第十三节课 数据采集与时间序列模型

https://github.com/cystanford 老师的GitHub

第十节课笔记

讲义内容概括	知识点	Question?

我们从哪些维度可以收集数据?
1、开放数据源 一般针对行业的数据库，比如美国人口调查局开放了美国的人口信息、地区分布和教育情况数据。除了政府外，企业和高校也会开放相应的大数据 很多研究都是基于开放数据集进行的，比如MovieLens， Netflix Prize DataSet，LETOR，MSLR，Yahoo Learning to Rank数据集等
2、爬虫抓取爬虫工具，可视化易操作 爬虫代码，自己编写，抓取特定网站或App 3、传感器 比如无人驾驶的传感器数据，新零售的传感器数据，采集的是物理信息（图像、视频、或者某个物体的速度、热度、压强等）
4、日志采集 用于统计用户的操作。可以在前端进行埋点，或在后端进行脚本收集、统计，来分析网站的访问情况，使用瓶颈等

开放数据源

单位	数据源	网址
美国人口调查局	提供人口信息，地区分布和教育情况等美国公民相关的数据	http://www.census.gov/data.html
欧盟	欧盟开放数据平台，提供欧盟各机构的大量数据，	http://open-data.europa.eu/en/data/
Facebook	Facebook官方提供的API，用于查询用户公开的海量信息	https://developers.facebook.com/docs/graph-api
Amazon	亚马逊网络服务开放数据集	http://aws.amazon.com/datasets
Google	谷歌金融，收录了40年以来的股票数据，实时更新	https://www.google.com/finance
北京大学	北京大学开放研究数据平台	http://opendata.pku.edu.cn/
ImageNet	目前世界上图像识别最大的数据库，包括近1500万张图像	http://www.image-net.org/

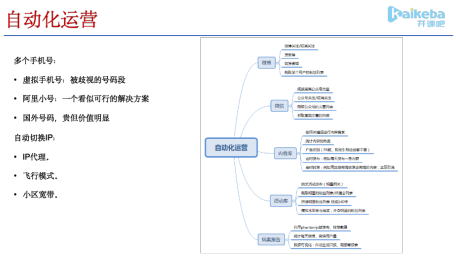
数据集	说明	网址
MovieLens	电影推荐系统数据集，包括多个大小的版本	https://grouplens.org/datasets/movielens/
Netflix Prize DataSet	1亿部电影评分，Netflix悬赏100万美金的知名数据集	https://www.netflixprize.com/
LETOR	信息检索数据集	https://www.microsoft.com/en-us/research/project/letor-learning-rank-information-retrieval/
MSLR	微软发布的Learning to Rank数据集	https://www.microsoft.com/en-us/research/project/mslr/
Yahoo LTR	雅虎发布的LTR比赛数据集	http://webscope.sandbox.yahoo.com/

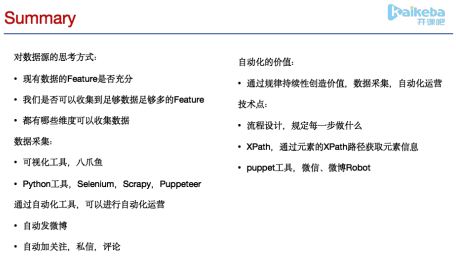
爬虫抓取工具
火车采集器
14年历史，老牌采集工具。不仅可以做抓取工具，也可以做数据清洗、数据分析、数据挖掘、可视化等工作。数据源适用于绝大部分的网页，网页中能看到的内容都可以通过采集规则进行抓取
搜集客
完全可视化操作，无需编程。采集过程所见即所得，抓取结果信息、错误信息等都反应在软件中。没有流程的概念，用户只需要关注抓取什么数据，而流程细节完全交给集搜客来处理。所有爬虫需要在用户自己电脑上跑
八爪鱼
知名采集工具，分为免费采集模板，云采集（付费）。 免费的采集模板实际上就是内容采集规则，包括了电商类、生活服务类、社交媒体类、论坛类的网站都可以采集，用起来非常方便。也可以自己来自定义任务。 云采集，就是当你配置好采集任务，就可以交给云端进行采集。八爪鱼一共有5000台服务器，通过云端多节点并发采集，采集速度远远超过本地采集。此外还可以自动切换多个 IP，避免IP被封影响采集
Python爬虫工具：

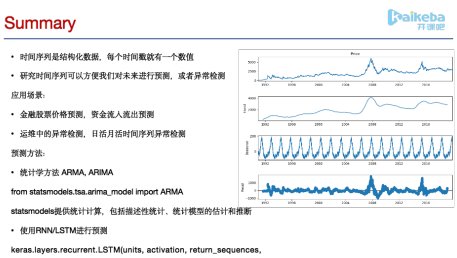
- Selenium，Web应用的自动测试工具，可以直接运行在浏览器中，原理是模拟用户在进行操作，支持当前多种主流的浏览器
  - Selenium工具：from selenium import webdriver 下载对应版本的ChromeDriver http://chromedriver.storage.googleapis.com/index.html 将下载的ChromeDriver放到某目录下，Python程序中指定chrome_driver路径即可 driver=webdriver.Chrome(executable_path=chrome_driver) WebDriver实际上就是Selenium 2，是一种用于Web应用程序的自动测试工具，提供了一套方便操作的API 通过WebDriver创建一个Chrome浏览器的drive，再通过drive获取访问页面的完整HTML
- Scrapy，Python爬虫框架，把基础爬虫功能抽象出来做成的脚手架，Selenium可以应用在Scrapy爬虫框架中
- Puppeteer，可以控制Headless Chrome，即无界面的自动化测试框架，支持模拟键盘输入、截图、表单提交等特殊场景操作，Google2018年推出的爬虫神器

数据采集
如何对数据Feature进行思考
数据源都有哪些维度
Project A：安居客房价抓取
Project B：抓取微博上关于LightGBM的内容
如何使用八爪鱼进行数据采集
Python爬虫工具
如何使用selenium自动化抓取数据

时间序列模型
Project C：对沪市指数走势进行预测
Project D：对北京PM2.5值进行预测
Project E：对北上广房价进行预测
什么是时间序列预测 AR、MA、ARMA、ARIMA模型
使用ARMA/ARIMA对沪市指数进行预测
使用LSTM进行预测







传感器
传感器采集是基于特定的设备，采集和交互信息，包括图像，语音，温度，重量，测速
AIot（人工智能物联网）=AI（人工智能）+Iot（物联网）
场景：智能家居，医疗，会议，交通等
美的的iot战略

日志采集
最大的作用，就是分析用户访问情况，对用户历史行为进行挖掘，同时能及时发现系统承载瓶颈，提高系统负载量，方便基于用户实际用户访问需求进行优化 日志采集是运维人员的重要工作，可以采集用户访问网站的全过程：哪些人在什么时间，通过什么渠道（比如搜索引擎、网址输入）来过，都执行了哪些操作；系统是否产生了错误；甚至包括用户的IP、HTTP请求的时间，用户代理等。这些日志数据可以被写在一个日志文件中，也可以分成不同的日志文件，比如访问日志、错误日志等
日志采集可以分两种形式通过Web服务器采集，例如 httpd、Nginx、Tomcat 都自带日志记录功能。同时很多互联网企业都有自己的海量数据采集工具，多用于系统日志采集，如Hadoop的Chukwa、Cloudera的Flume、Facebook的Scribe等，这些工具均采用分布式架构，能够满足每秒数百MB的日志数据采集和传输需求 自定义采集用户行为，例如用JavaScript代码监听用户的行为、AJAX异步请求后台记录日志等，比如Google Analysis
一般Web服务器会自带日志功能，也可以使用Flume从不同的服务器集群中采集、汇总和传输大容量的日志数据

自动化运营
自动化运营的步骤：
• 多个手机号

尽管早期注册只需要邮箱就可以，但现在账号注册都需要绑定手机号，所以手机号是必备的

• 多个IP

很多社交网站都会有反垃圾的措施，共用同一个IP，一定会被封号

• 模拟操作

因为需求是个性化的，所以在这一步，可以封装出一些基本的操作，比如关注、发布动态、转发、阅读文章等

多个手机号：
- 虚拟手机号：被歧视的号码段
- 阿里小号：一个看似可行的解决方案
- 国外号码，贵但价值明显

自动切换IP：
- IP代理。
- 飞行模式。
- 小区宽带。

时间序列：
建立了观察结果与时间变化的关系，能帮预测未来一段时间内的结果变化情况

时间序列分析与回归分析的区别

机器学习模型，包括AR、MA、ARMA、ARIMA
神经网络模型，用LSTM进行时间序列预测

RNN是什么?

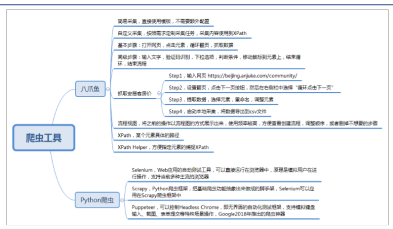
爬虫爬取的数据怎么存入结构性数据库中?

IOT方面有哪些可以学习的东西?

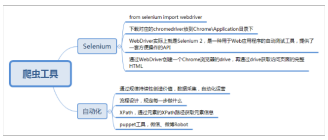
日志采集怎么用?

Summary

Summary



Summary



Summary

Summary

Thinking1：当我们思考数据源的时候，都有哪些维度，如果你想要使用爬虫抓取数据，都有哪些工具

Thinking2：今天讲解了时间序列预测的两种方式，实际上在数据库内建时间属性后，可以产生时序数据库，请思考什么是时序数据库？为什么时间序列数据成为增长最快的数据类型之一（请思考并分享到班级微信群中）

Thinking3：开源是当前重要的Trend，我们使用的statsmodels.tsa，tensorflow/keras都是开源工具你都知道有哪些和AI相关的开源工具？ 阿里，微软，百度 都有哪些和AI相关的开源工具（包括LightGBM） 了解和使用这些工具，对于我们有哪些价值？ （请思考并分享到班级微信群中）

Action：房价预测全国各城区房价走势，及未来3个月房价预测 基础版：使用开放数据集进行预测

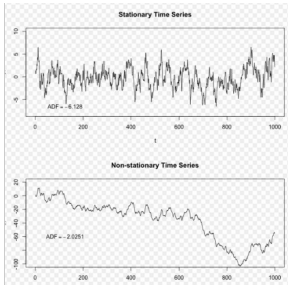
时间序列及分解：

- 平稳序列，stationary series

基本上不存在趋势（Trend）的序列，各观察值基本上在某个固定的水平上波动

- 非平稳序列，non-stationary series

包含趋势、季节性或周期性的序列，可以只有一种成分，也可能是多种成分的组合



时间序列及分解：

- 趋势（trend）：时间序列在长时期内呈现出来的某种持续上升或持续下降的变动，也称长期趋势
- 季节性（seasonality）：时间序列在一年内重复出现的周期波动。销售旺季，销售淡季，旅游旺季、旅游淡季

季节，可以是任何一种周期性变化，不一定是一年中的四季

含有季节成分的序列可能含有趋势，也可能不含有趋势

- 周期性（cyclicity）：通常是由经济环境的变化引起

不同于趋势变动，不是朝着单一方向的持续运动，而是涨落相间的交替波动

不同于季节变动，季节变动有比较固定的规律，变动周期大多为一年。周期性的循环波动无固定规律，变动周期多在一年以上，且周期长短不一

- 随机性（Irregular），指受偶然因素影响所形成的的不规则波动，在时间序列中无法预估

随机性是不规则波动，除去趋势、周期性、季节性的偶然性波动

因素	举例
长期趋势 Trend(T)	国内生产总值
季节变动 Season(S)	冰淇淋、暖宝宝、羽绒服、裙子等销售
周期性 Cyclic(C)	太阳黑子数量变化
随机性 Irregular(I)	股票市场受到突然的利好、利空等信息的影响，影响股价产生的波动

statsmodels工具：

statsmodels工具包提供统计计算，包括描述性统计以及统计模型的估计和推断 statsmodels主要包括如下子模块： 回归模型：线性回归，广义线性模型，线性混合效应模 方差分析（ANOVA） 时间序列分析：AR，ARMA，ARIMA等

```
import statsmodels.api as sm
```

AR模型：

Auto Regressive，中文叫自回归模型

认为过去若干时刻的点通过线性组合，再加上白噪声(可以把它理解为一个期望为0，方差为常数的纯随机过程)就可以预测未来某个时刻的点

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + u_t$$

ϕ

表示p阶的自回归过程，　　为自回归系数

MA模型：

Moving Average，中文叫做滑动平均模型

与AR模型大同小异，AR模型是历史时序值的线性组合，MA是通过历史白噪声进行线性组合来影响当前时刻点

$$x_t = u_t + \phi_1 u_{t-1} + \phi_2 u_{t-2} + \ldots + \phi_q u_{t-q}$$

ϕ　　u_t

MA(q)表示q阶移动平均过程，　　为移动回归系数，为不同时间点的白噪声

ARMA模型：

Auto Regressive Moving Average，中文叫做自回归滑动平均模型

AR模型和MA模型的混合，相比AR模型和MA模型，它有更准确的估计

ARMA模型存在p和q两个阶数，称为ARMA(p,q)模型：

$$x_t = u_t + \phi_1 u_{t-1} + \phi_2 u_{t-2} + \ldots + \phi_q u_{t-q} + \vartheta_1 x_{t-1} + \vartheta_2 x_{t-2} + \ldots + \vartheta_p x_{t-p}$$

自回归模型结合了两个模型的特点，AR解决当前数据与后期数据之间的关系，MA则可以解决随机变动，即噪声问题

ARIMA模型：

自动化运营：

内容上：采集数据策略—爬虫入库—按一定策略筛选信息—发布

运营上：发现资源收集的流程（比如涨粉）

这些方面还有没有其他可以深化的？

时间序列……怎么感觉AR，MA，ARMA，ARIMA都比较基础呢……

LSTM还需要进一步了解！

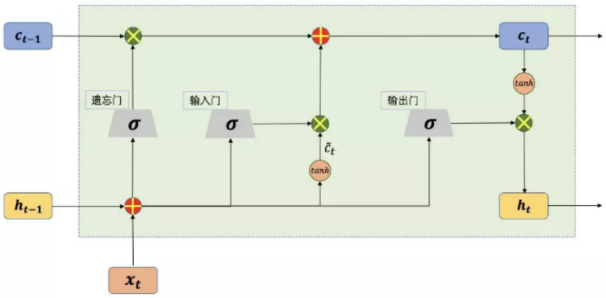
https://github.com/shujingoj/creprice_HousePriceData 说明使用的预测模型，可视化结果 升级版：自己抓取数据，可以采用抓取工具，或Python爬虫（Selenium，Scrapy，Puppeteer）
https://www.anjuke.com/fangjia/beijing2010/chaoyang/ 加强版： 前端可视化界面，比如flask 分组完成（没有找到组的，找班主任划分） 鼓励交流，数据处理 or 模型使用（助教）

Auto Regressive Integrated Moving Average模型，中文叫差分自回归滑动平均模型，也叫求合自回归滑动平均模型
相比于ARMA，ARIMA多了一个差分的过程，作用是对不平稳数据进行差分平稳，在差分平稳后再进行建模
ARIMA的原理和ARMA模型一样。
相比于ARMA(p,q)的两个阶数，ARIMA是一个三元组的阶数(p,d,q)，称为ARIMA(p,d,q)模型，其中d是差分阶数

ARIMA模型步骤：
Step1，观察时间序列数据，是否为平稳序列
Step2，对于非平稳时间序列要先进行d阶差分运算，化为平稳时间序列
Step3，使用ARIMA（p, d, q）模型进行训练拟合，找到最优的(p, d, q)，及训练好的模型
Step4，使用训练好的ARIMA模型进行预测，并对差分进行还原 ARIMA用差分将不平稳数据先变得平稳，再用ARMA模型

ARMA工具：from statsmodels.tsa.arima_model import ARMA
ARMA(endog,order,exog=None)
endog：endogenous variable，代表内生变量
order：代表是p和q的值，也就是ARMA中的阶数
exog：exogenous variables，代表外生变量。
fit函数，进行拟合predict(start, end)函数，进行预测，其中start为预测的起始时间，end为预测的终止时间

LSTM：
LSTM，Long Short-Term Memory，长短记忆网络 可以避免常规RNN的梯度消失，在工业界有广泛应用
引入了三个门函数：输入门（Input Gate）、遗忘门（Forget Gate）和输出门（Output Gate）来控制输入值、记忆值和输出值
输入门决定了当前时刻网络的状态有多少信息需要保存到内部状态中，遗忘门决定了过去的状态信息有多少需要丢弃 => 输入门和遗忘门是LSTM能够记忆长期依赖的关键
输出门决定当前时刻的内部状态有多少信息需要输出给外部状态。



使用LSTM进行时间序列预测

TimeDistributed的使用：

TimeDistributed是Keras中的包装器

model = Sequential()

将模型的shape由(None, 10,16) 变成 (None, 10, 8)

model.add(TimeDistributed(Dense(8), input_shape=(10, 16)))

TD层和Dense配合使用，主要应用于一对多，多对多的情况

input_shape = (10,16)，表示步长是10，特征维度为16，

首先使用TimeDistributed(Dense(8), input_shape = (10,16))把每一步的维度为16变成8，不改变步长的大小

若该层的batch输入shape为(50, 10, 16)，则这一层之后的输出shape为(50, 10, 8)

RepeatVector的使用：

RepeatVector(n) 将输入重复n次

model = Sequential()

原来模型的output_shape = (None, 32)，`None` 是batch 维度

model.add(Dense(32, input_dim=32))

使用RepeatVector，将模型output_shape 设置为 (None, 3, 32)

model.add(RepeatVector(3))

如果输入的形状为(None,32)，经过添加RepeatVector(3)层之后，输出变为(None,3,32), RepeatVector不改变我们的步长，改变我们的每一步的维数（即：属性长度）

TimeDistributed的使用：TimeDistributed是Keras中的包装器 model = Sequential() # 将模型的shape由(None, 10,16) 变成 (None, 10, 8) model.add(TimeDistributed(Dense(8), input_shape=(10, 16))) TD层和Dense配合使用，主要应用于一对多，多对多的情况 input_shape = (10,16)，表示步长是10，特征维度为16， 首先使用TimeDistributed(Dense(8), input_shape = (10,16)) 把每一步的维度为16变成8，不改变步长的大小 若该层的batch输入shape为(50, 10, 16)，则这一层之后的输出shape为(50, 10, 8)

RepeatVector的使用：RepeatVector(n) 将输入重复n次 model = Sequential() # 原来模型的output_shape = (None, 32)，`None` 是batch维度 model.add(Dense(32, input_dim=32)) # 使用RepeatVector，将模型output_shape 设置为 (None, 3, 32) model.add(RepeatVector(3)) 如果输入的形状为(None,32)，经过添加RepeatVector(3)层之后，输出变为(None,3,32), RepeatVector不改变我们的步长，改变我们的每一步的维数（即：属性长度）

总结Summary

- 记录作业内容
  - Thinking1：当我们思考数据源的时候，都有哪些维度，如果你想要使用爬虫抓取数据，都有哪些工具
  - Thinking2：今天讲解了时间序列预测的两种方式，实际上在数据库内建时间属性后，可以产生时序数据库，请思考什么是时序数据库？为什么时间序列数据成为增长最快的数据类型之一（请思考并分享到班级微信群中）
  - Thinking3：开源是当前重要的Trend，我们使用的statsmodels.tsa，tensorflow/keras都是开源工具你都知道有哪些和AI相关的开源工具？ 阿里，微软，百度 都有哪些和AI相关的开源工具（包括LightGBM） 了解和使用这些工具，对于我们有哪些价值？ （请思考并分享到班级微信群中）
  - Action：房价预测全国各城区房价走势，及未来3个月房价预测
    - 基础版：使用开放数据集进行预测 https://github.com/shuijingoj/creprice_HousePriceData 说明使用的预测模型，可视化结果
    - 升级版：自己抓取数据，可以采用抓取工具，或Python爬虫（Selenium，Scrapy，Puppeteer） https://www.anjuke.com/fangjia/beijing2010/chaoyang/
    - 加强版： 前端可视化界面，比如flask 分组完成（没有找到组的，找班主任划分） 鼓励交流，数据处理 or 模型使用（助教）
-