**《特征工程》**

**First draft**

目录

[**前言 1**](#_Toc482107685)

[**简介、特征工程 2**](#_Toc482107686)

[**一、数据清洗和采样 3**](#_Toc482107687)

[**（一）数据清洗 3**](#_Toc482107688)

[**（二）采样 6**](#_Toc482107689)

[**二、数据预处理 8**](#_Toc482107690)

[**（一）特征工程的数据类型及其特点 8**](#_Toc482107691)

[**（二）无量纲化 11**](#_Toc482107692)

[**（三）对定量特征二值化 12**](#_Toc482107693)

[**（四）对定性特征哑编码 12**](#_Toc482107694)

[**（五）数据变换 12**](#_Toc482107695)

[**三、数据降维 13**](#_Toc482107696)

[**（一）PCA（主成分分析算法） 13**](#_Toc482107697)

[**（二）LDA（线性判别式分析） 13**](#_Toc482107698)

[**（三）LLE（局部线性嵌入） 13**](#_Toc482107699)

[**（四）Laplacian Eigenmaps（拉普拉斯特征映射） 14**](#_Toc482107700)

[**四、特征选择 15**](#_Toc482107701)

[**（一）特征选择 15**](#_Toc482107702)

[**（二）、过滤法(Filter) 16**](#_Toc482107703)

[**（三）、包装法（Wrapper） 18**](#_Toc482107704)

[**（四）、嵌入法（Embedded） 19**](#_Toc482107705)

[**（五）过滤法和包装法的不同 21**](#_Toc482107706)

[**写在后面的话 22**](#_Toc482107707)

[**参考资料 23**](#_Toc482107708)

# 前言

大数据”是近年来IT行业的热词,大数据在各个行业的应用逐渐变得广泛起来，如何从海量的原始数据发现其中有用的特征为人们所用，一直是各个公司技术部门乐此不疲、津津乐道的话题。

有这么一句话在业界广泛流传：数据和特征决定了机器学习的上限，而模型和算法只是逼近这个上限而已。因为好的特征具有更强的灵活性，即使使用简单的模型做训练，就可以得到优秀的结果。“工欲善其事，必先利其器”，特征工程可以理解为利其器的过程。互联网公司里大部分复杂的模型都是极少数的数据科学家在做，大多数工程师们做的事情基本是在数据仓库里搬砖，不断地数据清洗，再一个是分析业务不断地找特征。

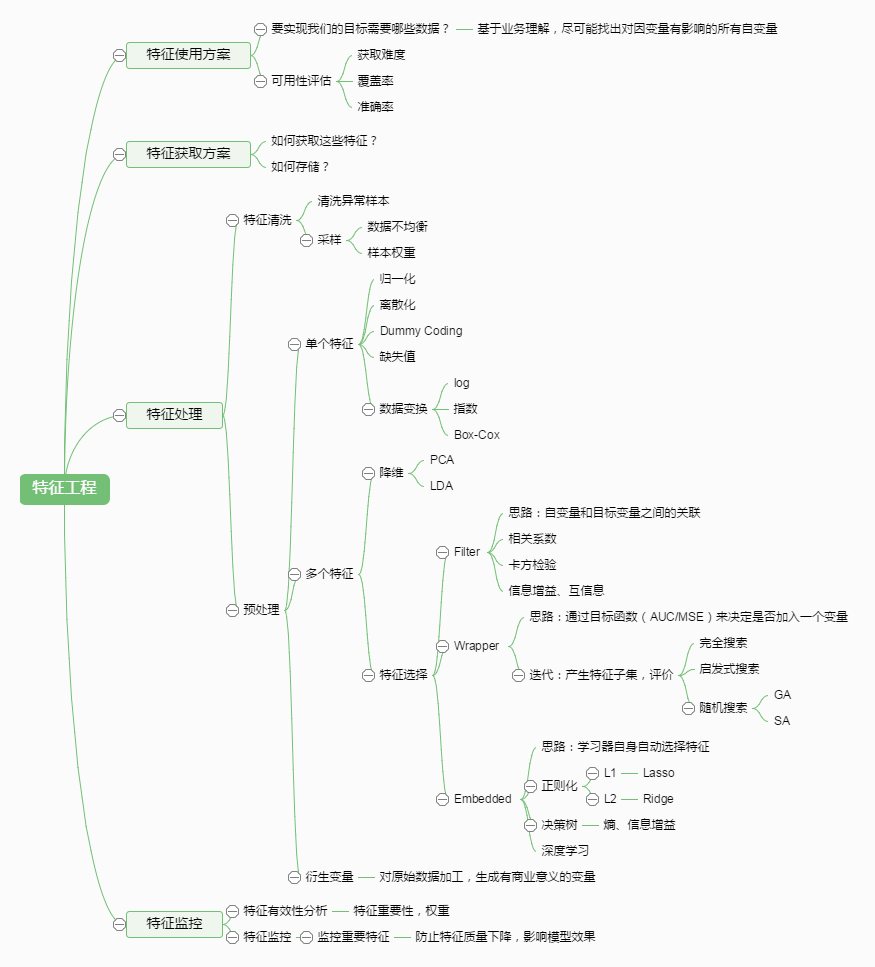
机器学习领域的大神Andrew Ng(吴恩达)老师曾说 “Coming up with features is difficult, time-consuming, requires expert knowledge. ‘Applied machine learning’ is basically feature engineering. ”表达了特征工程是比较难，比较耗时的一个过程，需要较强的领域知识。我们也了解到，Kaggle比赛和天池比赛的冠军其实在比赛中并没有用到很高深的算法，大多数都是在特征工程这个环节做出了出色的工作，然后使用一些常见的算法，比如逻辑回归，就能得到出色的性能。可见特征工程在实际的实验研究中占着重要的位置，那么这样一个耗时又重要的工作解决的是什么问题呢？这篇文档也许会帮助你比较好的认识和理解特征工程。

# 简介、特征工程

什么是特征？特征是数据中抽取出来的对结果预测有用的信息，可以是文本或者数据。人工选取出来的特征依赖人力和专业知识，不利于推广。于是我们需要通过机器来学习和抽取特征，促进特征工程的工作更加快速、有效。

什么是特征工程？使用专业背景知识和技巧处理数据，使得特征能在机器学习算法上发挥更好的作用的过程。过程包含了特征提取、特征构建、特征选择等模块。

特征工程的目的是筛选出更好的特征，获取更好的训练数据。通过总结和归纳，人们认为特征工程包括以下方面：



# 一、数据清洗和采样

数据清洗和采样， 是整个数据分析过程中不可缺少的一个环节，其结果质量直接关系到模型效果和最终结论。在实际操作中，数据清洗通常会占据分析过程的50%—80%的时间。通过数据清洗【1，2，4，5】和采样【3】可使数据获得用于分析的正确形状（shape）和质量（quality）。

## （一）数据清洗

在整个数据清洗过程中主要分为预处理阶段、缺失值清洗、格式内容清洗、逻辑错误清洗、非需求数据清洗、关联性验证等过程，如图1所示。（注：这篇文档所述内容是参考某文档，比较浅层次的梳理一下数据清洗的过程，如果有更高的需求和希望能够获知更多内容、细节请参考文档末所归纳的参考资料。）

1. **预处理阶段**

预处理阶段主要做两件事情：

一是将数据导入处理工具。通常来说，建议使用数据库，单机跑数搭建MySQL环境即可。如果数据量大(千万级以上)，可以使用文本文件存储+Python操作的方式。

二是看数据。这里包含两个部分：一是看元数据，包括字段解释、数据来源、代码表等等一切描述数据的信息;二是抽取一部分数据，使用人工查看方式，对数据本身有一个直观的了解，并且初步发现一些问题，为之后的处理做准备。

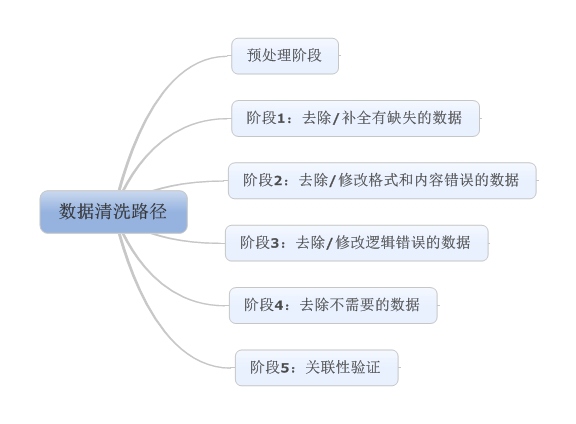


图1 数据清洗流程图

1. **缺失值清洗**

缺失值是最常见的数据问题，处理缺失值也有很多方法，通常按照以下四个步骤进行：

1. 确定缺失值范围：对每个字段都计算其缺失值比例，然后按照缺失比例和字段重要性，分别制定策略，可用下图2表示：

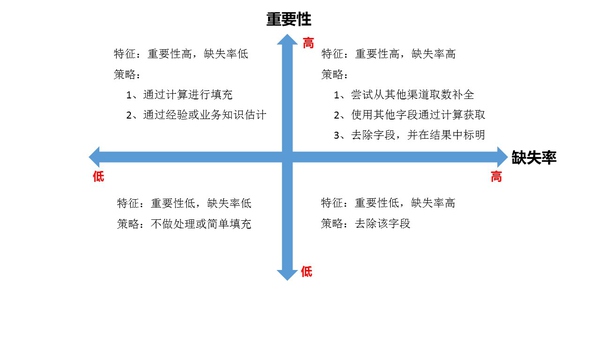


图2 重要性-缺失率及对应使用的策略图

B. 去除不需要的字段：这一步很简单，直接删掉即可……但强烈建议清洗每做一步都备份一下，或者在小规模数据上试验成功再处理全量数据，不然删错了会追悔莫及(多说一句，写SQL的时候delete一定要配where!)。

C. 填充缺失内容：某些缺失值可以进行填充，方法有以下三种：

* 以业务知识或经验推测填充缺失值
* 以同一指标的计算结果(均值、中位数、众数等)填充缺失值
* 以不同指标的计算结果填充缺失值

前两种方法比较好理解。关于第三种方法，举个最简单的例子：年龄字段缺失，但是有屏蔽后六位的身份证号，so……

D. 重新取数：如果某些指标非常重要又缺失率高，那就需要和取数人员或业务人员了解，是否有其他渠道可以取到相关数据。

以上，简单的梳理了缺失值清洗的步骤，但其中有一些内容远比我说的复杂，比如填充缺失值。很多讲统计方法或统计工具的书籍会提到相关方法，有兴趣的各位可以自行深入了解。

1. **格式内容清洗**

如果数据是由系统日志而来，那么通常在格式和内容方面，会与元数据的描述一致。而如果数据是由人工收集或用户填写而来，则有很大可能性在格式和内容上存在一些问题，简单来说，格式内容问题有以下几类：

A.时间、日期、数值、全半角等显示格式不一致

这种问题通常与输入端有关，在整合多来源数据时也有可能遇到，将其处理成一致的某种格式即可。

B.内容中有不该存在的字符

某些内容可能只包括一部分字符，比如身份证号是数字+字母，中国人姓名是汉字(赵C这种情况还是少数)。最典型的就是头、尾、中间的空格，也可能出现姓名中存在数字符号、身份证号中出现汉字等问题。这种情况下，需要以半自动校验半人工方式来找出可能存在的问题，并去除不需要的字符。

C.内容与该字段应有内容不符

姓名写了性别，身份证号写了手机号等等，均属这种问题。 但该问题特殊性在于：并不能简单的以删除来处理，因为成因有可能是人工填写错误，也有可能是前端没有校验，还有可能是导入数据时部分或全部存在列没有对齐的问题，因此要详细识别问题类型。

格式内容问题是比较细节的问题，但很多分析失误都是栽在这个坑上，比如跨表关联或VLOOKUP失败(多个空格导致工具认为“陈丹奕”和“陈 丹奕”不是一个人)、统计值不全(数字里掺个字母当然求和时结果有问题)、模型输出失败或效果不好(数据对错列了，把日期和年龄混了，so……)。因此，请各位务必注意这部分清洗工作，尤其是在处理的数据是人工收集而来，或者你确定产品前端校验设计不太好的时候……

1. **逻辑错误清洗**

这部分的工作是去掉一些使用简单逻辑推理就可以直接发现问题的数据，防止分析结果走偏。主要包含以下几个步骤：

A.去重

有的分析师喜欢把去重放在第一步，但我强烈建议把去重放在格式内容清洗之后，原因已经说过了(多个空格导致工具认为“陈丹奕”和“陈 丹奕”不是一个人，去重失败)。而且，并不是所有的重复都能这么简单的去掉……

有人曾经做过电话销售相关的数据分析，发现销售们为了抢单简直无所不用其极……举例，一家公司叫做“ABC管家有限公司“，在销售A手里，然后销售B为了抢这个客户，在系统里录入一个”ABC官家有限公司“。你看，不仔细看你都看不出两者的区别，而且就算看出来了，你能保证没有”ABC官家有限公司“这种东西的存在么。这种时候，要么去抱RD大腿要求人家给你写模糊匹配算法，要么肉眼看吧。上边这个还不是最狠的，请看下图3：



图3 “八里庄路”。你用的系统里很有可能两条路都叫八里庄路，敢直接去重不?(附送去重小tips：两个八里庄路的门牌号范围不一样)

当然，如果数据不是人工录入的，那么简单去重即可。

B.去除不合理值

一句话就能说清楚：有人填表时候瞎填，年龄200岁，年收入100000万(估计是没看见”万“字)，这种的就要么删掉，要么按缺失值处理。这种值如何发现?提示：可用但不限于箱形图(Box-plot).

C.修正矛盾内容

有些字段是可以互相验证的，举例：身份证号是1101031980XXXXXXXX，然后年龄填18岁，我们虽然理解人家永远18岁的想法，但得知真实年龄可以给用户提供更好的服务啊(又瞎扯……)。在这种时候，需要根据字段的数据来源，来判定哪个字段提供的信息更为可靠，去除或重构不可靠的字段。

逻辑错误除了以上列举的情况，还有很多未列举的情况，在实际操作中要酌情处理。另外，这一步骤在之后的数据分析建模过程中有可能重复，因为即使问题很简单，也并非所有问题都能够一次找出，我们能做的是使用工具和方法，尽量减少问题出现的可能性，使分析过程更为高效。

1. **非需求数据清洗**

这一步说起来非常简单：把不要的字段删了。但实际操作起来，有很多问题，例如：把看上去不需要但实际上对业务很重要的字段删了;某个字段觉得有用，但又没想好怎么用，不知道是否该删;一时看走眼，删错字段了。前两种情况我给的建议是：如果数据量没有大到不删字段就没办法处理的程度，那么能不删的字段尽量不删。第三种情况，请勤备份数据……

1. **关联性验证**

如果你的数据有多个来源，那么有必要进行关联性验证。例如，你有汽车的线下购买信息，也有电话客服问卷信息，两者通过姓名和手机号关联，那么要看一下，同一个人线下登记的车辆信息和线上问卷问出来的车辆信息是不是同一辆，如果不是(别笑，业务流程设计不好是有可能出现这种问题的!)，那么需要调整或去除数据。严格意义上来说，这已经脱离数据清洗的范畴了，而且关联数据变动在数据库模型中就应该涉及。但我还是希望提醒大家，多个来源的数据整合是非常复杂的工作，一定要注意数据之间的关联性，尽量在分析过程中不要出现数据之间互相矛盾，而你却毫无察觉的情况。

## （二）采样

这里所述的采样【3】主要是在数据挖掘过程遇到非平衡分类数据集时所采用的一种处理方式。

针对在对不平衡的分类数据集进行建模时，机器学习算法可能并不稳定，其预测结果甚至可能是有偏的，而预测精度此时也变得带有误导性。

总的说来，针对不平衡数据采用的这些采样方法都是为了把不平衡数据修正为平衡数据。修正方法就是调整原始数据集的样本量，使得不同类的数据比例一致。而在诸多学者研究得出基于平衡数据的模型整体更优的结论后，这一类方法越来越受到分析师们的青睐。下列是一些具体的处理方法名称：

* 欠采样法（Undersampling）：该方法主要是对大类进行处理。它会减少大类的观测数来使得数据集平衡。这一办法在数据集整体很大时较为适宜，它还可以通过降低训练样本量来减少计算时间和存储开销。欠采样法共有两类：随机（Random）的和有信息的（Informative）。
* 过采样法（Oversampling）：这一方法针对小类进行处理。它会以重复小类的观测的方式来平衡数据。该方法也被称作升采样（Upsampling）。和欠采样类似，它也能分为随机过采样和有信息的过采样两类。
* 人工数据合成法（Synthetic Data Generation)：人工数据合成法是利用生成人工数据而不是重复原始观测来解决不平衡性。它也是一种过采样技术。在这一领域，SMOTE法（Synthetic Minority Oversampling Technique）是有效而常用的方法。该算法基于特征空间（而不是数据空间）生成与小类观测相似的新数据，为了生成人工数据，我们需要利用自助法（Bootstrapping）和K近邻法（K-neraest neighbors）。详细步骤如下：
* 计算样本点间的距离并确定其近邻。
* 生成一个0到1上的均匀随机数，并将其乘以距离。
* 把第二步生成的值加到样本点的特征向量上。
* 这一过程等价于在在两个样本的连线上随机选择了一个点。
* 代价敏感学习法（Cose Sensitive Learning）：该方法会衡量误分类观测的代价来解决不平衡问题。这方法不会生成平衡的数据集，而是通过生成代价矩阵来解决不平衡问题。代价矩阵是描述特定场景下误分类观测带来的损失的工具。

当我们面对不平衡数据集时，我们常常发现利用采样法修正的效果不错。但在本例中，人工数据合成比传统的采样法更好。为了得到更好的结果，你可以使用一些更前沿的方法，诸如基于boosting 的人工数据合成。当然，如果不采用采样法，在调用目前的许多算法工具包中也可以在通过改变正负样本的权重上完成分类或者回归的目的。

# 二、数据预处理

通过特征提取，我们能得到未经处理的特征，这时的特征可能有以下问题：

不属于同一量纲： 即特征的规格不一样，不能够放在一起比较。无量纲化可以解决这一问题。

信息冗余：对于某些定量特征，其包含的有效信息为区间划分，例如学习成绩，假若只关心“及格”或不“及格”，那么需要将定量的考分，转换成“1”和“0”表示及格和未及格。二值化可以解决这一问题。

定性特征不能直接使用：某些机器学习算法和模型只能接受定量特征的输入，那么需要将定性特征转换为定量特征。最简单的方式是为每一种定性值指定一个定量值，但是这种方式过于灵活，增加了调参的工作。[通常使用哑编码的方式将定性特征转换为定量特征](http://www.ats.ucla.edu/stat/mult_pkg/faq/general/dummy.htm)：假设有N种定性值，则将这一个特征扩展为N种特征，当原始特征值为第i种定性值时，第i个扩展特征赋值为1，其他扩展特征赋值为0。哑编码的方式相比直接指定的方式，不用增加调参的工作，对于线性模型来说，使用哑编码后的特征可达到非线性的效果。

信息利用率低：不同的机器学习算法和模型对数据中信息的利用是不同的，之前提到在线性模型中，使用对定性特征哑编码可以达到非线性的效果。类似地，对定量变量多项式化，或者进行其他的转换，都能达到非线性的效果。

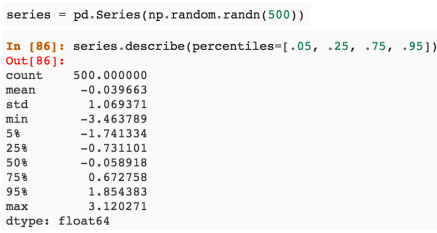
我们使用sklearn中的preproccessing库来进行数据预处理，可以覆盖以上问题的解决方案。

## （一）特征工程的数据类型及其特点

**1、数值型**

幅度调整/归一化：python中会有一些函数比如preprocessing.MinMaxScaler()将幅度调整到 [0,1] 区间。

统计值：包括max, min, mean, std等。python中用pandas库序列化数据后，可以得到数据的统计值。



离散化：把连续值转成非线性数据。例如电商会有各种连续的价格表，从0.03到100元，假如以一元钱的间距分割成99个区间，用99维的向量代表每一个价格所处的区间，1.2元和1.6元的向量都是 [0,1,0,…,0]。pd.cut() 可以直接把数据分成若干段。

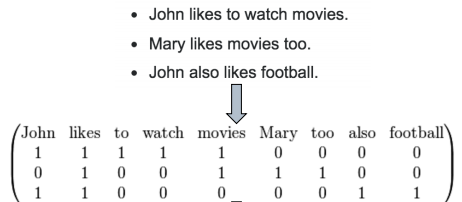
柱状分布：离散化后统计每个区间的个数做柱状图。

**2、类别型**

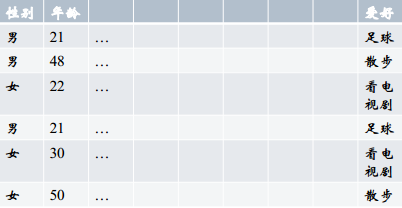
类别型一般是文本信息，比如颜色是红色、黄色还是蓝色，我们存储数据的时候就需要先处理数据。处理方法有：

a. one-hot编码，编码后得到哑变量。统计这个特征上有多少类，就设置几维的向量，pd.get\_dummies()可以进行one-hot编码。

b. Hash编码成词向量：



c. Histogram映射：把每一列的特征拿出来，根据target内容做统计，把target中的每个内容对应的百分比填到对应的向量的位置。优点是把两个特征联系起来。



上表中，我们来统计“性别与爱好的关系”，性别有“男”、“女”，爱好有三种，表示成向量 [散步、足球、看电视剧]，分别计算男性和女性中每个爱好的比例得到：男[1/3, 2/3, 0]，女[0, 1/3, 2/3]。即反映了两个特征的关系。

**3、时间型**

时间型特征的用处特别大，既可以看做连续值（持续时间、间隔时间），也可以看做离散值（星期几、几月份）。数据挖掘中经常会用时间作为重要特征，比如电商可以分析节假日和购物的关系，一天中用户喜好的购物时间等。

**4、文本型**

a.词袋：文本数据预处理后，去掉停用词，剩下的词组成的list，在词库中的映射稀疏向量。Python中用CountVectorizer处理词袋．

b. 把词袋中的词扩充到n-gram：n-gram代表n个词的组合。比如“我喜欢你”、“你喜欢我”这两句话如果用词袋表示的话，分词后包含相同的三个词，组成一样的向量：“我 喜欢 你”。显然两句话不是同一个意思，用n-gram可以解决这个问题。如果用2-gram，那么“我喜欢你”的向量中会加上“我喜欢”和“喜欢你”，“你喜欢我”的向量中会加上“你喜欢”和“喜欢我”。这样就区分开来了。

c. 使用TF-IDF特征：TF-IDF是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。TF(t) = (词t在当前文中出现次数) / (t在全部文档中出现次数)，IDF(t) = ln(总文档数/ 含t的文档数)，TF-IDF权重 = TF(t) \* IDF(t)。自然语言处理中经常会用到。

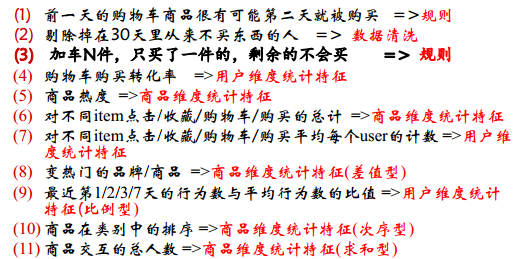
**5、统计型**

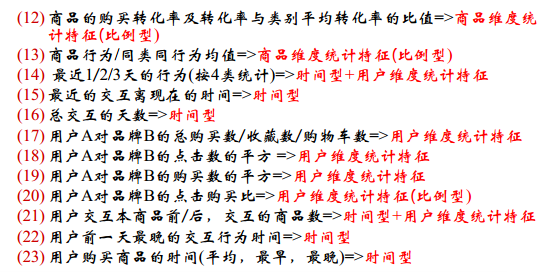
历届的Kaggle/天池比赛， 天猫/京东排序和推荐业务线里模型用到的特征。统计的内容包括加减平均、分位线、次序型、比例类等。

比如“天池大数据之移动推荐算法大赛”中，给比赛选手两张表，介绍用户和商品信息，要求预测把哪些商品推荐给用户，用户最有可能购买。下面是两张表的内容：



　　下面举例说明选手们是怎么进行特征处理的：





　可见，选手需要进行制定规则、数据清洗、各个种类的特征处理等，对特征的研究是非常细化的。

**6、组合特征**

a. 拼接型：简单的组合特征。例如挖掘用户对某种类型的喜爱，对用户和类型做拼接。正负权重，代表喜欢或不喜欢某种类型。

- user\_id&&category: 10001&&女裙 10002&&男士牛仔

- user\_id&&style: 10001&&蕾丝 10002&&全棉

b. 模型特征组合：

- 用GBDT产出特征组合路径

- 组合特征和原始特征一起放进LR训练

## （二）无量纲化

无量纲化使不同规格的数据转换到同一规格。常见的无量纲化方法有标准化和区间缩放法。标准化的前提是特征值服从正态分布，标准化后，其转换成标准正态分布。区间缩放法利用了边界值信息，将特征的取值区间缩放到某个特点的范围，例如[0, 1]等。

**1 、标准化**

标准化需要计算特征的均值和标准差，公式表达为：

C:\Users\dell\Desktop\2017年4月特征工程\927391-20160502113957732-1062097580.png

**2 、区间缩放法**

区间缩放法的思路有多种，常见的一种为利用两个最值进行缩放，公式表达为：

C:\Users\dell\Desktop\2017年4月特征工程\927391-20160502113301013-1555489078.png

**3 、标准化与归一化的区别**

简单来说，标准化是依照特征矩阵的列处理数据，其通过求z-score的方法，将样本的特征值转换到同一量纲下。归一化是依照特征矩阵的行处理数据，其目的在于样本向量在点乘运算或其他核函数计算相似性时，拥有统一的标准，也就是说都转化为“单位向量”。规则为l2的归一化公式如下：

C:\Users\dell\Desktop\2017年4月特征工程\927391-20160719002904919-1602367496.png

## （三）对定量特征二值化

定量特征二值化的核心在于设定一个阈值，大于阈值的赋值为1，小于等于阈值的赋值为0，公式表达如下：

C:\Users\dell\Desktop\2017年4月特征工程\927391-20160502115121216-456946808.png

## （四）对定性特征哑编码

## （五）数据变换

常见的数据变换有基于多项式的、基于指数函数的、基于对数函数的。4个特征，度为2的多项式转换公式如下：

C:\Users\dell\Desktop\2017年4月特征工程\927391-20160502134944451-270339895.png

# 三、数据降维

数据降维【6-8】，分为线性方法降维和非线性方法降维，其中常见的几种降维方法有PCA（主成分分析算法）、LDA（线性判别式分析）、LLE（局部线性嵌入）、Laplacian Eigenmaps（拉普拉斯特征映射）等，如下图4所示：

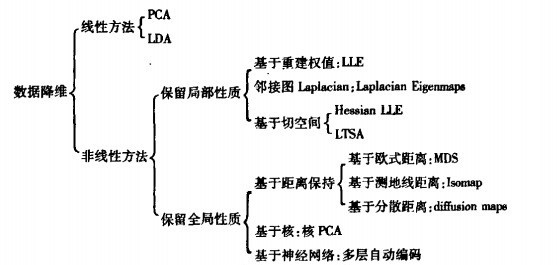


图4 现有数据降维方法分类

## （一）PCA（主成分分析算法）

主成分分析法【9,10】是一种数学变换的方法, 它把给定的一组相关变量通过线性变换转成另一组不相关的变量，这些新的变量按照方差依次递减的顺序排列。在数学变换中保持变量的总方差不变，使第一变量具有最大的方差，称为第一主成分，第二变量的方差次大，并且和第一变量不相关，称为第二主成分。依次类推，I个变量就有I个主成分。

## （二）LDA（线性判别式分析）

线性判别式分析【11，12】(Linear Discriminant Analysis, LDA)，也叫做Fisher线性判别(Fisher Linear Discriminant ,FLD)，是模式识别的经典算法，它是在1996年由Belhumeur引入模式识别和人工智能领域的。基本思想是将高维的模式样本投影到最佳鉴别矢量空间，以达到抽取分类信息和压缩特征空间维数的效果，投影后保证模式样本在新的子空间有最大的类间距离和最小的类内距离，即模式在该空间中有最佳的可分离性。因此，它是一种有效的特征抽取方法。使用这种方法能够使投影后模式样本的类间散布矩阵最大，并且同时类内散布矩阵最小。就是说，它能够保证投影后模式样本在新的空间中有最小的类内距离和最大的类间距离，即模式在该空间中有最佳的可分离性。

## （三）LLE（局部线性嵌入）

局部线性嵌入算法【13-15】(locallylinearembedding,LLE)是解决降维的方法,针对LLE计算速度和近邻点个数K的选取,研究了该方法的扩展,提出了基于聚类和改进距离的LLE方法.基于聚类LLE方法大大缩减了计算LLE方法的时间;改进距离的LLE方法在近邻点个数取值比较小时的情况下,可得到良好的效果,而原始的LLE方法要达到相同的效果,近邻点个数K的取值通常要大很多.同时,改进距离的LLE方法可以模糊近邻点个数选取.实验结果表明,基于聚类和改进距离相结合的LLE方法相比原来的LLE方法大大提高了降维速度和扩大了参数K的选取.

## （四）Laplacian Eigenmaps（拉普拉斯特征映射）

拉普拉斯特征映射【16-18】 ，是用局部的角度去构建数据之间的关系。如果两个数据实例i和j很相似，那么i和j在降维后目标子空间中应该尽量接近。它的直观思想是希望相互间有关系的点（在图中相连的点）在降维后的空间中尽可能的靠近。Laplacian Eigenmaps可以反映出数据内在的流形结构。

# 四、特征选择

机器学习中有一个简单的原则，种瓜得瓜，种豆得豆。当你往模型中扔入大量垃圾的数据（带有噪声的数据），输出也只会是有大量垃圾的结果（带有噪声的结果）。尤其是当选用的特征集更大时，那意味着同时也会有更大的概率引入更多的噪声，上面说的这个问题就会变得更加严重。因此你没有必要使用所有的特征来建模，我们需要对特征进行甄选，只需要放入那些真正重要的，就如 Rohan Rao 也说过的“Sometimes, less is better!”。使用特征选择的主要理由如下：

* 减少特征数量、降维，更快的模型训练速度
* 更低的模型复杂度和更好的解释性
* 更高的精度（选对特征）
* 减弱了过拟合，使模型泛化能力更强

通常而言，我们选择特征的目的是想选到有效的特征，一个有效的特征至少有以下的三个表现：同类样本的不变性、不同样本的鉴别性、对噪声的鲁棒性。接下来，我们先介绍什么是特征选择，然后将讨论各种选择特征子集的方法。

## （一）特征选择

特征选择( Feature Selection ) 【19-21】，也称特征子集选择( Feature Subset Selection , FSS ) ，或属性选择( Attribute Selection ) ，是指从全部特征中选取一个特征子集，使构造出来的模型更好。

**1、 特征选择的一般过程**

特征选择的一般过程可用图5表示。首先从特征全集中产生出一个特征子集，然后用评价函数对该特征子集进行评价，评价的结果与停止准则进行比较，若评价结果比停止准则好就停止，否则就继续产生下一组特征子集，继续进行特征选择。选出来的特征子集一般还要验证其有效性。

综上所述，特征选择过程一般包括产生过程，评价函数，停止准则，验证过程，这4个部分。

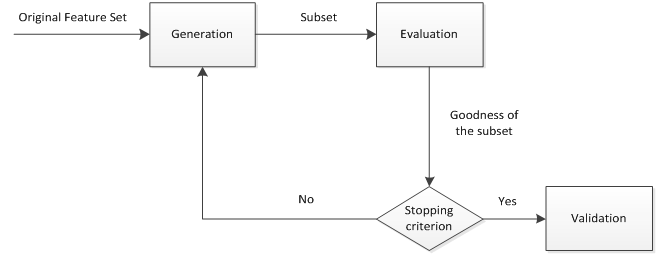


图5 特征选择的过程 ( M. Dash and H. Liu 1997 )

(1) 产生过程( Generation Procedure )

产生过程是搜索特征子集的过程，负责为评价函数提供特征子集。搜索特征子集的过程有多种，将在2.2小节展开介绍。

(2) 评价函数( Evaluation Function )

评价函数是评价一个特征子集好坏程度的一个准则。评价函数将在2.3小节展开介绍。

(3) 停止准则( Stopping Criterion )

停止准则是与评价函数相关的，一般是一个阈值，当评价函数值达到这个阈值后就可停止搜索。

(4) 验证过程( Validation Procedure )

在验证数据集上验证选出来的特征子集的有效性

**2、特征选择方法**

通常来说，从两个方面考虑来选择特征：

* **特征是否发散：**如果一个特征不发散，例如方差接近于0，也就是说样本在这个特征上基本上没有差异，这个特征对于样本的区分并没有什么用。
* **特征与目标的相关性：**这点比较显见，与目标相关性高的特征，应当优选选择。除方差法外，本文介绍的其他方法均从相关性考虑。

根据特征选择的形式又可以将特征选择方法分为3种：

（1）Filter：过滤法，按照发散性或者相关性对各个特征进行评分，设定阈值或者待选择阈值的个数，选择特征。

（2）Wrapper：包装法，根据目标函数（通常是预测效果评分），每次选择若干特征，或者排除若干特征。

（3）Embedded：嵌入法，先使用某些机器学习的算法和模型进行训练，得到各个特征的权值系数，根据系数从大到小选择特征。类似于Filter方法，但是是通过训练来确定特征的优劣。

以下会具体介绍这三种特征选择的方法。

## （二）、过滤法(Filter)

https://pic3.zhimg.com/v2-b5c2a3a5ce0da7ef49fdd6f0a1614372_b.png

过滤式特征选择的评价标准从数据集本身的内在性质获得，与特定的学习算法无关，因此具有较好的通用性。通常选择和类别相关度大的特征或者特征子集。过滤式特征选择的研究者认为，相关度较大的特征或者特征子集会在分类器上可以获得较高的准确率。过滤式特征选择的评价标准分为四种，即距离度量、信息度量、关联度度量以及一致性度量。过滤式特征选择算法的优缺点分别是：

**优点**：算法的通用性强；省去了分类器的训练步骤，算法复杂性低，因而适用于大规模数据集；可以快速去除大量不相关的特征，作为特征的预筛选器非常合适。

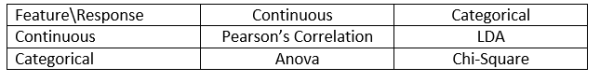
**缺点**：只评估了单个特征对结果的影响，没有考虑到特征之间的关联作用， 可能把有用的关联特征误踢掉。因此工业界使用比较少。由于算法的评价标准独立于特定的学习算法，所选的特征子集在分类准确率方面通常低于Wrapper方法。

**方法**：评估单个特征和结果值之间的相关程度， 排序留下Top相关的特征部分。

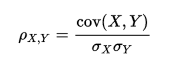
**评价方式**：互信息， 距离相关度、Pearson相关系数等。

**python包**：SelectKBest指定过滤个数、SelectPercentile指定过滤百分比。

过滤法时常应用于预处理阶段，不依赖于任何机器学习算法。它使用基于统计检验的得分作为筛选条件（检验特征和响应变量的相关性），这里所定义相关性带有一些主观色彩。最基本地，你可以参考下面的表格来定义相关性。



* Pearson’s Correlation: 它能反应两个连续变量间的线性相关程度，取值在[-1, 1]上，计算方法如下（译者注：实际上，Pearson相关系数更多反应是两个服从正态分布的随机变量的线性相关性。如果变量虽然连续但是分布和正态分布相距比较远，建议采用非参数的spearman相关系数或kendall相关系数。）：



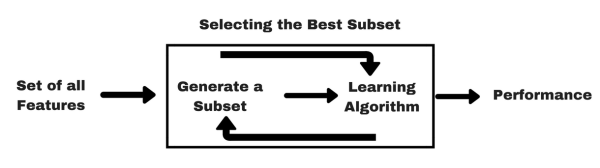
* LDA：全名线性判别分析，可以考察特征的线性组合能否区分一个分类变量。
* ANOVA：中文叫方差分析，原理和LDA类似只是它常用于特征是分类变量，响应变量是连续变量的情况下，它提供了不同组的均值是否相同的统计量。
* Chi-Square：卡方检验是基于频率分布来检验分类变量间的相关性的工具。（引文注：生物统计常用的列联表检验就是卡方检验，实际上和方差分析一毛一样，且都能通过回归形式来表示）
* 互信息法

经典的互信息也是评价定性自变量对定性因变量的相关性的，互信息计算公式如下：

C:\Users\dell\Desktop\2017年4月特征工程\927391-20160502145723247-1184422794.png

同时，请牢记过滤法不能减弱特征间的共线性，在训练模型前还需要针对特征的多重共线性做相应处理。

## （三）、包装法（Wrapper）



封装式特征选择是利用学习算法的性能来评价特征子集的优劣。因此，对于一个待评价的特征子集，Wrapper方法需要训练一个分类器，根据分类器的性能对该特征子集进行评价。Wrapper方法中用以评价特征的学习算法是多种多样的，例如决策树、神经网络、贝叶斯分类器、近邻法以及支持向量机等等。封装式特征选择算法的优缺点分别是：

**优点：**相对于Filter方法，Wrapper方法找到的特征子集分类性能通常更好。

缺点：Wrapper方法选出的特征通用性不强，当改变学习算法时，需要针对该学习算法重新进行特征选择；由于每次对子集的评价都要进行分类器的训练和测试，所以算法计算复杂度很高，尤其对于大规模数据集来说，算法的执行时间很长。

**方法：**把特征选择看做一个特征子集搜索问题， 筛选各种特征子集，用模型评估效果。

典型算法：“递归特征删除算法”。

应用在逻辑回归的过程：用全量特征跑一个模型；根据线性模型的系数(体现相关性)，删掉5-10%的弱特征，观察准确率/auc的变化；逐步进行， 直至准确率/auc出现大的下滑停止。

**python包：**RFE

包装法会仅用特征的一个子集来训练模型，并利用之前模型的结果来判断是否需要增删新的特征。这其实就是一个关于特征空间的搜索问题，但它的计算可能需要耗费大量时间空间。常用的包装法包括了前向选择法，后向剔除法，迭代剔除法等。

（1）前向选择法：这是一种基于循环的方法，开始时我们训练一个不包含任何特征的模型，而后的每一次循环我们都持续放入能最大限度提升模型的变量，直到任何变量都不能提升模型表现。

（2）后向剔除法：该方法先用所有特征建模，再逐步剔除最不显著的特征来提升模型表现。同样重复该方法直至模型表现收敛。

（3）递归特征消除法：这是一种搜索最优特征子集的贪心优化算法。它会反复地训练模型并剔除每次循环的最优或最劣特征。下一次循环，则使用剩余的特征建模直到所有特征都被剔除。之后，按照剔除的顺序给所有特征排序作为特征重要性的度量。

利用包装法选特征还可以借助R语言中的Boruta包，它的工作原理如下：

首先，它会把原来的特征打乱顺序作为新特征（称为影特征）添加到数据集中

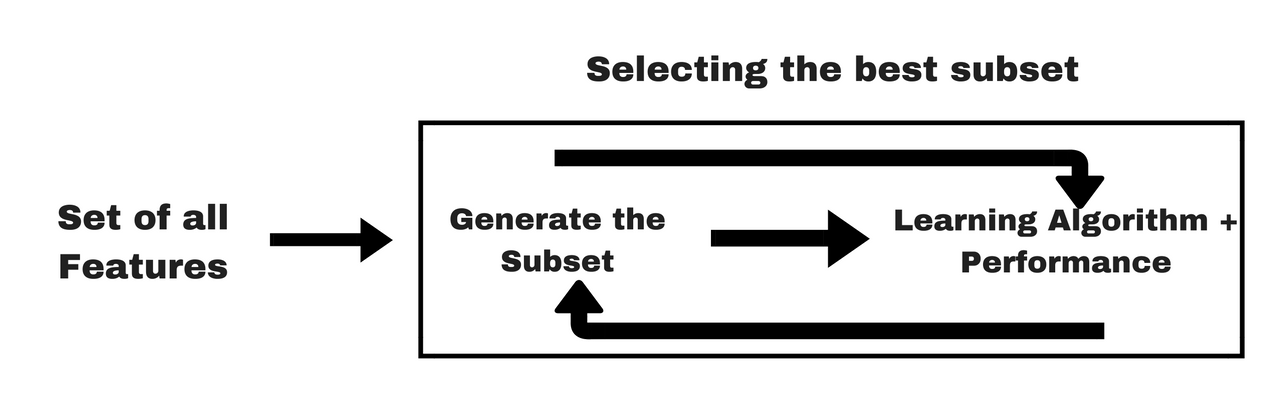
而后，基于所有特征训练随机森林模型，并评价每个特征的重要性（默认基于平均精度降低测度）

每一次迭代中，该方法都会检测真实特征相对其影特征是否更重要，并移除哪些重要性差别最低的特征。

最后，该算法会在所有特征都被判定为重要或无用之后停止（或者在达到给定的迭代次数后停止）

**注**：这里包装法的分类方法和特征工程思维导图的包装法的分类有所不同，主要是所参考文献的有所不同，如果读者希望从完全搜索、启发式搜索、随机搜索出发进行考虑特征选择，请参考文献【20】。

## （四）、嵌入法（Embedded）



在嵌入式特征选择中，特征选择算法本身作为组成部分嵌入到学习算法里。最典型的即决策树算法，如ID3、C4.5以及CART算法等，决策树算法在树增长过程的每个递归步都必须选择一个特征，将样本集划分成较小的子集，选择特征的依据通常是划分后子节点的纯度，划分后子节点越纯，则说明划分效果越好，可见决策树生成的过程也就是特征选择的过程。

方法：嵌入法综合了过滤法和包装法的特点，它要借助那些自带特征选择方法的算法。最常用的嵌入法实例是LASSO和岭回归，他们的优化目标都带有惩罚项来减弱过拟合。LASSO使用L1正则，也就是对系数的绝对值大小加以惩罚。岭回归使用L2正则也就是对系数的平方值加以惩罚。

 举例：最早在电商用LR做CTR预估， 在3-5亿维的系数特征上用L1正则化的LR模型。

1. **基于树模型的特征选择法**

树模型中GBDT也可用来作为基模型进行特征选择，使用feature\_selection库的SelectFromModel类结合GBDT模型。

随机森林具有准确率高、鲁棒性好、易于使用等优点，这使得它成为了目前最流行的机器学习算法之一。随机森林提供了两种特征选择的方法：mean decrease impurity和mean decrease accuracy。

（1） 平均不纯度减少 mean decrease impurity

随机森林由多个决策树构成。决策树中的每一个节点都是关于某个特征的条件，为的是将数据集按照不同的响应变量一分为二。利用不纯度可以确定节点（最优条件），对于分类问题，通常采用基尼不纯度或者信息增益，对于回归问题，通常采用的是方差或者最小二乘拟合。当训练决策树的时候，可以计算出每个特征减少了多少树的不纯度。对于一个决策树森林来说，可以算出每个特征平均减少了多少不纯度，并把它平均减少的不纯度作为特征选择的值。

使用基于不纯度的方法的时候，要记住：1、这种方法存在偏向，对具有更多类别的变量会更有利；2、对于存在关联的多个特征，其中任意一个都可以作为指示器（优秀的特征），并且一旦某个特征被选择之后，其他特征的重要度就会急剧下降，因为不纯度已经被选中的那个特征降下来了，其他的特征就很难再降低那么多不纯度了，这样一来，只有先被选中的那个特征重要度很高，其他的关联特征重要度往往较低。在理解数据时，这就会造成误解，导致错误的认为先被选中的特征是很重要的，而其余的特征是不重要的，但实际上这些特征对响应变量的作用确实非常接近的（这跟Lasso是很像的）。

需要注意的一点是，关联特征的打分存在不稳定的现象，这不仅仅是随机森林特有的，大多数基于模型的特征选择方法都存在这个问题。

（2）平均精确率减少 Mean decrease accuracy

另一种常用的特征选择方法就是直接度量每个特征对模型精确率的影响。主要思路是打乱每个特征的特征值顺序，并且度量顺序变动对模型的精确率的影响。很明显，对于不重要的变量来说，打乱顺序对模型的精确率影响不会太大，但是对于重要的变量来说，打乱顺序就会降低模型的精确率。

1. **基于惩罚项的特征选择法**

正则化就是把额外的约束或者惩罚项加到已有模型（损失函数）上，以防止过拟合并提高泛化能力。使用带惩罚项的基模型，除了筛选出特征外，同时也进行了降维。损失函数由原来的E(X,Y)变为E(X,Y)+alpha||w||，w是模型系数组成的向量（有些地方也叫参数parameter，coefficients），||·||一般是L1或者L2范数，alpha是一个可调的参数，控制着正则化的强度。当用在线性模型上时，L1正则化和L2正则化也称为Lasso和Ridge。

（1）L1正则化/Lasso

L1正则化将系数w的l1范数作为惩罚项加到损失函数上，由于正则项非零，这就迫使那些弱的特征所对应的系数变成0。因此L1正则化往往会使学到的模型很稀疏（系数w经常为0），这个特性使得L1正则化成为一种很好的特征选择方法。

然而，L1正则化像非正则化线性模型一样也是不稳定的，如果特征集合中具有相关联的特征，当数据发生细微变化时也有可能导致很大的模型差异。

（2）L2正则化/Ridge regression

L2正则化将系数向量的L2范数添加到了损失函数中。由于L2惩罚项中系数是二次方的，这使得L2和L1有着诸多差异，最明显的一点就是，L2正则化会让系数的取值变得平均。对于关联特征，这意味着他们能够获得更相近的对应系数。

还是以Y=X1+X2为例，假设X1和X2具有很强的关联，如果用L1正则化，不论学到的模型是Y=X1+X2还是Y=2X1，惩罚都是一样的，都是2alpha。但是对于L2来说，第一个模型的惩罚项是2alpha，但第二个模型的是4\*alpha。可以看出，系数之和为常数时，各系数相等时惩罚是最小的，所以才有了L2会让各个系数趋于相同的特点。

可以看出，L2正则化对于特征选择来说一种稳定的模型，不像L1正则化那样，系数会因为细微的数据变化而波动。所以L2正则化和L1正则化提供的价值是不同的，L2正则化对于特征理解来说更加有用：表示能力强的特征对应的系数是非零。

关于LASSO和岭回归的更多细节，可以参考这篇文章【22】。其他的方法则有正则树，文化基因算法和随机多项logit等。

## （五）过滤法和包装法的不同

* 过滤法测量特征和被解释变量的相关性，包装法则是基于模型测量特征的有效性。
* 过滤法由于不依赖于模型，速度更快。
* 过滤法基于统计检验选择特征，包装法基于交叉验证。
* 过滤法时常失效，但包装法常常被发现很有用。
* 使用包装法筛选的特征更容易导致模型过拟合。

# 写在后面的话

特征工程是数据挖掘过程中比较重要的一环，关于特征工程，目前已经有许多慷慨分享的作者在其博客网页上做了详细和完整的介绍。存在一个比较小的问题是，当你去读各个作者总结的文档时，也会发现各个作者其实在特征工程的分类方式和侧重点各有不同。于是，笔者本着虚心学习的精神，从各作者博客和论坛网页上搜集了关于特征工程这方面的资料，归纳梳理了一遍，总结的这篇文档可谓是各个作者智慧的结晶。当然，由于本人对特征工程的理解还不够深刻，归纳的这篇文档会有不少的错误和不足之处，希望大家指出错误，及时联系我予以其中错误的更正和内容的补充（邮箱：gb\_xiao@bupt.edu.cn）。

# 参考资料

1. [机器学习项目中的数据预处理与数据整理之比较](http://www.36dsj.com/archives/81824)

<http://www.36dsj.com/archives/81824>

1. 数据清洗的一些梳理

<http://www.36dsj.com/archives/44958>

1. R语言不平衡数据分类指南

<https://zhuanlan.zhihu.com/p/24826792>

1. 机器学习中的数据清洗与特征处理综述

<http://tech.meituan.com/machinelearning-data-feature-process.html?utm_source=tuicool&utm_medium=referral>

1. 使用python进行数据清洗

<http://bluewhale.cc/2016-08-21/python-data-cleaning.html>

1. 高维数据的降维方法

<http://www.cnblogs.com/lyfruit/articles/3068062.html>

1. 数据降维方法小结

<http://blog.csdn.net/yujianmin1990/article/details/48223001>

1. [四大机器学习降维算法：PCA、LDA、LLE、Laplacian Eigenmaps](http://www.36dsj.com/archives/26723)

<http://www.36dsj.com/archives/26723>

1. 主元分析(PCA)理论分析及应用

<http://www.360doc.com/content/10/0318/20/1024901_19297267.shtml>

1. 主成份分析（PCA）详解

<http://blog.jobbole.com/109015/>

1. LDA线性判别分析

<http://blog.csdn.net/porly/article/details/8020696>

1. 线性判别分析

<http://blog.csdn.net/daunxx/article/details/51881956>

1. 局部线性嵌入（LLE）原理总结

<http://blog.csdn.net/tcdpyh6sa3/article/details/61191617>

1. 用scikit-learn研究局部线性嵌入(LLE)

<http://www.w2bc.com/article/206468>

1. LLE算法主页

<http://www.cs.nyu.edu/~roweis/lle/code.html>

1. 《Laplacian Eigenmaps for Dimensionality Reduction and Data Representation》,
2. 降维（二）----Laplacian Eigenmaps

<http://blog.csdn.net/jwh_bupt/article/details/8945083>

1. 机器学习降维算法四：Laplacian Eigenmaps 拉普拉斯特征映射

<http://blog.csdn.net/xbinworld/article/details/8855796>

1. Scikit-learn介绍几种常用的特征选择方法

<http://blog.csdn.net/bryan__/article/details/51607215>

1. 特征选择常用算法综述

<http://blog.csdn.net/pi9nc/article/details/8145901>

1. 特征选择方法导论（如何选取合适的变量）

<https://zhuanlan.zhihu.com/p/24552933>

1. A Complete Tutorial on Ridge and Lasso Regression in Python

<https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-ridge-lasso-regression-python/>