下载股票的历史日交易数据并存入数据库——基于tushare

作者 竹间为简 (/u/a508c9751b83) + 关注
2016.07.14 15:07* 字数 864 阅读 3737 评论 6 喜欢 8 阅读 3737 评论 6 喜欢 8
(/u/a508c9751b83)

> 1. tushare (http://tushare.waditu.com/trading.html#id2)是一个非常神奇的Python模块包，基于新浪的API，可提供并不限于股票的历史数据。
> 2. 数据库选用的是sqlite3，单文件，轻量化，不需要配置。

以下是完整代码，且使用的是多线程的方式。此处提到的多线程的方法可以参考Python黑魔法，一行实现并行化 (http://dataunion.org/24295.html)这篇文章，讲的很好。

准备工作

```
import tushare as ts
from sqlalchemy import create_engine #注1
import sqlite3
import pandas as pd #注2
from multiprocessing.dummy import Pool as ThreadPool #注3

conn1=sqlite3.connect('Stocklist.db') #注4
engine = create_engine('sqlite:///History.db', echo = False) #注5

conn2 = sqlite3.connect('History.db')
cur2=conn2.cursor() #注6

stocklist = []
errorlist = []
alreadylist = []
cur1=conn1.cursor()
query1 = "select * from Allist" #注7
cur1.execute(query1) #注8
stocklist = cur1.fetchall() #注9
cur1.close()
conn1.close()
query2 = "select name from sqlite_master where type='table' order by name" #注10
alreadylist = pd.read_sql(query2, conn2) #注11
```

注1： `sqlalchemy` 是Python自带的与数据库联结的包，导入创建数据库联结的函数

注2：导入pandas包，Python上的科学计算用的包

注3：多线程

注4：*Stocklist.db*是存放股票列表的数据库

注5：创建与sqlite数据库的联结，名字为*History.db*

注6：创建一个游标

注7：SQL语句。 `Allist` 是*Stocklist.db*中的表，存放股票列表的。

注8：执行SQL语句

注9：获取执行SQL查询后的结果， `stocklist` 是tuple类型

注10：意思是获取所有表名

注11：另一种读数据库的方法，直接用pandas读取数据库， `alreadylist` 是DataFrame类型，有代码的那一列名为 `name`

获取数据并保存的函数

```
def save(stock):
    code = stock[0][:6] #注1
    if code not in list(alreadylist.name): #注2
        marketday = stock[1] #注1
        i= 0
        try:
            startday = pd.Timestamp(marketday)
            df = ts.get_h_data(code, start=str(startday)[:10], retry_count = 5)
            df = df.sort_index(ascending=True) #注4
            ma_list = [5,10,20,60]
            for ma in ma_list:
                df['MA_' + str(ma)] = pd.rolling_mean(df.close, ma) #注5
            df.to_sql(code, engine, if_exists='append') #注6
        except:
            errorlist.append(stock[0])
    print errorless  #注7
```

该函数的思路是这样的：

1. 利用tushare的 `get_h_data` 函数获取数据。
2. 由于会出现如网络错误或其他错误，导致该程序重新执行，所以必须验证以防止添加重复数据。

注1： `stock` 取自上文的 `stocklist` ， 由于是tuple，含有两列，第一列取做 `code` ， 第二列取作 `marketday` ，后者是该股票的上市日。
注2：上文中用SQL语句查询出了一个DataFrame `alreadlist` ， 包含了*History.db*数据库中已有的表名，用 `alreadlist.name` 取出， `name` 是 `alreadlist` 的列名。
注3： `pd.Timestamp` 可以把文本类型的日期转成时间戳类型的，这样就可以进行时间的运算，例如通过 `pd.Timedelta` 。然后就照 `startday` 和 `enday` 的写法，三年一个跨度拉取数据。
注4：数据拉过来是以 `date` 为索引的，但是还需要重新排序，因而这样写以升序排列。
注5：没有移动均线的数据，因而手动计算。pandas直接自带移动平均数的计算函数 `pd.rolling_mean` ， 两个参数分别是*计算对象*和*计算参数*。
注6：写入数据库， `if_exists='append'` 意为追加的形式。
注7：用 `try...except` 的方式来避免异常中断，错误的股票写入 `errorlist` ， 最后程序结束时打印出来。

多线程处理

```
pool = ThreadPool(4)
try:
    pool.map(save, stocklist)
except:
    pool.map(save, stocklist)
f = open('Notsaved.txt', 'w')
print >> f, errorlist
f.close()
pool.close()
pool.join()
```

注： `pool.map(save, stocklist)` 意思就是从 `stocklist` 中取每一个元素送入 `save` 的函数中运行。最后把上段代码的 `errorlist` 打印成文件。

每日的更新

```
import tushare as ts
from sqlalchemy import create_engine
import sqlite3
import pandas as pd
from datetime import datetime as dt

con = sqlite3.connect('History.db')
query1 = "select name from sqlite_master where type='table' order by name"
stocklist = pd.read_sql(query1, con).name

engine = create_engine('sqlite:///History.db', echo = False)

updatestock = []
for stock in stocklist:
    query2 = "select * from '%s' order by date" %stock
    df = pd.read_sql(query2, con)
    df = df.set_index('date')
    if dt.now().weekday() == 5: #注1
        today = str(pd.Timestamp(dt.now())-pd.Timedelta(days = 1))[:10]  #注2
    elif dt.now().weekday() == 6:
        today = str(pd.Timestamp(dt.now())-pd.Timedelta(days = 2))[:10]
    else:
        today = str(pd.Timestamp(dt.now()))[:10]
    if today != df.ix[-1].name[:10]:
        try:
            df = ts.get_h_data(stock, start = df.ix[-1].name[:10], retry_count =
5)
            df.to_sql(stock, engine, if_exists='append')
            updatestock.append(stock)
        except:
            continue

f = open('updated.txt','w')
print >>f, updatestock
f.close()
```

注1： `dt.now()` 是指今天， `dt.now().weekday` 是返回今天是星期几，5代表星期六，6代表星期天。

注2： `today` 指的是最近的一个交易日， `df.ix[-1].name` 是数据库中最新的一天， `if today != df.ix[-1].name[:10]` 意思就是，如果数据库最新的一天不是最近一个交易日，则要开始更新数据。

清洗数据库

```
import pandas as pd
import sqlite3
from multiprocessing.dummy import Pool as ThreadPool

con = sqlite3.connect('History.db')
query1 = "select name from sqlite_master where type='table' order by name"
stocklist = pd.read_sql(query1, con).name

delstock = []
f = open('Deleted.txt', 'w')
for stock in stocklist:
    query2 = "select * from '%s' order by date" %stock
    df = pd.read_sql(query2, con)
    cur=con.cursor()
    query4 = "delete from '%s' where rowid not in(select max(rowid) from '%s' gro
up by date)" %(stock, stock) #注1
    cur.execute(query4)
    con.commit()

con.close()
print >> f, delstock
f.close()
```

注1：这句SQL语句的意思是以 `date` 分组，删除重复的行

注2：最后执行cur.execute(...)完后要 `con.commit()` 提交，才能有效

竹间为简 (/u/a508c9751b83)

写了 26560 字，被 80 人关注，获得了 159 个喜欢

+ 关注

写了 265697 字，被 80 人关注，获得了 159 个喜欢

经济学科班，爱生活，爱Python，爱macOS/iOS，互联网金融风控。跨界狂魔。

如果觉得我的文章对您有用，请随意赞赏。您的支持将鼓励我继续创作！

赞赏支持

♡ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) 8

更多分享

(http://cwb.assets.jianshu.io/notes/images/4789051/

(/sign后发表评论in?utm_source=desktop&utm_medium=not-signed-in-comment-form)

6条评论 只看作者 按喜欢排序 按时间正序 按时间倒序

6889aa6963e1 (/u/6889aa6963e1)
2楼 · 2017.02.03 22:03
(/u/6889aa6963e1)
很好,文章写得非常好,让我学到很多知识,您是实现了程序化交易吗

赞 回复

> 竹间为简 (/u/a508c9751b83)： @6889aa6963e1 (/users/6889aa6963e1) 没有，程序化交易的水太深了，这仅仅是万里长征的第一个半步，推荐去 优矿 上看，你就知道有多复杂
> 2017.02.03 22:48 回复

添加新评论

Nora_Jianshu (/u/06ff1db1651f)
3楼 · 2017.02.18 11:42
(/u/06ff1db1651f)
errorless?

赞 回复

有梦想的咸鱼_7815 (/u/09d129d27170)
4楼 · 2017.03.09 21:12
(/u/09d129d27170)
厉害了，word哥，正想用tushare做一个数据库，看来直接copy就可以了。

赞 回复

cyberortrig (/u/b49f801555d2)
5楼 · 2017.03.15 15:46
(/u/b49f801555d2)
line 18, in <module>
cur1.execute(query1) #8
sqlite3.OperationalError: no such table: Allist
这是什么原因

赞 回复

> cyberortrig (/u/b49f801555d2)： 需要预先创建Stocklist.db对吧
> 怎么创建能说下吗
> 还有History.db

2017.03.15 15:54 回复

添加新评论

被以下专题收入，发现更多相似内容


stock (/c/57a6d887b220?utm_source=desktop&utm_medium=notes-included-collection)

推荐阅读 更多精彩内容 > (/)

买了OmniFocus (/p/1cceabe297dc?utm_campaign... (/p/1cceabe297dc?utm_campaign=maleskine&utm_content=note&utm_

终于狠狠心咬咬牙买了OmniFocusforiOS版，因为觉得这几个月以来拖延症太严重，做事效率极低。再加上未来有好几件大事要并行做，必须要依靠一个十分

竹间为简 (/u/a508c9751b83?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

用Automator实现网页标题自动存为Markdown文字超... (/p/40d9b0961317?utm_campaign=maleskine&utm_content=note&utm_

众所周知，当用Markdown引用网页的时候，最好采取文本超链接形式（题目附上链接），例如下述：MacOS/iOS自动化指南-专题但是要人工操作其实挺麻

竹间为简 (/u/a508c9751b83?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

地球上最好看的100部电影（下） (/p/ce715c01a604... (/p/ce715c01a604?utm_campaign=maleskine&utm_content=note&utm_

（前面50部，请看前篇：地球上最好看的100部电影 - 上） 最近一段时间，地球也进入了信息文明的初级阶段。网上可下载的电影也很多，可是老电影却很少

Graceland (/u/ae8a590a2c9f?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

林肯公园主唱自杀：你永远不知道，有些人为什么痛哭 ... (/p/8be1f1c7b893?utm_campaign=maleskine&utm_content=note&utm_

凌晨两点多，BBC发出报道：美国知名乐队林肯公园主唱查斯特·贝宁顿在家中上吊自杀，年仅41岁。 多年来，林肯公园载誉无数，他们曾5次获得全美音乐

槽值 (/u/ad73e614982f?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

这5本书将彻底颠覆你的认知 (/p/3133b2b3d400?ut... (/p/3133b2b3d400?utm_campaign=maleskine&utm_content=note&utm_

文|麦家理想谷 本文原创，转载请联系 / 《乌合之众》 居斯塔夫·勒庞 著 豆瓣评分8.3分，一本研究人类的群体行为不可不读的佳作。 没有一片雪花会认为是自

麦家理想谷 (/u/009eac2d558e?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)