imputation

Guangyao Zhao

2022-10-18

Contents

拿到的数据中，经常会有缺失值，最简单的办法当然是直接删除 (pd.dropna())，但是如果仅有几个特征有缺失值的话，直接删除未免有点可惜，更好的策略是估算缺失值，即从数据的已知部分推断出缺失值。

- 单变量插补：平均值，众数等
- 多元插补算法：KNN

```
import numpy as np
from sklearn import impute

X = np.array([[1, 2], [np.nan, 3], [7, 6]])
print('X:\n ', X)
```

```
X:
  [[ 1.  2.]
 [nan  3.]
 [ 7.  6.]]
```

单变量插补

```
imp = impute.SimpleImputer(missing_values= np.nan, strategy='mean') # 单变量插补
imp.fit(X)

X_simple = imp.transform(X)
print('X_simple:\n ', X_simple)
```

```
X_simple:
  [[1. 2.]
 [4. 3.]
 [7. 6.]]
```

同样的策略还有：median, most_frequent, constant。如果 strategy=constant，则使用 fill_value 参数填补定值，更多介绍请参考官方文档。

多变量插补

多变量插补相比稍显复杂，但原理也很简单，它将每个包含缺失值的特征建模为其他特征的函数，并将该估计值用于插补。最常用的是 KNN：

```
imp = impute.KNNImputer(n_neighbors=2, weights='uniform')
imp.fit(X)

X_KNN = imp.transform(X)
print('X_KNN:\n ', X_KNN)
```

```
X_KNN:
  [[1. 2.]
 [4. 3.]
 [7. 6.]]
```