

BlendingFuel

BlendingFuel

1 程序简介

1.1 依赖的包

1.2 使用方法

2 输入文件

2.1 概述

2.2 参数

2.2.1 task

2.2.2 model_type

2.2.3 poly

2.2.4 data_path

2.2.5 target_properties

2.2.6 x_range

2.2.7 train_ratio

2.2.8 out_shuffle

2.2.9 y_scale

2.2.10 target_properties_weight

2.2.11 x0

1 程序简介

BlendingFuel是一个专门为寻找给定体系下的最优混合燃料配方而开发的程序。程序整体的思路分为两步：

1. 根据已经测得的混合燃料性质数据训练能够准确预测该体系下的混合燃料性质的机器学习模型；
2. 将燃料设计问题转化为多目标优化问题，根据人为设定的目标函数对优化问题求解。

1.1 依赖的包

程序中使用的python包有：os, pickle, numpy, pandas, scipy, sklearn

1.2 使用方法

配置好环境并写好输入文件（INPUT.py, 详见第2节）后，运行main.py文件。

INPUT.py可以这样写：

```
"""
task: the task you want BlendingFuel to do.
1: train regression model for predicting fuel properties
2: train regression model for predicting fuel properties and use trained model to
get a best blending fuel scheme
"""
task = 2

# train regression model
"""
```

```

model_type: the model you want to use.
1: linear regression
2: polynomial regression
"""

model_type = 2

"""
poly will only be read when model_type is 2, control the exponents of polynomial
regression
"""
poly = 3

"""
path of dataset file
"""
data_path = os.path.join('fuel_dataset', 'data.xlsx')

"""
target properties you want BlendingFuel to model
"""
target_properties = ['density', 'viscosity_0', 'viscosity_20',
'net_volume_calorific_value', 'freezing_point']

"""
x column in init_data file, [a, b)
"""
x_range = [0, 2]

train_ratio = 0.7
out_shuffle = False

# finding best fuel blending scheme

y_min_max_scale = {
    'density': {'min': 0.93, 'max': 1.07},
    'viscosity_0': {'min': 4.65, 'max': 83.20},
    'viscosity_20': {'min': 8.29, 'max': 353.50},
    'net_volume_calorific_value': {'min': 39.0, 'max': 44.45},
    'freezing_point': {'min': 158.0, 'max': 230.0},
}

target_properties_weight = {
    'density': 0.20,
    'viscosity_0': -0.20,
    'viscosity_20': -0.20,
    'net_volume_calorific_value': 0.20,
    'freezing_point': -0.20,
}

x0 = [0.33, 0.33, 0.34]

```

2 输入文件

2.1 概述

本程序完全基于python编写，为了简洁和运行效率，输入文件也采用python的格式（INPUT.py），因此在输入文件中写（多行）注释的方法与python完全一致，如：

```
# 这是注释

"""
这是多行注释
"""
```

本程序所有的行为都可以在输入文件（默认为INPUT.py）中指定，包括任务、模型类型、训练细节、优化等项

2.2 参数

2.2.1 task

可选项，如不设置程序的行为默认为1

`task` 控制程序要做的任务。

- 1：只训练预测性质的机器学习模型；
- 2：训练预测性质的机器学习模型，并搜索最优混合燃料配方（须额外设定参数）

例如：

```
task = 2
```

2.2.2 model_type

必选项，不设置程序什么也不会做，甚至可能报错

`model_type` 控制使用什么机器学习模型拟合燃料性质输入。

- 1：多元线性回归（linear regression）
- 2：多项式回归（polynomial regression）

例如：

```
model_type = 2
```

2.2.3 poly

可选项，仅当 `model_type=2` 时应该设置，此时若不设置相当于 `model_type=1`。

`poly` 控制多项式回归模型中多项式的最高次数。

例如：

```
poly = 3
```

2.2.4 data_path

必选项，不设置会报错。

`data_path` 指定程序从哪里读取数据。

例如：

```
data_path = os.path.join('fuel_dataset', 'data.xlsx')
```

2.2.5 target_properties

必选项，不设置会报错。

`target_properties` 控制程序会关注哪些数据集中的燃料性质。设置时用python中列表（list）类型的格式。

例如：

```
target_properties = ['density', 'viscosity_0', 'viscosity_20',  
                    'net_volume_calorific_value', 'freezing_point']
```

2.2.6 x_range

必选项，不设置会报错。

`x_range` 控制程序读取哪几列作为机器学习模型的输入。

例如：

```
x_range = [0, 2] # [a, b) 前闭后开区间，指第0、1行作为模型输入
```

2.2.7 train_ratio

必选项，不设置会报错。

`train_ratio` 控制模型选取原始数据集中多少百分比的数据作为训练集

例如：

```
train_ratio = 0.7
```

2.2.8 out_shuffle

必选项，不设置会报错。

`out_shuffle` 控制程序在训练模型时是否输出打乱顺序的数据集，bool。

例如：

```
out_shuffle = False
```

2.2.9 y_scale

可选项，仅当 `task = 2` 时必须设置，此时若不设置会报错。

在搜索最佳混合配方时将不同燃料性质的数值都压缩至[0, 1]区间上，因此要给出不同性质的最大、最小值。`y_scale` 控制不同性质的最大最小值。设置时用python中字典（dict）类型的格式。

例如：

```
y_scale = {
    'density': {'min': 0.93, 'max': 1.07},
    'viscosity_0': {'min': 4.65, 'max': 83.20},
    'viscosity_20': {'min': 8.29, 'max': 353.50},
    'net_volume_calorific_value': {'min': 39.0, 'max': 44.45},
    'freezing_point': {'min': 158.0, 'max': 230.0},
}
```

2.2.10 target_properties_weight

可选项，仅当 `task = 2` 时必须设置，此时若不设置会报错。

在搜索最佳混合配方时将不同燃料性质给定权重汇总为一个线性加和的单目标函数，因此要给出不同性质的权重[^注1] `target_properties_weight` 控制不同性质的权重。设置时用python中字典（dict）类型的格式。

例如：

```
target_properties_weight = {  
    'density': 0.20,  
    'viscosity_0': -0.20,  
    'viscosity_20': -0.20,  
    'net_volume_calorific_value': 0.20,  
    'freezing_point': -0.20,  
}
```

2.2.11 x0

可选项，仅当 `task = 2` 时必须设置，此时若不设置会报错。

`x0` 控制在搜索最佳混合配方时的初始位置，不同组分之和应该为1。设置时用python中列表（list）类型的格式。

例如：

```
x0 = [0.33, 0.33, 0.34]
```