一、概述

本文主要讲述机器学习中对数据的一些预处理，主要参考《Python机器学习经典实例》，以及各类博客。

二、内容

1. 均值移除

概述：

为了统一样本矩阵中不同特恒的基准值和分散度，可以将各个特征的平均值调整为0，标准差调整为1，这个过程称为均值移除。

公式为：(X-mean)/std 计算时对每个属性/每列分别进行。

sklearn.preprocessing.scale(原始样本矩阵) --> return：均值移除后的样本矩阵(mean=0, std=1)

2. 范围缩放

概述：

统一样本矩阵中不同特征的最大值和最小值范围。将属性缩放到一个指定的最大和最小值（通常是1-0）之间，这样处理可对方差非常小的属性增强其稳定性，也可维持稀疏矩阵中为0的条目。

$$x^* = \frac{x - x.\min(axis = 0)}{x.\max(axis = 0) - x.\min(axis = 0)}$$

sklearn.preprocessing.MinMaxScaler(feature_range=期望最小最大值, copy=True) --> return: 范围缩放，范围缩放器.fit_transform(原始样本矩阵) --> return：范围缩放后的样本矩阵

3. 归一化（正则化）

概述：

为了用占比表示特征，用每个样本的特征值除以该样本的特征值绝对值之和，以使每个样本的特征值绝对值之和为1（转化为占比 normalized）

语法：

sklearn.preprocessing.normalize(原始样本矩阵，norm='l1') --> return：归一化后的样本矩阵

什么情况下（不）需要归一化

- 需要： 基于参数的模型或基于距离的模型，都是要进行特征的归一化
- 不需要：基于树的方法是不需要进行特征的归一化，例如随机森林，bagging 和 boosting等

备注:

l1即L1范数，矢量中各元素绝对值之和。

l2即L2范数，矢量元素绝对值的平方和再开方。

p-范数的计算公式：||X||p=(|x1|^p+|x2|^p+...+|xn|^p)^1/p

向量的 p-范数

1,文字表达:

若 x 为 n 维向量，那么定义 p-范数为：

当 p=1,2，∝时候是比较常用的范数。

1-范数是向量个分量绝对值之和。

2-范数（Euclid 范数）就是通常所说的向量的长度。

∝-范数 是通常所说的最大值范数，指的是向量各个分量绝对值的最大值。

2,数学表达:

令 x=(x1,x2,...,xn)T （T 是转置的意思）

1-范数:‖x‖1=|x1|+|x2|+...+|xn|

2-范数:‖x‖2=(|x1|^2+|x2|^2+...+|xn|^2)^1/2

∞-范数:‖x‖∞=max(|x1|,|x2|,...,|xn|)

易得推论： ‖x‖∞≤‖x‖2≤‖x‖1≤n1/2‖x‖2≤n‖x‖∞

4. 二值化Binarizer

概述:

用0和1来表示样本矩阵中相对于某个给定阈值高于或低于它的元素

语法:

1）生成二值化器

sklearn.preprocessing.Binarizer(threshold=阈值, copy=True) --> return：二值化器,

2）二值化

二值化器.transform(原始样本矩阵) --> return: 二值化后的样本矩阵.

备注:

1) threshold:

feature <= threshold: feature = 0;

feature > threshold: feature = 1;

2) 二值化方法不可逆，若希望0-1可逆话 可考虑使用独热编码进行可逆的transform

三.心得体会

这些数据预处理是最基本的用法，数据在使用处理时第一步就是经过这些步骤。之后的yolo3、

参考博客：https://blog.csdn.net/tcsjk/article/details/82774376
https://www.cnblogs.com/chaosimple/p/4153167.html