昵称：ChaoSimple
园龄：5年9个月
粉丝：321
关注：3
+加关注

【原】关于使用sklearn进行数据预处理 —— 归一化/标准化/正则化

一、标准化（Z-Score），或者去除均值和方差缩放

公式为：(X-mean)/std 计算时对每个属性/每列分别进行。

将数据按期属性（按列进行）减去其均值，并处以其方差。得到的结果是，对于每个属性/每列来说所有数据都聚集在0附近，方差为1。

实现时，有两种不同的方式：

- 使用sklearn.preprocessing.scale()函数，可以直接将给定数据进行标准化。

```
>>> from sklearn import preprocessing
>>> import numpy as np
>>> X = np.array([[ 1., -1.,  2.],
...               [ 2.,  0.,  0.],
...               [ 0.,  1., -1.]])
>>> X_scaled = preprocessing.scale(X)

>>> X_scaled
array([[ 0.  ..., -1.22...,  1.33...],
       [ 1.22...,  0.  ..., -0.26...],
       [-1.22...,  1.22..., -1.06...]])

>>>#处理后数据的均值和方差
>>> X_scaled.mean(axis=0)
array([ 0.,  0.,  0.])

>>> X_scaled.std(axis=0)
array([ 1.,  1.,  1.])
```

- 使用sklearn.preprocessing.StandardScaler类，使用该类的好处在于可以保存训练集中的参数（均值、方差）直接使用其对象转换测试集数据。

```
>>> scaler = preprocessing.StandardScaler().fit(X)
>>> scaler
StandardScaler(copy=True, with_mean=True, with_std=True)

>>> scaler.mean_
array([ 1. ...,  0. ...,  0.33...])

>>> scaler.std_
array([ 0.81...,  0.81...,  1.24...])

>>> scaler.transform(X)
array([[ 0.  ..., -1.22...,  1.33...],
       [ 1.22...,  0.  ..., -0.26...],
       [-1.22...,  1.22..., -1.06...]])


>>>#可以直接使用训练集对测试集数据进行转换
>>> scaler.transform([[-1.,  1., 0.]])
array([[-2.44...,  1.22..., -0.26...]])
```

二、将属性缩放到一个指定范围

除了上述介绍的方法之外，另一种常用的方法是将属性缩放到一个指定的最大和最小值（通常是1-0）之间，这可以通过preprocessing.MinMaxScaler类实现。

使用这种方法的目的包括：

1、对于方差非常小的属性可以增强其稳定性。

2、维持稀疏矩阵中为0的条目。

```
>>> X_train = np.array([[ 1., -1.,  2.],
...                     [ 2.,  0.,  0.],
...                     [ 0.,  1., -1.]])
...
>>> min_max_scaler = preprocessing.MinMaxScaler()
>>> X_train_minmax = min_max_scaler.fit_transform(X_train)
>>> X_train_minmax
array([[ 0.5       ,  0.        ,  1.        ],
       [ 1.        ,  0.5       ,  0.33333333],
       [ 0.        ,  1.        ,  0.        ]])

>>> #将相同的缩放应用到测试集数据中
>>> X_test = np.array([[ -3., -1.,  4.]])
>>> X_test_minmax = min_max_scaler.transform(X_test)
>>> X_test_minmax
array([[-1.5       ,  0.        ,  1.66666667]])


>>> #缩放因子等属性
>>> min_max_scaler.scale_
array([ 0.5       ,  0.5       ,  0.33...])

>>> min_max_scaler.min_
array([ 0.        ,  0.5       ,  0.33...])
```

当然，在构造类对象的时候也可以直接指定最大最小值的范围：feature_range=(min, max)，此时应用的公式变为：

X_std=(X-X.min(axis=0))/(X.max(axis=0)-X.min(axis=0))

X_scaled=X_std/(max-min)+min

三、正则化（Normalization）

正则化的过程是将每个样本缩放到单位范数（每个样本的范数为1），如果后面要使用如二次型（点积）或者其它核方法计算两个样本之间的相似性这个方法会很有用。

Normalization主要思想是对每个样本计算其p-范数，然后对该样本中每个元素除以该范数，这样处理的结果是使得每个处理后样本的p-范数（l1-norm,l2-norm）等于1。

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

该方法主要应用于文本分类和聚类中。例如，对于两个TF-IDF向量的l2-norm进行点积，就可以得到这两个向量的余弦相似性。

1、可以使用preprocessing.normalize()函数对指定数据进行转换：

```
>>> X = [[ 1., -1.,  2.],
...      [ 2.,  0.,  0.],
...      [ 0.,  1., -1.]]
>>> X_normalized = preprocessing.normalize(X, norm='l2')

>>> X_normalized
array([[ 0.40..., -0.40...,  0.81...],
       [ 1.  ...,  0.  ...,  0.  ...],
       [ 0.  ...,  0.70..., -0.70...]])
```

2、可以使用processing.Normalizer()类实现对训练集和测试集的拟合和转换：

```
>>> normalizer = preprocessing.Normalizer().fit(X)  # fit does nothing
>>> normalizer
Normalizer(copy=True, norm='l2')

>>>
>>> normalizer.transform(X)
array([[ 0.40..., -0.40...,  0.81...],
       [ 1.  ...,  0.  ...,  0.  ...],
       [ 0.  ...,  0.70..., -0.70...]])

>>> normalizer.transform([[-1.,  1., 0.]])
array([[-0.70...,  0.70...,  0.  ...]])
```

补充:

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

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: $\|x\|\infty=\max(|x1|,|x2|,....,|xn|)$

易得推论:　$\|x\|\infty\leq\|x\|2\leq\|x\|1\leq n1/2\|x\|2\leq n\|x\|\infty$

分类: Python

ChaoSimple
关注 - 3
粉丝 - 321
+加关注

8　　1

« 上一篇: 【原】关于使用Sklearn进行数据预处理 —— 缺失值（Missing Value）处理
» 下一篇: 【转】The difference between categorical（Nominal）, ordinal and interval variables

posted @ 2014-12-09 14:14 ChaoSimple 阅读(90072) 评论(8) 编辑 收藏

评论

#1楼 2016-01-14 17:42 | yinsua

赞一个，通俗易懂

支持(2)　反对(0)

#2楼 2017-07-22 10:33 | jzhx107

不错，谢谢!

支持(0) 反对(0)

#3楼 2017-08-29 09:46 | study_cc

非常感谢分享。
不过有个问题求教：

>>>#可以直接使用训练集对测试集数据进行转换
>>> scaler.transform([[-1., 1., 0.]])
array([[-2.44..., 1.22..., -0.26...]])

上面对测试集进行标准化，为啥可以直接使用训练集的fit对测试集进行转换呢?

这样转换的结果导致训练集的 均值！=0，方差！=1.

支持(0) 反对(0)

#4楼[楼主] 2017-08-29 11:03 | ChaoSimple

@ study_cc
一般情况下，或者严格点说，在监督学习中，我们需要利用训练集数据对测试集数据进行预测。这里隐含了一个假设，就是训练数据和测试数据实际上是同分布的（因此我们才可以使用训练数据集来预测测试数据集），来自于同一个总体。

在进行标准化的过程中就将训练集的均值和方差当做是总体的均值和方差，因此对测试集使用训练集的均值和方差进行预处理。

这样子解释不知道您是否能够理解?

支持(3) 反对(0)

#5楼 2017-08-29 11:08 | study_cc

@ ChaoSimple
有这个假设就明白了。
楼主厉害啊，向你学习

支持(0) 反对(0)

#6楼 2017-09-03 22:18 | WOTGL

std是标准差吧?

支持(0) 反对(0)

#7楼 2017-09-10 15:26 | jiushiyzl

好文，感谢

支持(0) 反对(0)

#8楼 2018-02-06 23:06 | 一遇卧琳便化龙

公式为：(X-mean)/std 计算时对每个属性/每列分别进行。
第一句描述的公式是正确的，但是std的意思是**标准差**；因此第二局的解释应该如下：将数据按期属性（按列进行）减去其均值，并除以其**标准差**。得到的结果是，对于每个属性/每列来说所有数据都聚集在0附近，**标准差**为1。

支持(0) 反对(0)

最新IT新闻:

· 爱奇艺招股书精华版：每用户日均看1.7小时视频内容
· 博通并购高通已基本没戏 不愿把收购价格提高至每股90美元
· 苹果大胆专利：要用触控屏取代笔记本键盘
· 95后回乡见闻：尴尬的伪互联网生活
· 连收两家智能门铃 亚马逊正不遗余力进入用户家中
» 更多新闻...

最新知识库文章:

· 写给自学者的入门指南
· 和程序员谈恋爱
· 学会学习
· 优秀技术人的管理陷阱
· 作为一个程序员，数学对你到底有多重要
» 更多知识库文章...