数据操作说明

**energy.csv文件**是一个无缺失值的真实多元时间序列数据集。为了降低计算量，只考虑“T1”~“T9”这9个属性。

通过如下代码读取数据：

import pandas as pd

import numpy as np

org\_data = pd.read\_csv("energy.csv")[["T1","T2","T3","T4","T5","T6","T7","T8","T9"]].to\_numpy()

org\_data是一个形状为(19735, 9)的numpy.ndarray数据，表示这是一个时间长度为19735，属性数量为9的多元时间序列

**mask文件夹**中存放5个csv文件，每个csv文件的形状同样为(19735, 9)。

以energy\_0.1\_0.csv为例，该文件中大部分位置为0，少部分位置为1。为0的部分对应到org\_data表示这个位置没有缺失；为1的部分对应到org\_data表示这个位置缺失。

通过如下代码读取缺失位置：

mask\_flag = np.loadtxt("energy\_0.1\_0.csv", delimiter=",")

**需要进行填补的时间序列input\_data**通过org\_data和mask\_flag得到：

input\_data = np.array(org\_data)

input\_data[np.where(mask\_flag == 1)] = -200

此时input\_data数据中值为-200的地方就表示这个地方的值已经缺失。你需要根据其他方法的输入把缺失值替换成其他方法能识别的符号（holoclean和iim都是空值）

你需要通过复现的方法得到一个形状为(19735, 9)的无缺失数据**output\_data**。

并计算output\_data缺失位置和org\_data中的ground truth的Root Mean Square Error：

missing\_pos = np.where(mask\_flag == 1)

predicts = output\_data[missing\_pos]

targets = org\_data[missing\_pos]

rmse = np.sqrt(((np.array(targets) - np.array(predicts)) \*\* 2).mean())

此外，你还需要记录你复现的方法从输入input\_data到填补完成输出output\_data的**运行时间**，单位为秒。

最后你需要将mask文件夹中的其他csv文件按照上述步骤处理，并得到**5次运行rmse和时间的平均值。**