

MIDAS模型代码及解读：

代码来自GitHub: [Python 中的混合数据采样 \(MIDAS\) 建模](#)

1. 安装库midaspy

```
pip install --user git+git://github.com/Yoseph-Zuskin/midaspy.git@master
```

安装时可能会遇到的问题：

1. 确保以下设置已关闭：关闭设置中网络--代理--手动设置代理，会影响库的安装
2. **conda命令配置**：找到 .condarc 文件，修改里面的内容如下：

```
show_channel_urls: true

channels:
- http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
- http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch
- http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/conda-forge/
- http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/msys2/
- http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/bioconda/
- http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/

ssl_verify: false
```

3. **安装git命令**：需要执行 `conda install git`
4. **fatal: unable to connect to github** 问题，执行以下命令：

```
git config --global url."https://github.com".insteadOf git://github.com
```


如果还是报错，再执行

```
git config --global --unset http.proxy
git config --global --unset https.proxy
```

2. 加载midaspy与Pandas库：

```
import numpy as np
import pandas as pd
from itertools import product
```

```
from midaspy.iolib import *
from midaspy.model import MIDASRegressor
```

示例中使用以下函数从 zip 文件中提取 CSV 文件并从加拿大统计局加载数据：

```
from zipfile import ZipFile
from requests import get
from io import BytesIO
def extract_zip_from_url(link):
    """
    从URL中提取ZIP
    此函数从在线找到的ZIP文件中提取字节数据

    参数
    link (str): ZIP文件的URL链接
    返回
    输出(dict):压缩文件的字节数据字典
    """
    u=get(link)
    f=BytesIO()
    f.write(u.content)
    input_zip=ZipFile(f)
    output={i:input_zip.read(i) for i in input_zip.namelist()}
    return output
```

```
def load_stats_can_data(pid,low_memory=False):
    """
    加拿大统计数据下载
    自定义函数直接从在线zip文件加载数据到pandas的dataframe

    参数
    pid (str):the Statistics Canada online data asset的 产品id
    Low_memory (bool):记住设置读取的为csv文件 memory setting for the pandas.read_csv
    function

    返回
    df (pandas.DataFrame):来自 the data asset的所有变量的dataframe

    """
```

```
# 首先使用extract_zip_from_url从加拿大统计局网站提取zip文件
# first extract the zip file from the Statistics Canada website using
extract_zip_from_url
    online_zip=extract_zip_from_url('https://www150.statcan.gc.ca/n1/tbl/csv/{}-
eng.zip'.format(pid))
# 然后下载数据资产的结果CSV文件，并将日期设置为datetime64[ns]数据类型
# then load the resulting csv file for the data asset and set the date to
datetime64[ns] data type
    df=pd.read_csv(BytesIO(online_zip['{}.csv'.format(pid)]),parse_dates=
[0],low_memory=low_memory)
# 筛选数据资产以排除与领域相关的数据
df=df[df.iloc[:,1]=='Canada'] # filter the data asset to exclude data related
to the territories
# 定义变量类型列表，如第4列所示
variables=list(set(df.iloc[:,3])) # define list of variable types as they
appear in the 4th column
# 将每一系列变量储存在这一字典中
variable_series={} # store each variable series in this temporary dictionary
instance
# 遍历data asset中所有变量
for v in variables: # iterable over all the variables in this data asset
    series=df[df.iloc[:,3]==v].set_index(df.columns[0]).VALUE # filter
dataframe for the variable
    series.rename(v,inplace=True) # rename the series to what Statistics Canada
refers to it as
    variable_series[v]=series # store the resulting series in the temporary
dictionary instance
df=pd.DataFrame() # create dataframe to store all combined series
for v in variables: # iterate over each variable and merge it as this works
when pd.concat may fail

df=pd.merge(right=df,left=variable_series[v],right_index=True,left_index=True,how='
outer')
return df
```

2.2 带有领先项的 MIDAS 模型

由于高频数据发布总是比低频数据早，在预测低频数据时可以将已经发布的高频数据加入到预测方程中，可以得到更好的预测结果，与预测（forecast）相对，这种预测方法称为“即时预测”（nowcast）。假设我们在 $t+1$ 季度中的第 3 个月的第一天，那么使用 $t+1$ 季度前两个月发布日度数据来预测 $t+1$ 季度的待预测数据就是“即时预测”。

(Giannone, Reichlin and Small 2008)基于动态因子状态空间模型和 Kalman 滤波方法率先实现了即时预测，将最新发布的数据用于更新预测结果，提升预测精度。(Ghysels, et al. 2009, Andreou, et al. 2010a)发现使用带有领先项的 MIDAS 同样可以完成即时预测，而且相比(Giannone, et al. 2008)的方法，MIDAS 有两大优势：第一，MIDAS 不需要像状态空间模型那样设定系统方程的形式，避免出现模型设定误差；第二，MIDAS 是一种节省参数的方法，待估参数远远比 kalman 滤波少，计算压力更小。

带有领先项的 MIDAS 模型有如下形式：

$$Y_{t+1}^Q = \mu + \sum_{i=0}^{p_F-1} \beta_i^F F_{t-i}^Q + \sum_{j=0}^{p_Y^Q-1} \mu_{j+1} Y_{t-j}^Q + \beta \left[\sum_{i=0}^{J_X^D-1} w_{J_X^D-i} (\theta^D) X_{J_X^D-i,t+1}^D + \sum_{j=0}^{p_X^D-1} \sum_{l=0}^{N_D-1} w_{J_X^D+N_D-l+j \cdot N_D} (\theta^D) X_{N_D-l,t-j}^D \right] + u_{t+1} \quad (10)$$

式(10)记作 $FADL-MIDAS(p_F, p_Y^Q, p_X^D, J_X^D)$ 模型，当 $t+1$ 季度已经过去两个月， J_X^D 的大小就是两个月的交易天数。