p.signif
#> * <dbl>     <dbl> <dbl>    <dbl> <chr>
#> 1   891      103.     2 4.55e-23 ****
```

```
chisq_test(tbl)            # 卡方检验
#> # A tibble: 1 x 6
#>       n statistic        p    df method           p.signi
#> * <int>     <dbl>    <dbl> <int> <chr>            <chr>
#> 1   891      103. 4.55e-23     2 Chi-square test ****
```

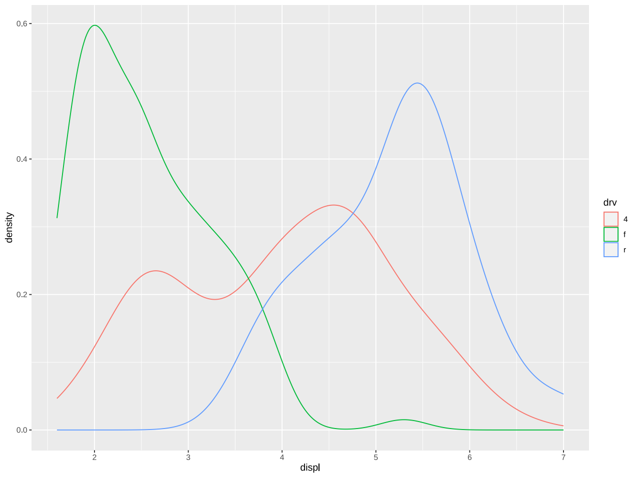
Cramer'V 统计量是修正版本的 Φ 系数，一般法则是：$|\Phi| < 0.3$, 很少或没有相关性；$0.3 \leq |\Phi| \leq 0.7$, 有弱相关性；$|\Phi| > 0.7$, 有强相关性。

2. 分类变量与连续变量

探索分类变量与连续变量的常用方法:

- 可视化:按分类变量分组的箱线图、直方图、概率密度曲线;
- 描述统计:按分类变量分组汇总;
- 比较均值的假设检验:t 检验、方差分析、Wilcoxon 秩和检验等

```r
ggplot(mpg, aes(displ, color = drv)) +
  geom_density()                    # 概率密度曲线
```



```r
mpg %>%
  group_by(drv) %>%
  get_summary_stats(displ, type = "five_number") # 五数汇总
#> # A tibble: 3 x 8
#>   drv   variable     n   min   max    q1 median    q3
#>   <chr> <fct>    <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>
#> 1 4     displ      103   1.8   6.5   2.9    4     4.7
#> 2 f     displ      106   1.6   5.3   2      2.4   3
#> 3 r     displ       25   3.8   7     4.6    5.4   5.7
```

```r
mpg %>%
  anova_test(displ ~ drv)          # 方差分析
#> ANOVA Table (type II tests)
#>
#>   Effect DFn DFd   F        p p<.05   ges
#> 1    drv   2 231 110 3.03e-34     * 0.487
```

3. 两个连续变量

探索两个连续变量的常用方法:

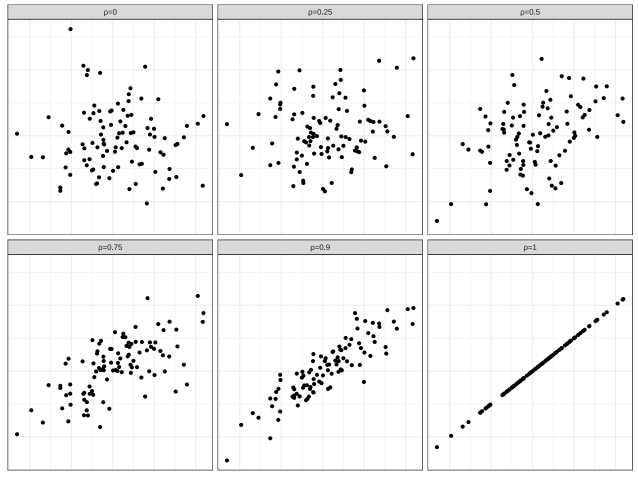
- 可视化：散点图（或 + 光滑曲线)、折线图，3 个连续变量可用气泡图
- 线性相关系数：协方差能反映两个变量的影响关系，定义为

$$\mathrm{Cov}(X,Y)=\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x_i})(y_i-\bar{x_i})$$

但是协方差的单位是不一致的，不具有可比性，解决办法就是做标准化，得到相关系数:

$$\rho_{XY}=\frac{\mathrm{Cov}(X,Y)}{s_X s_Y}=\frac{\sum_{i=1}^{n}(x_i-\bar{x_i})(y_i-\bar{x_i})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}$$

线性相关系数介于 -1 和 1 之间，反映了线性相关程度的大小。



ρ=0
ρ=0.25
ρ=0.5
ρ=0.75
ρ=0.9
ρ=1

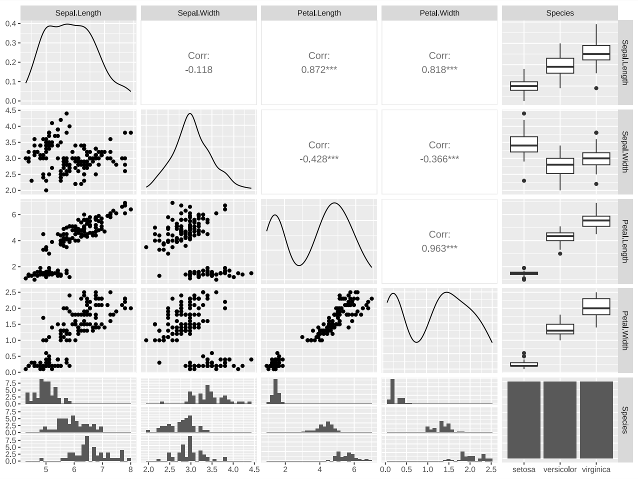
用 rstatix 包计算相关系数矩阵，并去掉重复，按相关系数大小排序：

```
iris[-5] %>%
  cor_mat() %>%                        # 相关系数矩阵
  replace_triangle(by = NA) %>%        # 将下三角替换为 NA
  cor_gather() %>%                     # 宽变长
  arrange(- abs(cor))                  # 按绝对值降序排列
#>           var1         var2   cor        p
#> 1 Petal.Length  Petal.Width  0.96 4.68e-86
#> 2 Sepal.Length Petal.Length  0.87 1.04e-47
#> 3 Sepal.Length  Petal.Width  0.82 2.33e-37
#> 4  Sepal.Width Petal.Length -0.43 4.51e-08
#> 5  Sepal.Width  Petal.Width -0.37 4.07e-06
#> 6 Sepal.Length  Sepal.Width -0.12 1.52e-01
```

注意：统计相关并不代表因果相关！线性不相关也可能具有非线性关系！

GGally 包提供的 ggpair() 函数绘制散点图矩阵，非常便于可视化探索因变量与多个自变量之间的相关关系：

```
library(GGally)
ggpairs(iris, columns = names(iris))
```


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species
Corr:
-0.118
Corr:
0.872***
Corr:
0.818***
Corr:
-0.428***
Corr:
-0.366***
Corr:
0.963***
setosa
versicolor
virginica

实际中，经常需要从许多自变量中筛选对因变量有显著影响的，根据相关系数是方法之一，更系统的方法是机器学习中的特征选择。另外，`correlationfunnel` 包能够快速探索自变量，特别是大量分类变量，对因变量的相关性影响大小，并绘制”相关漏斗图”进行可视化。

最后，还可以通过构建线性回归或广义线性回归模型，查看回归系数是否显著来探索自变量（无论是连续还是分类）对因变量的影响。

本篇主要参阅 (张敬信, 2022), (锡南·厄兹代米尔, 2019), (Hadley Wickham, 2017), 以及包文档，模板感谢 (黄湘云, 2021), (谢益辉, 2021).

参考文献

Hadley Wickham, G. G. (2017). *R for Data Science*. O′ Reilly, 1 edition. ISBN 978-1491910399.

张敬信 (2022). *R* 语言编程：基于 *tidyverse*. 人民邮电出版社, 北京.

谢益辉 (2021). *rmarkdown: Dynamic Documents for R*.

锡南·厄兹代米尔, 迪夫娅·苏萨拉， . (2019). 特征工程入门与实践. 人民邮电出版社, 北京.

黄湘云 (2021). *Github: R-Markdown-Template*.