## 0 概述

机器学习应用到实际问题中的一般流程为：数据预处理 -> 特征工程 -> 建模 -> 训练 -> 预测。

可以看出，第一个阶段就是数据预处理。数据预处理主要是将收集到的数据进行清洗、格式转换等。因为收集的raw数据往往存在各种各样的问题（缺失值、格式不符合我们的需求等）。

现在业界流行的深度学习模型，号称是“端对端”的建模，但只是可以省略特征工程这一步，而数据预处理这一步仍然是需要的。

## 数据格式转换

数据格式转换就是将采集到的数据的格式转换成模型规定的输入数据格式。比如对于文本来说，有时候采集到的原始数据是标签形式的，需要解析成纯文本；文本数据还可能需要对编码方式作转换。

很多模型都对输入数据的格式（或者，准确地说，数据的形式）有特定的要求。比如，fasttext要求输入的待分类的样本形式为：

已经分词的文档 + \t + \_\_label\_\_标签

## 处理缺失值

一般来说，若某个特征缺失值非常多，就把该特征删掉；若某个样本在很多的特征上都缺失值，则删掉该样本；否则的话，需要填充缺失值。填充缺失值有两种策略：（1）用相应特征的平均数或者众数填充缺失值，（2）根据相应样本的其他特征，按照一些先验信息填充。

## 类别型变量的独热编码

one-hot向量是对离散的类别型特征进行编码的方法。离散型特征分为类别型特征和非类别型特征。类别型特征的取值没有大小之分（比如科目：语文、数学、英语），非类别型特征的取值有大小之分（比如成绩的等级：优、良、及格、不及格）。对类别型特征进行编码时，为了使取值不具有大小的区分，故使用one-hot向量。

如何使用one-hot向量对一个类别型特征进行编码呢？很简单，就是将该类别型特征分裂成n个特征（n是该类别型特征的取值个数），每个新的特征对应于原类别型特征的一个取值。对于每个样本，新的n个特征的取值为：若样本在原始类别特征上取某个值，那么在该值对应的新的特征上取值为1，否则为0。

注意，对于决策树模型，情况比较特殊：大多数模型都是针对连续型变量的，所以需要对离散型中的类别型特征作独热编码。但像决策树这种模型，它可以处理类别型变量，因为它pick到类别型特征作为节点分裂的特征时，可以按照类别的取值进行分裂（类别型特征取几个值，就分裂成几个子节点，当然，也可以用one-vs-rest策略进行分裂），因此不用作独热编码。对于离散型的非类别型变量，决策树可以按照类别型进行处理，也可以按照连续型进行处理（找分割点进行二分，当然也可以多分），因此也不用作特殊处理。但是实际中，如果类别型的特征跟非类别型特征混在一起，那就得对类别型特征作独热编码，因为决策树无法识别哪个特征是连续的，哪个特征是离散的以及是类别型的，因此通常都按照连续型的来处理。

## 数据标准化

数据的标准化是为了使数据的每个特征的分布都处于相同的范围，或者使每个特征都具有相同的均值和方差。

数据标准化的场景：

(1) 线性模型：对于线性模型（或者广义线性模型），在建模时，我们希望衡量数据的每一个特征的变化对预测结果的变化的影响大小。而这里说的“特征的变化”，指的是相对变化，而不是绝对变化。我们关心的是相对变化（因为“特征对预测结果的影响”本质上就意味着相对变化，只要有了相对变化的规律，模型可以通过缩放（权系数等）等找出特征跟预测结果的直接关系），所以需要对各个特征作标准化。对于线性模型，作标准化后，就可以使每个特征的权系数间的相对大小能够完全反映每个特征的重要程度的相对大小（如果不归一化，权系数中还会蕴含着相应的特征的值域信息）。

(2) 基于样本间距离的模型：比如KNN，直接计算距离的话，若两个特征的值域范围相差较大，那么距离的大小基本上由值域较大的特征决定，值域较小的特征基本上起不到什么作用。所以要做标准化。

(3) 对于基于梯度下降优化的模型，提前对数据归一化可能会加速收敛。

<https://www.julyedu.com/question/big/kp_id/26/ques_id/920>

## 数据降维

数据降维就是在尽可能保留原始数据信息的前提下，压缩数据的特征维数，使得降维后各维特征尽可能去相关。数据降维的主要作用是减小存储量，提高计算效率，并有去噪的神奇功效。

## 数据不平衡问题处理

“样本不平衡”，更完整的说法是“样本类别不平衡”，是针对分类问题而言的。它是指样本中某些类别的样本数相对其它类别的样本数少很多。

样本不平衡分两种情况：

(1) 大样本的不平衡：样本的总数较大。如有1000万样本，其中A类样本只有5万。这种情况下模型很容易忽视A类样本；

(2) 小样本不平衡：样本总数较小。如有100个样本，其中10个A类样本，90个B类样本。这种情况下A类样本数（绝对数量）很少，包含的信息太少，无法从中提取模式。

实际中样本不平衡的典型场景包括：CTR预估（点击事件是少数类样本）、异常检测（异常点是少数类）、罕见事件分析（罕见事件是少数类）。

工业上的样本不平衡的极限是30：1。

样本不平衡的影响：对于分类问题来说，由于我们训练模型的损失函数一般是交叉熵损失，这就相当于，模型的优化方向是提高训练集的样本的分类准确率。因此如果训练样本不平衡，比如有一个二分类问题，正例很少而负例很多，那么模型就会倾向于把新样本预测为负例，而不是去努力学习数据的特征的规律和特征与分类标签之间的关系。

### 1. 欠采样

欠采样，又称下采样，是将多数类样本采少，以跟少数类样本平衡。

1、随机欠采样：在多数类样本中随机删除一些样本。

2、PG欠采样算法：基于k-means或其他原型算法的欠采样算法。设少数类样本数是，多数类样本数是，多数类样本平衡后的样本数是（与接近），则PG算法迭代次，每次从多数类样本中选出

### 2. 过采样

过采样，又称上采样，是将少数类样本采多，以跟多数类样本平衡。

(1) 随机复制：就是简单的复制少数类样本。这种方法的缺点是容易造成过拟合，因为少数类样本本来信息就少，随机复制相当于加强了少数类样本的信息，但是没有提供新的信息，所以模型比原来更不易在新的样本上泛化。

(2) smote样本构建算法：与随机复制相反，它构建新的样本。

smote过采样方法的思路很简单：在少数类样本中，任意选取两个样本，然后进行线性插值，产生新的样本。但是这个做法也过于简单，没有考虑到一些信息，比如少数类样本的分布特征，这样在少数类与多数类的分界线附近容易造成误差；

基于简单的smote过采样算法，有很多改进的方法。如borderline-smote算法、kmeans-smote算法等，并且smote算法可以与Boosting算法结合，形成smoteBoost。

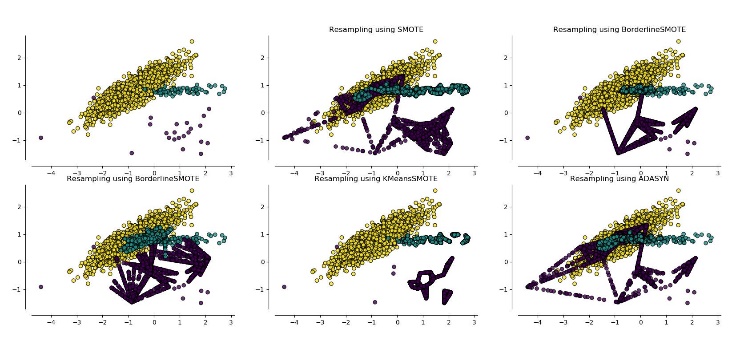


图1 各种smote的改进算法

注意，这里介绍的过采样是针对连续特征的样本的（因为需要插值）。

### 3. 基于模型的方法

欠采样和过采样都是从训练样本出发去处理样本不平衡问题。另一种思路是从模型的角度来考虑解决样本不平衡问题。其基本思路是，对少数类样本，增加其loss的权重，这样，若少数类样本被错分，其相对多数类样本有更大的loss。如此一来，模型在学习的时候就会增加对少数类样本的关注。

对于端到端的神经网络NLP模型，输入都是句子，这相当于非连续的特征，因此不能使用欠采样或过采样的方法，这时，除了文本数据增强以外，还可以考虑基于模型的方法。

cost-sensitive方法是一种通用的方法，它适用于任意模型，基于模型对样本的类别标签进行修改。关于cost-sensitive方法，请参见这篇论文：

《The Foundations of Cost-Sensitive Learning》

cost-sensitive方法的具体实现有meta-cost算法和focal-cost算法等。关于focal-cost，请参见这篇论文：《Focal Loss for Dense Object Detection》。

## 第七章 数据增强