北京大学
PEKING UNIVERSITY

# 特征处理

时林

2022年9月2日

# 特征处理的相关定义

## 降维

在机器学习和统计学领域，降维是指在某些限定条件下，降低随机变量个数，得到一组“不相关”主变量的过程

有效信息的提取综合及无用信息的摒弃。

## 特征选择

从原来的特征中选择出子集。这里的特征只是被选择出来，性质和原来的特征是一致的。

## 特征提取

通过原来存在的特征的集合创造一个新的特征子集。

# 特征处理对比

## 特征选择

优点　与单个特征相关的重要信息不会丢失

缺点　如果需要一组小的特征，并且原始特征非常多样，则有可能丢失信息

## 特征提取

优点　在不丢失原始特征空间信息的情况下减小特征空间的大小

缺点　原始特征的线性组合通常是不可解释的，关于原始特征贡献多少的信息常常丢失

# 特征选择

**特征选择必要条件**　　特征发散　　特征与目标的相关性

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

## 特征选择方法——过滤法（Filter）

过滤法就是按照发散性或者相关性对各个特征进行评分，设定阈值或者选择阈值的个数，完成特征选择。

方差法：这种方法通过计算每个特征的均值和方差，设定一个基础阈值，当该维度的特征方差小于基础阈值时，则丢弃该特征。

单变量：单变量特征选择能够对每一个特征进行测试，衡量该特征和响应变量之间的关系，根据得分扔掉不好的特征。

卡方检验：对于回归和分类问题可以采用卡方检验等方式对特征进行测试（检验独立性）。

互信息法：互信息可以看成是一个随机变量中包含的关于另一个随机变量的信息量。

# 特征选择

## 特征选择方法——包裹法（Wrapper）

包裹法就是选择特定算法，然后根据算法效果来选择特征集合。就是通过不断的启发式方法来搜索特征，主要分为如下两类。

- 增加法
- 减少法

✓ RF选取重要性特征

- 平均不纯度减少（MDI）
- 平均精确度减少（MDA）

具体的方法就是：

1. 对于每一棵决策树，用OOB 计算袋外数据误差，记为 errOOB1；
2. 然后随机对OOB所有样本的特征i加入噪声干扰，再次计算袋外数据误差，记为errOOB2；
3. 假设有N棵树，特征i的重要性为sum(errOOB2-errOOB1)/N;

# 特征选择

## 特征选择方法——包裹法（Wrapper）

✓ 梯度提升树（GBDT）选取重要性特征

主要是通过计算特征i在单棵树中重要度的平均值，计算公式如下：

$$\hat{J}_j^2 = \frac{1}{M}\sum_{m=1}^{M}\hat{J}_j^2(T_m)$$

其中，M是树的数量。特征i在单棵树的重要度主要是通过计算按这个特征i分裂之后损失的减少值。

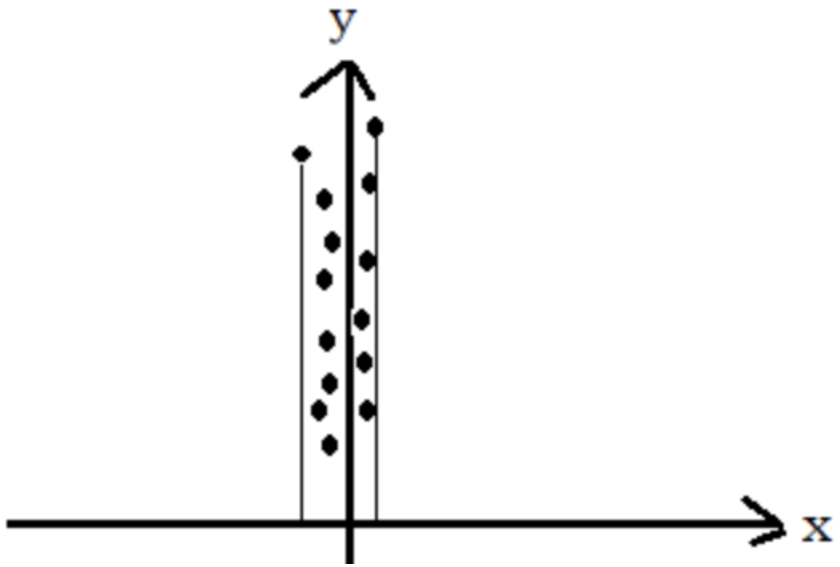
# 特征选择

## 特征选择方法——嵌入法（Embedded）

就是利用正则化的思想，将部分特征属性的权重调整到0，则这个特性相当于就是被舍弃了。（其实就是在损失函数上再加入正则项，不断的利用梯度下降极小化损失函数，调整一些特征的权重，有些权重变为0了则相当于被舍弃了，没被舍弃的相当于被选择出来的向量）

# 特征选择

**特征提取方法——主成分分析（PCA）：** 将数据从**原始的空间**中转换到**新的特征空间**中

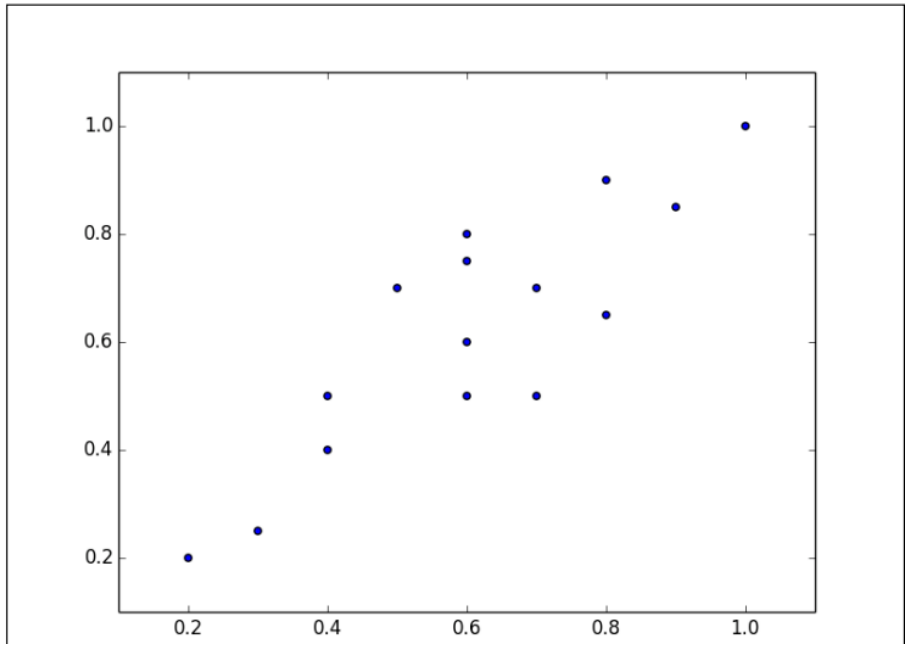


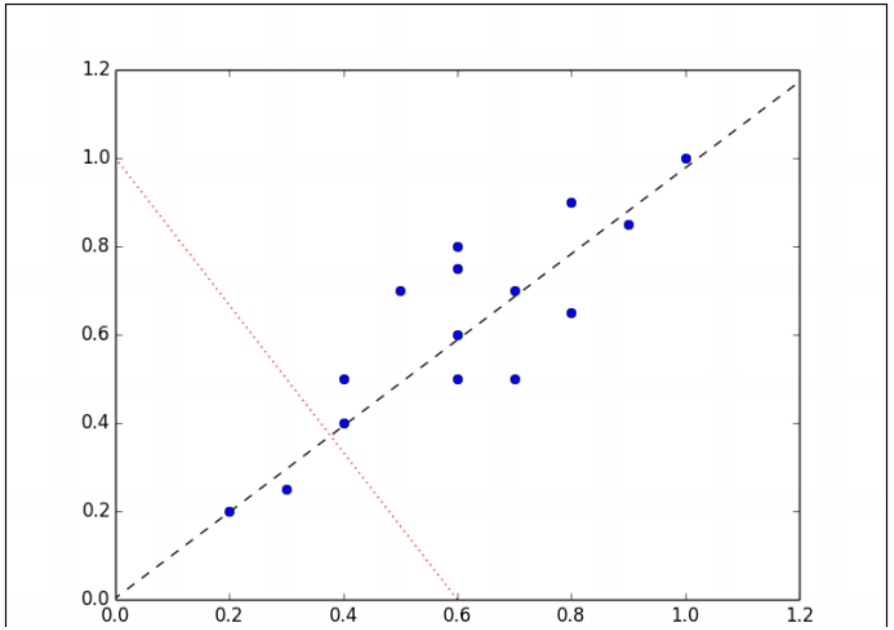

数据在x轴的投影在0附近，我们可以舍去x轴，只用数据在y轴的投影来表示数据。

# 特征选择

**特征提取方法——主成分分析（PCA）：** 将数据从**原始的空间**中转换到**新的特征空间**中

PCA可以将高维数据集映射到低维空间的同时，尽可能的保留更多变量。PCA旋转数据集与其主成分对齐，将最多的变量保留到第一主成分中。

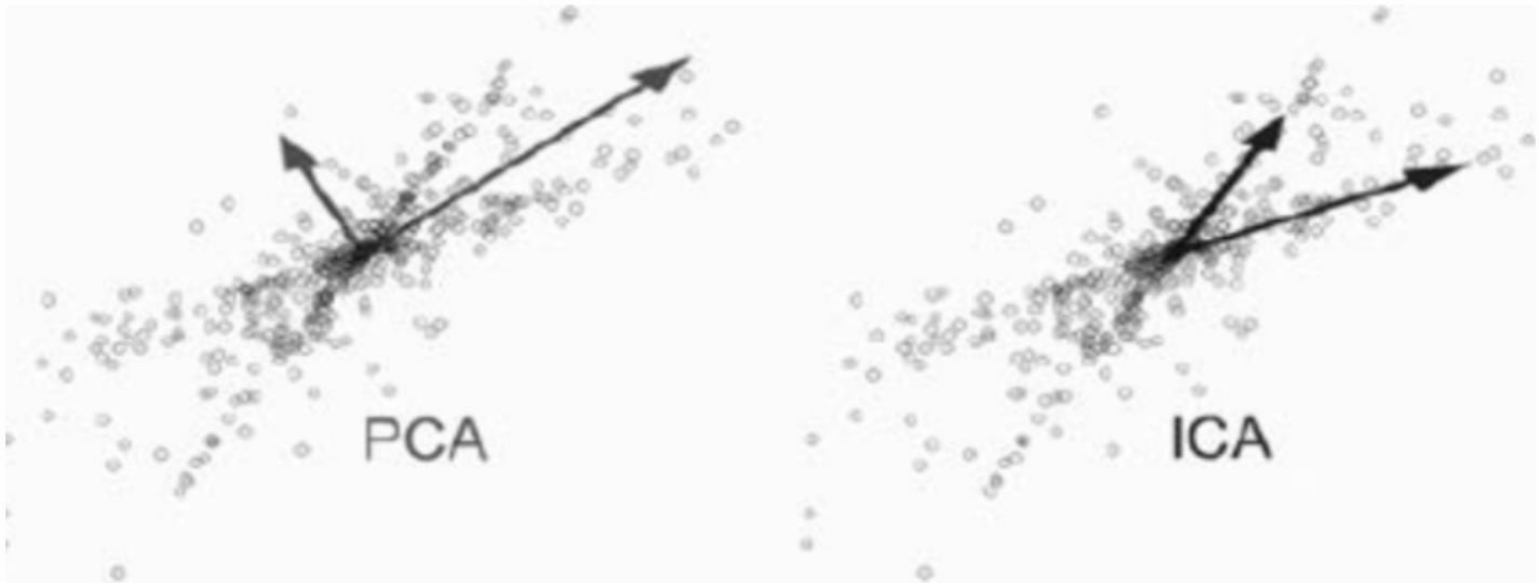


# 特征选择

**特征提取方法——独立成分分析（ICA）：**寻找最能使数据的相互独立的方向

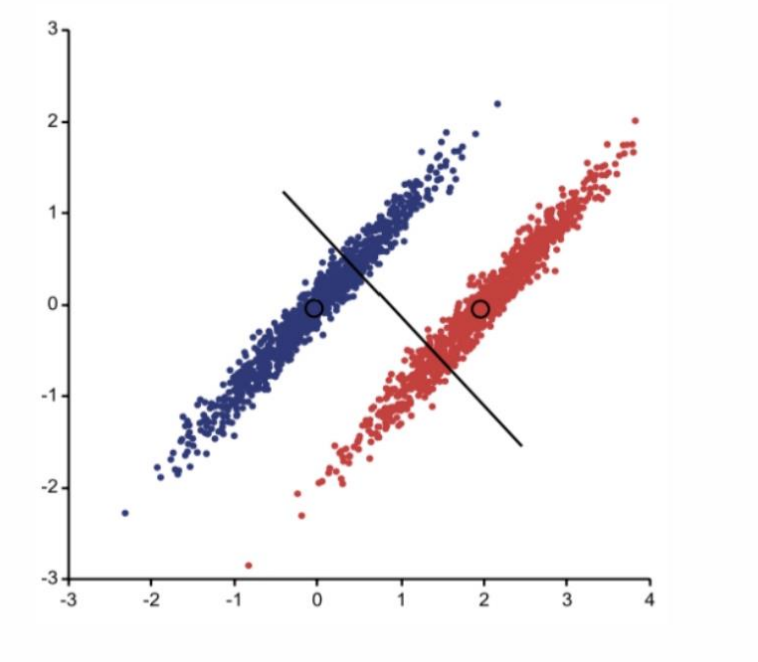
PCA仅要求方向是不相关的。咱们知道，独立能够推出不相关，反之则不能够，而高斯分布的状况下独立等价于不相关。



# 特征选择

**特征提取方法——线性判别分析（LDA）：**从更利于分类的角度来降维

利用到了训练样本的类别标记，追求的是最可以分开各个类别数据的投影方法。

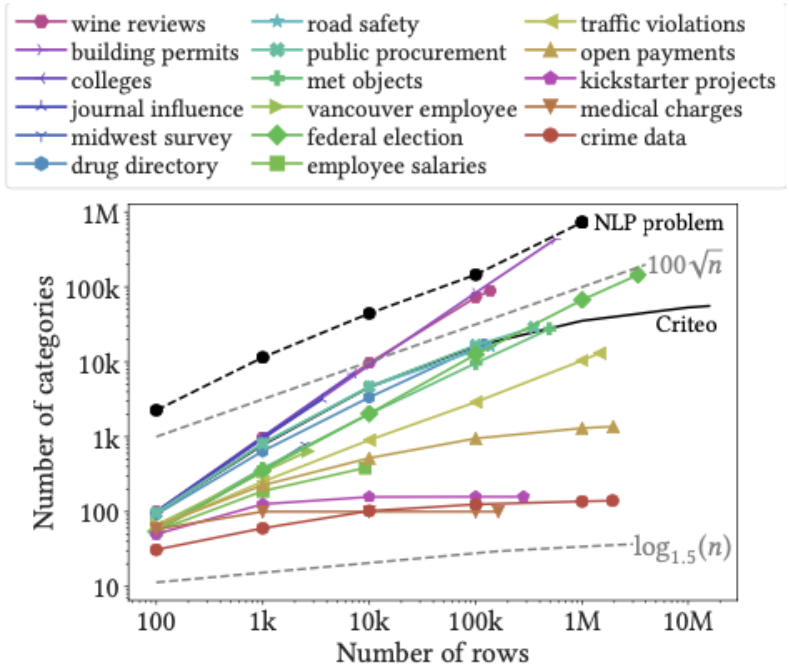


# 高基数数据

## 高基数数据特性

高基数（High-Cardinality）的定义为在一个数据列中的数据基本上不重复，或者说重复率非常低。

例如我们常见的识别号，邮件地址，用户名等都可以被认为是高基数数据。 例如我们常定义的 USERS 数据表中的 USER_ID 字段，这个字段中的数据通常被定义为 1 到 n。




低重复性

无限性

不确定性

# 高基数数据特征提取

One-hot

从统计分析的角度来看，条目与相关信息的倍增具有挑战性，原因有两个：

- 丢失了原始信息的相关内容
- 当类别数量增加时，该方法会崩溃，因为它会创建高维特征向量。

此外，一个热编码不能将特征向量分配给可能出现在测试集中的新类别，即使其表示接近训练集中的一个。因此，一种热编码不适合在线学习设置：如果新类别到来，则必须重新计算数据集的整个编码，并且特征向量的维数变得无界。

# 高基数数据特征提取

## 字符串热编码

对于由字符串表示的类别变量，相似性编码通过考虑类别对之间的字符串相似性度量扩展了一种热编码。

对应于给定训练数据集的第i个样本的类别。给定字符串相似性sim：

$$\mathrm{sim}_{\text{one-hot}}(s_i, s_j) = \mathbb{1}[s_i = s_j],$$

## n-grams编码字符串

捕获字符串中形态的一种简单方法是通过其字符或单词n-gram的计数来表征它。这有时被称为字符串的n-grams表征。

# 高基数数据特征提取

## 词嵌入

词嵌入实际上是一类技术，单个词在预定义的向量空间中被表示为实数向量，每个单词都映射到一个向量。

神经网络的词向量基于分布式表达方式，核心是上下文的表示以及上下文与目标词之间的关系映射。

### word2vec

可以把对文本内容的处理简化为k维向量空间中的向量运算，而向量空间上的相似度可以用来表示文本语义上的相似度，因此word2vec输出的词向量是一个基础性的工作，比如聚类、同义词、词性分析等。

# 高基数数据特征提取

寻求高基数字符串分类变量的低维编码：

可扩展到许多类别　可向最终用户解释　　便于统计分析

- 最小散列编码器（min-hash encoder）
- 伽马-泊松矩阵分解（Gamma-Poisson matrix factorization）

# 高基数数据特征提取

## min-hash

LSH（局部敏感哈希 Locality Sensitive Hashing）

LSH算法大致分为三个步骤：

1.Shingling:将文本文档转换为集合表示 (通常是转换为布尔型向量）

2.Min-Hashing: 将高维度的向量转换为低维的哈希签名，此时再计算哈希签名的相似性

3.Locality-Sensitive Hashing: 重点关注来自相似文档的一对候选哈希签名

min-hash 算法是LSH算法中的一个步骤，其主要工作是对输入的高维向量（可能是几百万维甚至更高）转换为低维的向量（降维后的向量被称作数字签名），然后再对低维向量计算其相似，以达到降低计算成本，提高运行效率的目的。

# 高基数数据特征提取

## min-hash

Jaccard距离

Jaccrad距离是度量集合相似度的方法之一，其基本公式如下：

$$Jaccard(C_i, C_j) = \frac{|C_i \bigcap C_j|}{|C_i \bigcup C_j|}$$

前文中所提及的“集合”（也即公式中的Ci、Cj），你可以将其视为一个矩阵中的列，而行则代表集合中的元素：

| $C_1$ | $C_2$ |
|---|---|
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |
| 0 | 0 |
| 1 | 1 |
| 0 | 1 |

$$Jaccard(C_1, C_2) = 2/5 = 0.4$$

# 高基数数据特征提取

## min-hash

min-hash算法就是一个在Jaccard距离基础之上进行改进，带有降维功能的进阶版Jaccard距离：

$$minhash(C_i, C_j) = Jaccard(sig(C_i), sig(C_j))$$

min-hash算法的公式

凭什么认为完成了最小哈希操作后的集合之间的Jaccard距离与原始集合之间的Jaccard距离还是相等的呢？

# 高基数数据特征提取

## min-hash

最小哈希操作

想象一个由我们需要对比的原始集合**按列组成**的矩阵，将这个矩阵**按行打乱**排序P次，在每一次打乱之后，找出每个集合（即矩阵的列向量）的**第一个**值为1的行索引，并将这个索引分别填充进一个新的集合中（每个原始集合拥有各自的一个新的集合），这个新的集合就是原始集合的哈希签名。用公式表达如下：

$$sig(C_i) = P个C_i列每次打乱后第一个值为1的行索引值$$

1.假设有三个需要对比的（5，）原始集合Ci，首先将他们按列组合成一个（5，3）的矩阵，其中Ri表示行号：

| | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| $R_1$ | 1 | 0 | 1 |
| $R_2$ | 0 | 1 | 1 |
| $R_3$ | 1 | 0 | 0 |
| $R_4$ | 1 | 0 | 1 |
| $R_5$ | 0 | 1 | 0 |

# 高基数数据特征提取

## min-hash

2.假设我们希望将之降维至3维，即将原始集合Ci降维至（3，），则将P设为3，即进行三次按行乱序：

| | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| $R_1$ | 1 | 0 | 1 |
| $R_2$ | 0 | 1 | 1 |
| $R_3$ | 1 | 0 | 0 |
| $R_4$ | 1 | 0 | 1 |
| $R_5$ | 0 | 1 | 0 |

**Signatures**

| | $S_1$ | $S_2$ | $S_3$ |
|---|---|---|---|
| Perm 1 = (12345) | 1 | 2 | 1 |
| Perm 2 = (54321) | 4 | 5 | 4 |
| Perm 3 = (34512) | 3 | 5 | 4 |

3.最后，我们计算对Ci进行min-hash操作后得到的数字签名Si的Jacarrd系数，得到集合之间的相似度（下图Col_Col是原始集合的Jaccard系数，Sig-Sig是哈希签名的Jaccard系数，也就是原始集合的min-hash系数）：

**Similarities**

| | 1-2 | 1-3 | 2-3 |
|---|---|---|---|
| Col-Col | 0.00 | 0.50 | 0.25 |
| Sig-Sig | 0.00 | 0.67 | 0.00 |

# 高基数数据特征提取

## min-hash

**凭什么认为完成了最小哈希操作后的集合之间的Jaccard距离与原始集合之间的Jaccard距离还是相等的呢？**

现仅考虑集合C1和C2，那么这两列集合的相同的每一行会有下面3种情况：

| $C_1$ | $C_2$ |
|---|---|
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |
| 0 | 0 |
| 1 | 1 |
| 0 | 1 |

1.C1和C2在该行的值都为1，我们将这种情况的次数记为X

2.C1和C2的其中一个在该行的值为1，另一个值为0，我们将这种情况的次数记为记为Y

3.C1和C2的值都为0，这种情况的次数记为Z

第一种情况X代表了集合C1和C2的交集，X+Y则代表了C1和C2的并集，即此时Jaccard(C1,C2) = X/(X+Y)

接下来计算min-hash(C1,C2) 。经过随机行打乱后，从上往下扫描，在碰到Y行之前碰到X行的概率为X/(X+Y)。

$$sig(C_i) = P个C_i列每次打乱后第一个值为1的行索引值$$

# 高基数数据特征提取

## 伽马-泊松模型

**伽马-泊松模型最初是为了在给定文档的字数表示的情况下，找到文档的低维表示，即主题。**

我们这里考虑子字符串的表示方法：

$$\mathbf{f} \approx \mathbf{x}\mathbf{\Lambda},$$

向量f描述可能存在的字符串。

$x \in R^d$ 是分解的激活（方式），$\Lambda$ 是文本中的主题矩阵。

给定一个具有n个样本的训练数据集，模型通过分解数据的bag-of-n-grams表示F来估计未知原型∧：

$$\mathbf{F} \approx \mathbf{X}\mathbf{\Lambda}, \quad \text{with } \mathbf{X} \in \mathbb{R}^{n \times d}, \mathbf{\Lambda} \in \mathbb{R}^{d \times m}$$

# THANKS