北京理工大学
大数据处理技术课程展示

——用户行为预测

答辩人：苗雨 王昱 董思超 刘琦玥 沈雨婷

指导老师：张华平

时间：2022/4/17

目录

01

用户行为预测综述

Overview of user behavior prediction

汇报人：苗雨

用户行为预测综述

什么是用户行为预测？

用户行为预测综述

用户行为预测可以理解为个性化推荐，其本质就是预测用户的行为并为其进行个性化推荐。

用户行为预测综述

对用户行为数据进行采集分析，以建立用户画像，再根据实际问题需求将数据代入相应的算法模型进行运算。

用户行为预测综述

发展历程

用户行为预测综述

数学理论基础难以实现

Before. 20th century

Since. 1970s

关系数据库出现
商业智能兴起

互联网的崛起数据量飙升

Late 20th century —— 21st century

Since. 2004

数据大爆炸时代开始

用户行为预测综述

举个例子

#美团被曝杀熟外卖会员#　分享　申请主持人

阅读5.8亿　讨论4.6万

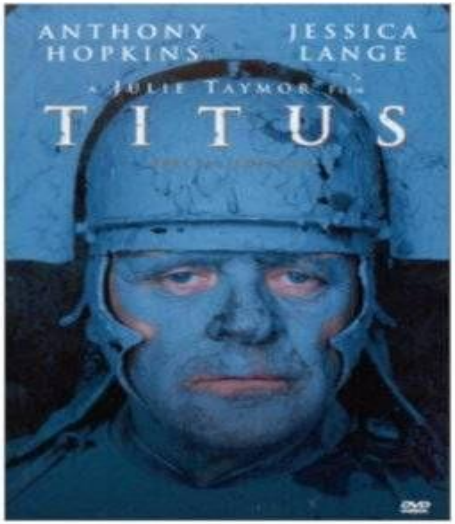
综合　实时　热门　视频

导语：近日，自媒体"漂移神父"发布文章《我被美团会员割了韭菜》爆料称，在美团上的同一家店铺，同一配送地址，同一时间点单，会员账号的配送费仍为6元，而非会员账号仅为2元。此外，不仅是一家店这种情况，一部开通美团外卖会员的手机，附近几乎所有外卖商户的配送费都要超出非会员配送费1~5元不等。

软件会根据用户的消费习惯来给产品定价，以增加利润。咱们管这个现象叫大数据杀熟。

用户行为预测综述

2000年9月，亚马逊选择了68种畅销DVD进行试验，根据潜在用户的人口统计资料、购物历史、上网行为等，对这些DVD光盘进行差别定价。其中，亚马逊将名为《Titus》的光盘，定价为新用户需要22.74美元，而老用户则需要26.24美元。亚马逊认为老用户的购买意愿更强，因此给出更高的价格，从而提高销售利润。



Titus

Anthony Hopkins (Actor), Jessica Lange (Actor), Julie Taymor (Director, Writer) | Rated: R | Format: DVD

433 ratings

Blu-ray	DVD	VHS Tape
from $75.88	$31.62	from $69.00

Additional DVD options	Edition	Discs	Price	New from	Used from
DVD	—	1	$12.00	$12.00	—
DVD	—	1	$20.54	$20.54	$14.49
DVD April 18, 2006	Special	2	$62.95	$56.95	$21.87
DVD August 15, 2000	DVD Video	2	$31.62	$75.00	$3.70

See More

用户行为预测综述

BBC NEWS

You are in: Business

Front Page
World
UK
UK Politics
Business
Market Data
Economy
Companies
E-Commerce
Your Money
Business Basics
Sci/Tech
Health
Education
Entertainment
Talking Point
In Depth
AudioVideo

Friday, 8 September, 2000, 17:07 GMT 18:07 UK

Amazon's old customers 'pay more'

Some Amazon customers are refusing to accept some DVD prices

By BBC News Online internet reporter Mark Ward

Search BBC News Online

GO

Advanced search options

Launch console for latest audio/video

BBC RADIO NEWS
BBC ONE TV NEWS
WORLD NEWS SUMMARY
BBC NEWS 24 BULLETIN
PROGRAMMES GUIDE

See also:

- 28 Aug 00 | Business
  IT giants in e-book deal
- 23 Aug 00 | Business
  Amazon to sell cars
- 06 Aug 00 | Business
  Amazon condemned as

用户行为预测综述

大数据杀熟主要是依靠算法来实现的。平台会根据用户信息构建画像，去进一步分析预测，更新用户推荐。

当用户对平台产生了信任度和黏性以后，再把价格提高，这就是所谓的**大数据杀熟**。

用户行为预测综述

这是如何实现的呢？

程序员通过算法获取用户资料、分析数据、构建用户画像，通过算法实现差异化定价完成价格歧视。这就是一种用户行为预测。

用户行为预测综述

什么是用户行为?

用户行为预测综述

用户行为由最简单的五个元素构成：时间、地点、人物、交互、交互的内容——我们称之为事件。

对用户行为分析，要将其定义为各种事件

用户行为预测综述

温馨提示：您点主食了吗？ 去添加 ›

折 十翅桶可乐餐 ¥90 ¥49.9

十翅桶可乐餐

x1

配送费 ¥9 ¥6

活动减3元配送费

美团红包 2张可用 ›

券 商家代金券 暂无可用 ›

优惠规则 已优惠¥43.1 小计 ¥55.9

历史搜索

肌肉小王子 鸡肉肠 鱼肉肠无淀粉 蛋清

鸡蛋罐头 耳环 玫瑰耳环 指套篮球

搜索发现

塑料擀面杖 面棍 鳕鱼鱼肠 鸡蛋罐头

低脂鸡肉肠 篮球坎肩 零食 平衡车

用户行为预测综述

有了这些事件以后，就可以把用户行为串联起来，用户进入网站后就是一个新用户，他可能要注册，那么注册行为就是一个事件。注册要填写个人信息，之后他可能开始搜索买东西，所有这些都是用户行为的事件。

网站上有很多需要加载代码的地方，比如注册按钮上面要加、下订单的地方要加，这样我们才能知道用户是否点击了注册按钮、用户下了什么订单。

所有这些通过写代码来详细描述事件和属性的方式，国内都统称为“埋点”，来进行数据采集。

用户行为预测综述

为什么要做用户行为分析？

用户行为分析又有什么用？

用户行为预测综述

构建画像

将用户分类针对性处理

用户行为预测综述

我们的软件平台希望能吸引更多地用户，并尽可能地留住他们减少流失。

用户行为分析就可以帮助我们来了解到用户是怎么流失、为什么会流失、在哪里流失的。

用户行为预测综述

我们可以从一个简单的搜索行为来进行分析用户。

而用户的这一系列的行为被我们捕捉后。我们就可以构建用户画像，来我们帮助为其制定个性化方案，提升用户满意度并留住客户，减少流失。

用户行为预测综述

如何进行用户行为分析？

用户行为预测综述

当你有了很多用户行为分析、定义事件之后，你可以把用户数据做成一个按时间，或用户级别、事件级别的一个表。

这个表帮助我们通过一些极简单的用户事件进行分析。

比如登录或者是购买，就可以知道哪些是优质用户，哪些是即将流失的客户。

用户行为预测综述

时间	新增用户	1天后	2天后	3天后	4天后	5天后	6天后	7天后	14天后	30天后
2020-03-31	56	21.43%	7.14%	5.36%	7.14%	5.36%	7.14%			
2020-04-01	64	31.25%	20.31%	12.5%	14.06%	12.5%				
2020-04-02	58	18.97%	13.79%	12.07%	6.9%					
2020-04-03	60	30%	21.67%	16.67%						
2020-04-04	40	30%	22.5%							
2020-04-05	33	21.21%								

用户行为预测综述

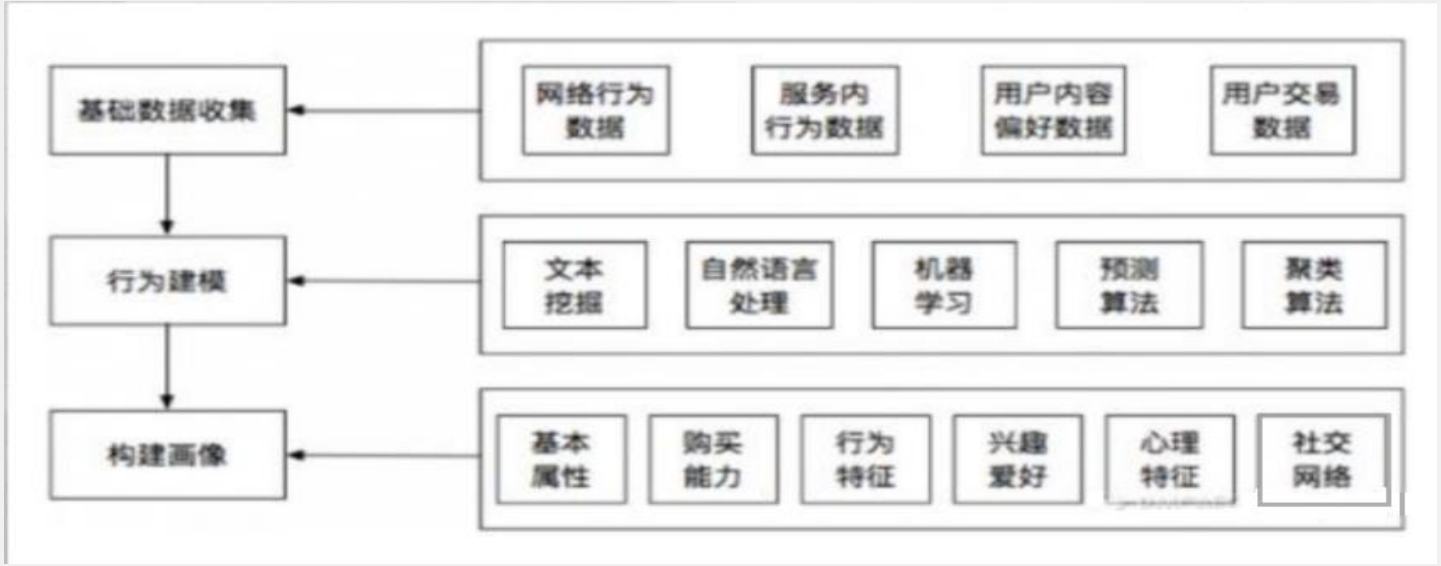
user_visit_action
- date:日期，代表用户点击行为是哪一天发生的。
- user_id:代表点击用户。
- session_id:标识用户session。
- page_id:点击某些商品/品类，或者搜索了某个关键词，然后进入某个页面，页面id。
- action_time:点击行为的发生时间点
- search_keyword:搜索关键词
- click_category_id:点击品类id
- click_product_id:点击商品id
- order_product_ids:订单中包含商品
- order_category_ids:订单中包含品类id。
- pay_category_ids:某次支付对应品类id。
- pay_product_ids:某次支付，对应商品id

user_info
- user_id:每一个用户的唯一标识，通常是自增长Long类型，BigInt类型。
- username:登录名
- name：昵称或者真实名称
- age：年龄
- professional：用户的职业
- city：用户所在城市

那我们现在将采集所有用户行为数据。

进行预处理（数据清理和整合）再根据我们所需把这个用户行为数据展现出来，比如用户的搜索过程。他的搜索内容、点击了第几个搜索结果、到底有没有加入购物车、加入购物车之后有没有下订单等等的这些全量的采集数据。

用户行为预测综述



将这些数据进行分析，我们就可以了解到一些用户的行为习惯。

针对不同的情况，我们会用相应的算法模型来进行分析预测。

2.1

用户行为分析预测模型

——数据采集与数据预处理

Data acquisition and data preprocessing

汇报人：王昱

数据采集

数据采集是掌握足够的信息是做出正确决策的基础

01.埋点

02.无埋点

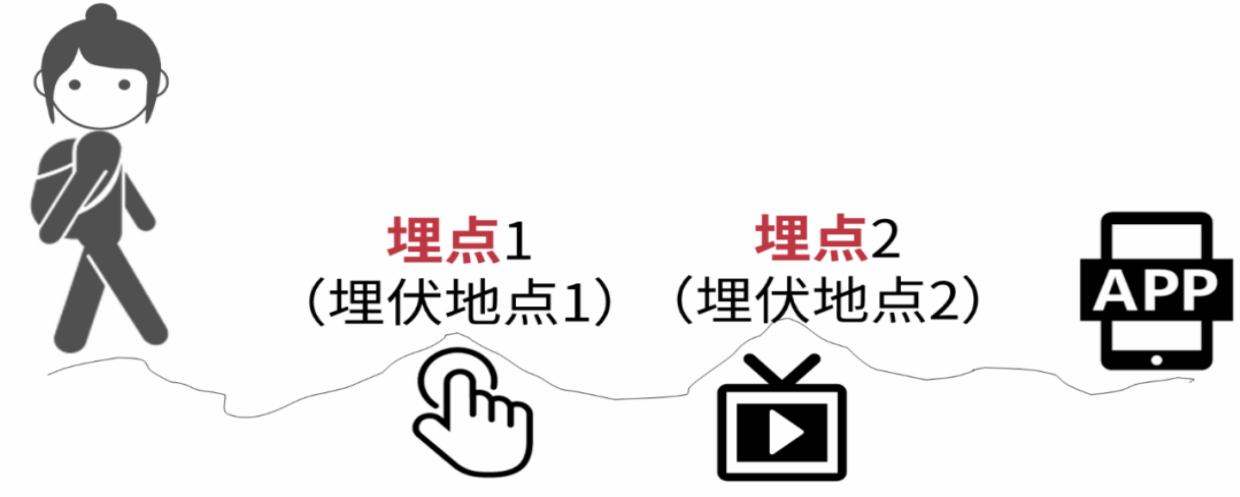
03.可视化埋点

数据采集

信息采集的重要性：信息化时代及时掌握足够的信息是做出正确预测的基础。要选择正确的方向，就必须有足够的信息来作为基础，所以我们便需要收集到大量的信息，并从众多信息中选择出有价值的信息，从而根据这些信息做出判断。

数据采集

一、埋点



1、埋点又称为事件追踪，指的是针对用户行为或流程事件进行捕获，对想要目标数据进行收集的这一过程。

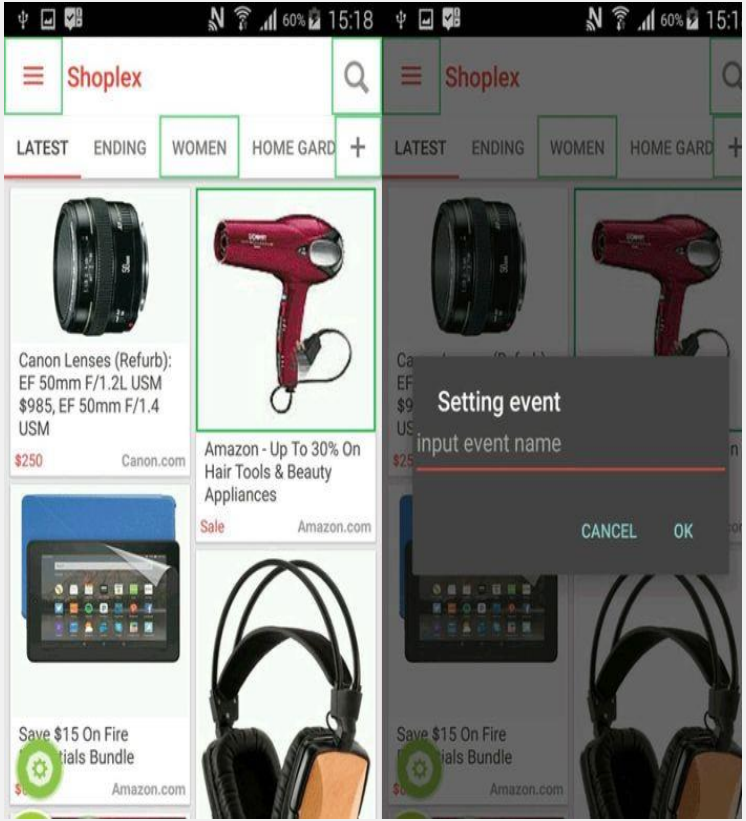
2、巨大且繁琐的埋点工作带来更加全面、精确的用户数据。

数据采集

二、无埋点

1、更加智能的信息监测方式，减少了传统埋点所带来巨大且繁琐的工作量。

2、在监测工具开启无埋点部署的选项后，点击所需预测的点，即可对该位置的用户行为信息进行采集



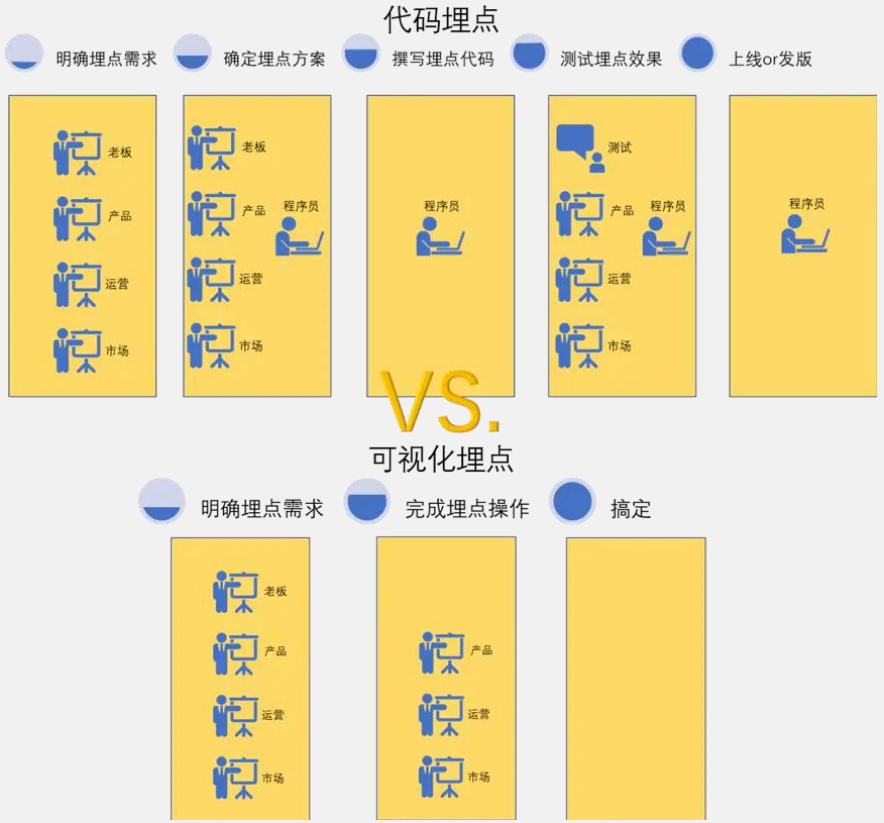
（ptengine的无埋点监测示例图）

数据采集

三、可视化埋点

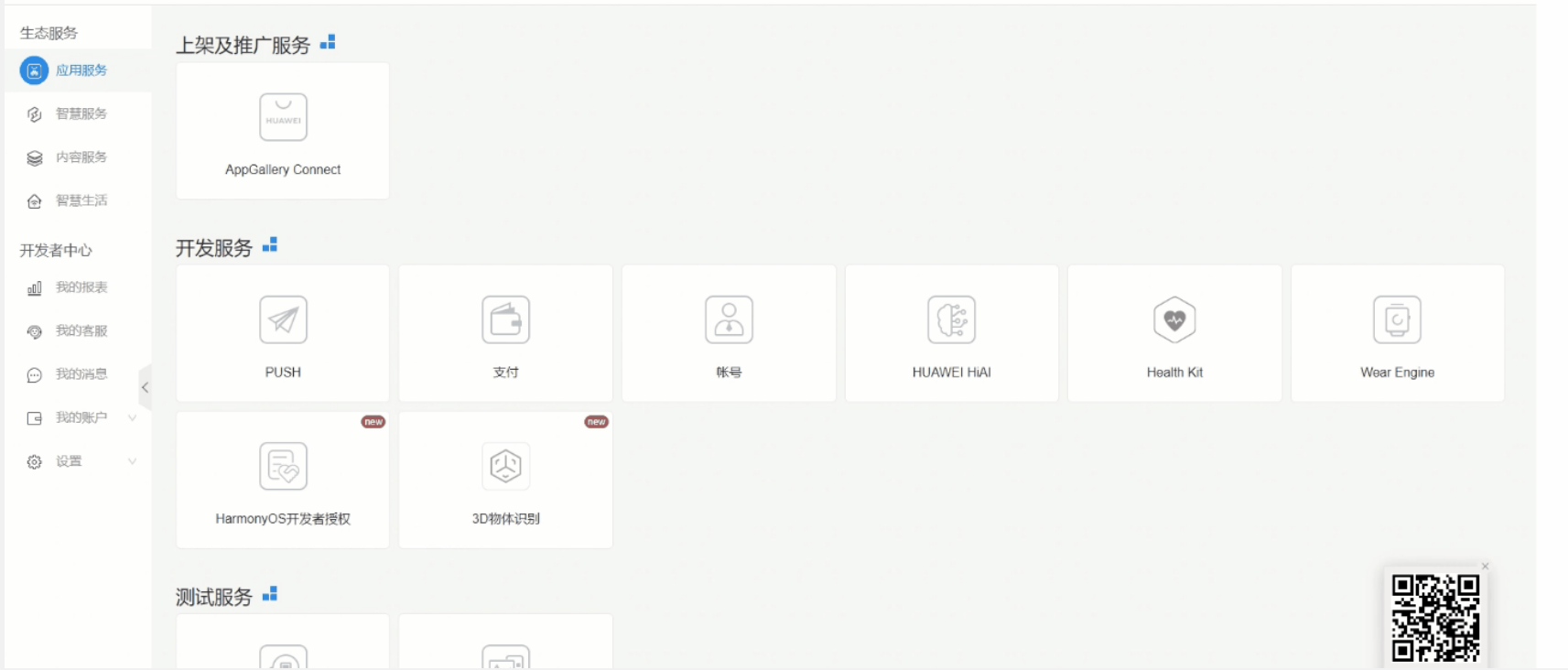
1、可视化埋点的出现，是为更好的解决代码埋点流程复杂、成本高等问题。

2、这种方式所见即所得，跳过代码部署、测试验证和发版过程，使得更加简便。并且可视化埋点在没有对于埋点编程并不熟悉的的情况下都可以非常低的门槛获取到用于分析的数据。



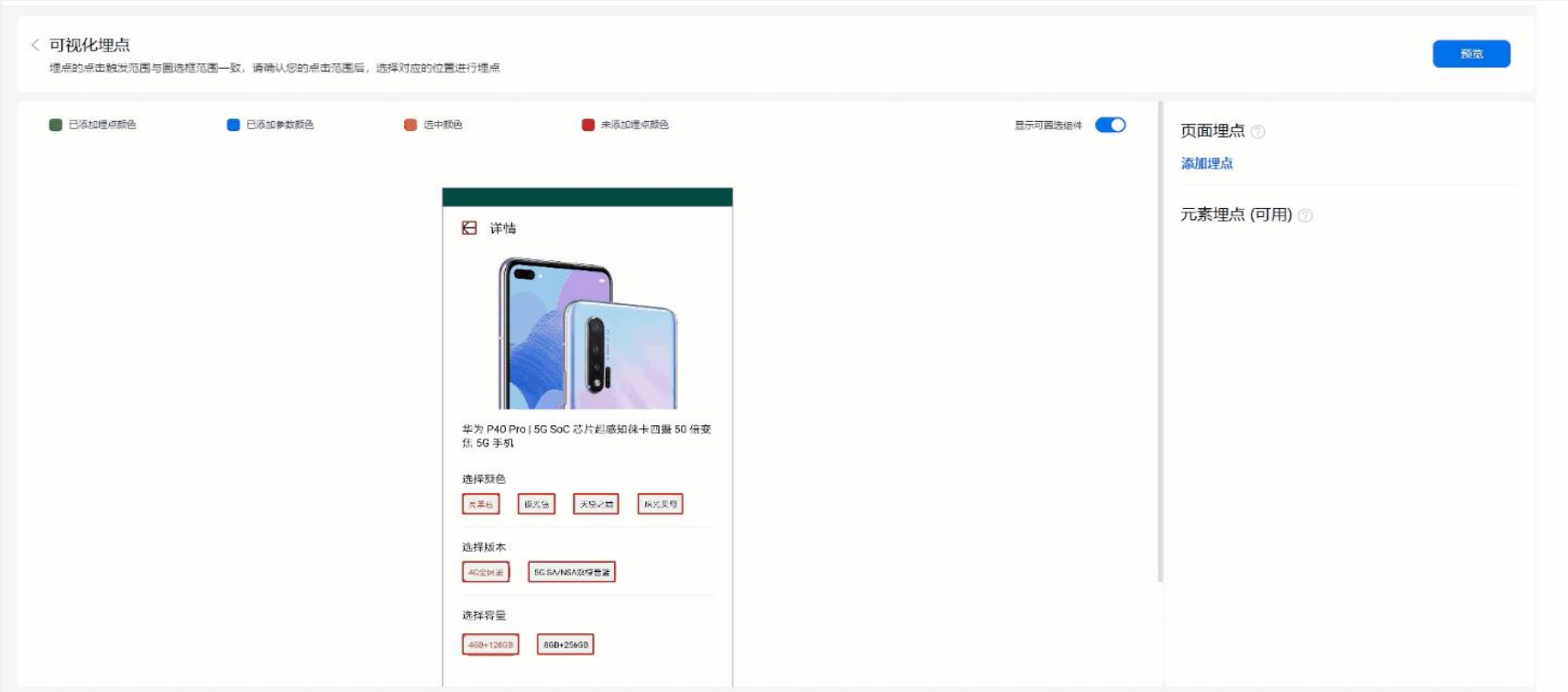
数据采集

基于Huawei Analytics的可视化埋点演示（Android）



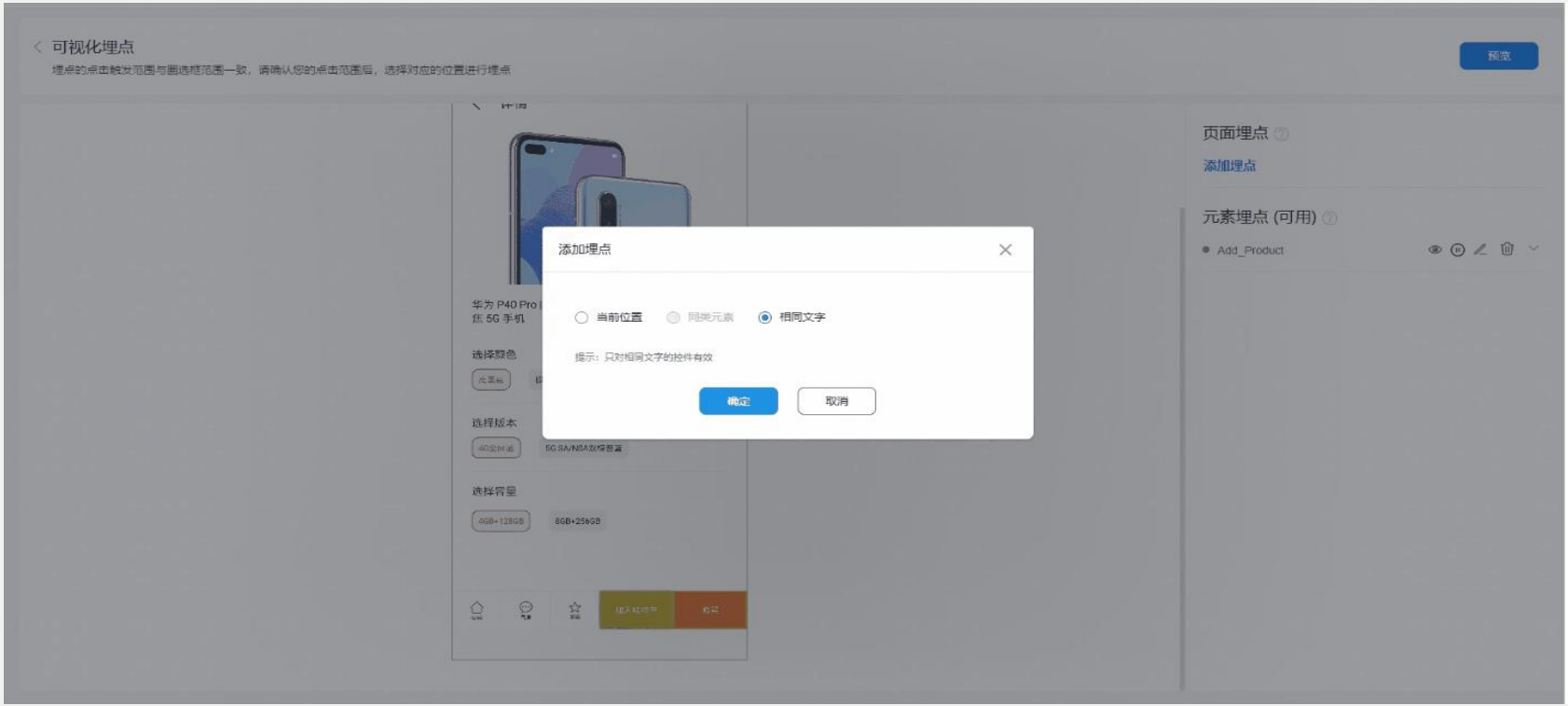
数据采集

根据当前位置进行可视化埋点演示



数据采集

根据文字所进行可视化埋点演示



数据预处理

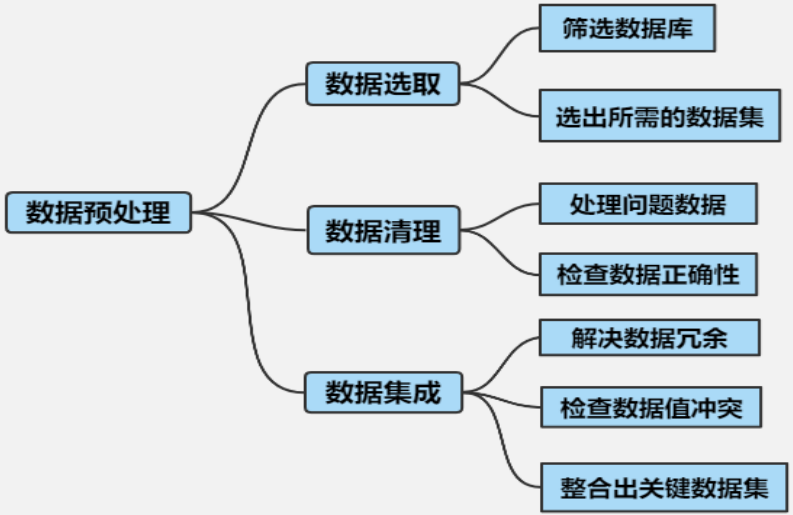
高质量的原始数据来为后续的工作打下坚实的基础

01.数据选取

02.数据清理

03.数据集成

数据预处理



数据预处理的意义：数据的质量是影响数据分析结果的准确性和有效性的重要因素，在用户行为的分析中也是如此，且异常的数据会对预测的结果造成很大的影响，从而对用户行为的预测产生负面影响，数据预处理技术起到关键作用。

数据预处理

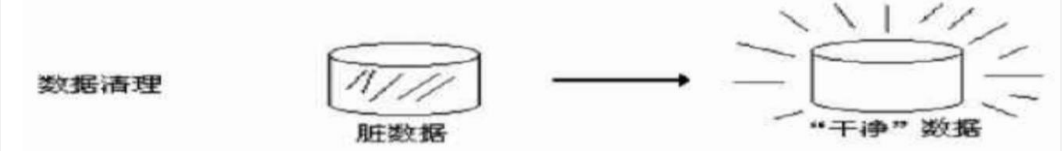
一、数据选取

1、数据选取具体是指根据分析所需在原始数据集中选出所需的用户数据，在数据集内存在大量数据，具有相对广泛的覆盖范围。

2、在部分数据表格内会存在着完全没有联系的数据。如果不能简单筛选数据库，则会使其挖掘过程存在大量无用数据，进而造成资源浪费。

数据预处理

二、数据清理



1、在处理缺失数据时，如果在一个元组内有多个属性值并同时出现空缺，可以将其忽略并直接在数据表格中进行删除。而当元组内属性值缺失较少时，则需要填补空缺直。一般可采取全局常量、人工填补等方式进行填补。

2、当对错误数据进行处理时，首先需要对元组进行分辨，决定是忽略元组还是更改数据。

数据预处理

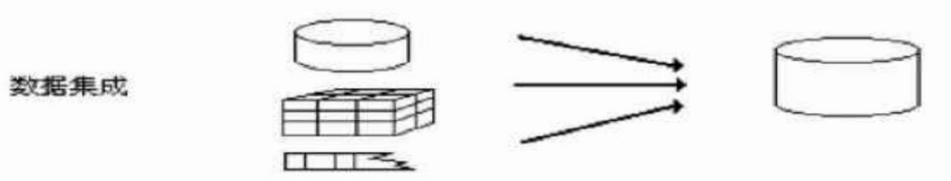
二、数据清理

3.噪音数据：噪声数据是指数据中存在着错误或偏差值较大的数据，当其偏差较大时，则会出现孤立点这些数据对数据的分析造成了干扰。

当出现噪音数据时，可采用分箱方法来对噪音数据进行处理，也就是通过考察相邻数据来确定最终值。

数据预处理

三、数据集成



1、数据集成的本质是数据整合，数据集成通过不同的数据交换从而达到集成，主要解决数据的分布性和异构性的问题。

2、在进行数据集成时会出现数据冗余的情况，例如多次出现同一属性，而对于这种情况，可以对其相关性进行有效的分析检测，从而删除掉其中的无用数据，得到最终的用户行为数据。

2.2

用户行为分析预测模型

——特征提取与预测

Feature Extraction and Learning to Prediction

汇报人：董思超

特征提取

特征提取是数据降维的一种技术

01. 图像特征提取

02. 文字特征提取

-词库模型（Bag-of-words model）

03. 特征提取方法

-主成分分析（PCA）
-线性判别分析（LDA）

特征提取

特征提取是从原始特征中找出最有效（同类样本的不变性、不同样本的鉴别性、对噪声的鲁棒性）的特征,对数据起到降维的目的。

01. 图像特征提取

几种常用的图像特征

Option 01

颜色特征

是一种全局特征，描述了图像或图像区域所对应的景物的表面性质。

Option 02

纹理特征

纹理特征也是一种全局特征，它也描述了图像或图像区域所对应景物的表面性质。

Option 03

形状特征

各种基于形状特征的检索方法都可以比较有效地利用图像中感兴趣的目标来进行检索。

Option 04

空间关系特征

可加强对图像内容的描述区分能力，但空间关系特征常对图像或目标的旋转、反转、尺度变化等比较敏感。

02. 文字特征提取

词库模型（Bag-of-words model）

给定两句简单的文档：

文档 1：“我喜欢游泳，小明也喜欢。”

文档 2：“我也喜欢跑步。”

基于以上这两个文档，便可以构造一个由文档中的关键词组成的词典：

词典={1:“我”，2:“喜欢”，3:“游泳”，4:“小明”，5:“也”，6:“跑步”}

02. 文字特征提取

词库模型（Bag-of-words model）

这个词典一共包含6个不同的词语，利用词典的索引号，上面两个文档每一个都可以用一个6维向量表示；表示某个单词在文档中出现的次数。
这样，根据各个文档中关键词出现的次数，便可以将上述两个文档分别表示成向量的形式：

文档 1：“我喜欢游泳，小明也喜欢。”　　文档 1：[1, 2, 1, 1, 1, 0]

文档 2：“我也喜欢跑步。”　　文档 2：[1, 1, 0, 0, 1, 1]

词典={1:“我”，2:“喜欢”，3:“游泳”，4:“小明”，5:“也”，6:“跑步”}

03. 特征提取方法——主成分分析（PCA）

最常用的线性降维方法，它的目标是通过某种线性投影，将高维的数据映射到低维的空间中，并期望在所投影的维度上数据的信息量最大（方差最大），以此使用较少的数据维度，同时保留住较多的原数据点的特性。

假设有一组数组：

	x	y
	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
Data =	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

有10篇文档：
x是10篇文档中“learn”出现的TF-IDF
y是10篇文档中“study”出现的TF-IDF。

03. 特征提取方法——主成分分析（PCA）

第一步：分别求x和y的平均值，然后对于所有的样例，都减去对应的均值。
这里x的均值是1.81，y的均值是1.91

	x	y
	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
Data =	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

求解得到：

	x	y
	0.69	0.49
	-1.31	-1.21
	0.39	0.99
	0.09	0.29
DataAdjust =	1.29	1.09
	0.49	0.79
	0.19	-0.31
	-0.81	-0.81
	-0.31	-0.31
	-0.71	-1.01

03. 特征提取方法——主成分分析（PCA）

第二步：求特征协方差矩阵

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(y,x) & cov(y,y) \end{pmatrix}$$

求解得到：$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$

03. 特征提取方法——主成分分析（PCA）

第三步：求协方差的特征值和特征向量

求解得到特征值：

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

求解得到特征向量：

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

03. 特征提取方法——主成分分析（PCA）

第四步：将特征值按照从大到小的顺序排序，选择其中最大的k个，然后将其对应的k个特征向量分别作为列向量组成特征向量矩阵。

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

例子特征值只有两个，选择其中最大的，特征值是1.28402771，对应的特征向量是(-0.677873399, -0.735178656)T

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

03. 特征提取方法——主成分分析（PCA）

第五步：将样本点投影到选取的特征向量上。

样例数为m，特征数为n，

减去均值后的样本矩阵为DataAdjust(m*n)，协方差矩阵是n*n，

选取的k个特征向量组成的矩阵为EigenVectors(n*k)。

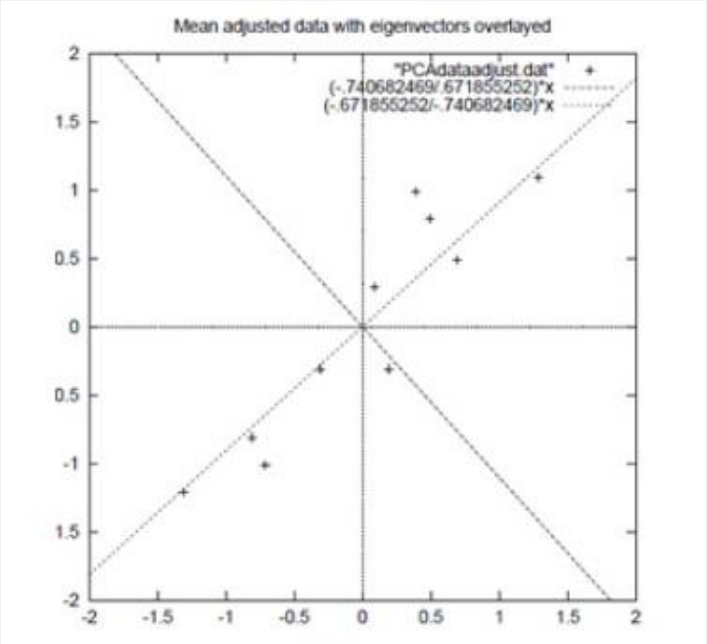
Transformed Data (Single rigenvector)
x
-0.827970186
1.77758033
-0.992197494
-0.274210416
-1.67580142
-0.912949103
0.991094375
1.14457216
0.438046137
1.22382056

那么投影后的数据FinalData为：

FinalData(10*1) = DataAdjust(10*2矩阵) *特征向量(-0.677873399, -0.735178656)T

03. 特征提取方法——主成分分析（PCA）

这样，就将原始样例的n维特征变成了k维，这k维就是原始特征在k维上的投影。

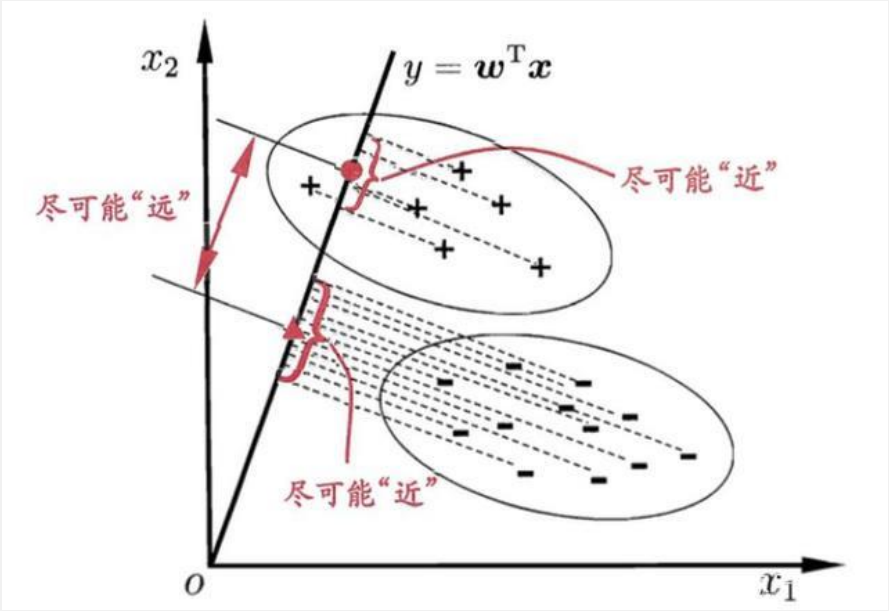


03. 特征提取方法——线性判别分析（LDA）

线性判别分析（LDA）

属于监督学习的降维算法。
与PCA这种无监督的降维算法不同，LDA要求输入数据有对应的标签。

LDA降维的基本思想是映射到低维之后，最大化类间均值，最小化类内方差



03. 特征提取方法——线性判别分析（LDA）

LDA与PCA的对比

相同点：

1.两者均可以对数据进行降维。
2.两者在降维时均使用了矩阵特征分解的思想。
3.两者都假设数据符合高斯分布。

不同点：

1.LDA是有监督的降维方法，而PCA是无监督的降维方法
2.LDA降维最多降到类别数k-1的维数，而PCA没有这个限制。
3.LDA除了可以用于降维，还可以用于分类。
4.LDA选择分类性能最好的投影方向，而PCA选择样本点投影具有最大方差的方向。

预测

01. 预测案例

-2016年人工智能预测美国大选案例

02. “个性化”？ 智能预测的问题

-“个性化”推送干预选民意见

01.预测案例

——2016年人工智能预测美国大选案例

2016年美国大选

01.预测案例

——2016年人工智能预测美国大选案例

2016年美国大选——人工智能用于政治选举的新时代

系统名称	分析来源	分析手段
Bing Predicts	Bing、社交媒体	机器学习
Unanimous AI	人为分析对象	集体预测
UNO	直接询问选民	“集群智能”
MogIA	谷歌、YouTube、twitter 的数据点	大数据分析

2016 大选中出现的人工智能系统

如今选民倾向于使用互联网和社交媒体来获取信息，所以人工智能应用于大选的预测已经开始凸显其优势。

01.预测案例

——2016年人工智能预测美国大选案例

01.
分类和排序

年龄、性别、身高、学历

02.
过滤

将与目标本身最直接相关的数据作为放置于优先地位

03.
建模

使用建模了解选民之间投票分析特征差异等

04.
评估

评估的结果足够精确，就可以对未来总统大选走向进行预测

01.预测案例

——2016年人工智能预测美国大选案例

系统名称	分析来源	分析手段	预测结果
Bing Predicts	Bing、社交媒体	机器学习	希拉里胜出
Unanimous AI	人为分析对象	集体预测	希拉里胜出
UNO	直接询问选民	“集群智能”	希拉里胜出
MogIA	谷歌、YouTube、twitter 的数据点	大数据分析	特朗普胜出

2016 大选中出现的人工智能系统

02. “个性化”？ 智能预测的问题

——“个性化”推送干预选民意见

“过滤气泡”理论

基于大数据与算法推荐为底层机构，根据用户使用习惯等生成用户画像，通过针对个人化搜索而提供筛选后结果的算法为其提供数据推荐。

“回声室”效应

在网络空间内，人们经常接触相对同质化的人群和信息，听到相似的评论，倾向于将其当作真相和真理，不知不觉中窄化自己的眼界和理解，走向故步自封甚至偏执极化

02. “个性化”？ 智能预测的问题

——“个性化”推送干预选民意见

剑桥分析丑闻

剑桥分析丑闻，指8700万Facebook用户数据被不当泄露给政治咨询公司剑桥分析，用于在2016年总统大选时支持美国总统特朗普。

02. “个性化”？ 智能预测的问题

——“个性化”推送干预选民意见

心理分析

早在 2016 年之前，剑桥分析公司的员工就曾通过问卷对用户进行性格测试，通过有偿的的形式吸引用户填写问卷并收集用户的个人数据，例如购物数据、汽车数据、Facebook 个人账户信息等，特别关注的是选民的个人偏好和态度。然后利用心理学普遍应用的“大五”分析法：对选民进行分类整合，接着将数据发送到剑桥分析公司研发的模型中加以分析。

表 1　五维度人格模型

分析维度	描述内容	分类
外倾性	个人对关系的舒适程度	外倾者倾向于喜欢群居、善于社交和自我决断；内倾者倾向于封闭内向、胆小害羞和安静少语
随和性	个人服从别人的倾向性	高随和性的人是合作的、热情的、信赖他人的；低随和性的人是冷淡的、敌对的、不受欢迎的
责任心	测量个人的信誉	高责任心的人是负责任的、有条不紊的、值得信赖的、持之以恒的；低责任心的人很容易精力分散、缺乏规划性且不可信赖
情绪稳定性	描述个体承受压力的能力，经常使用对立面神经质进行解释	积极情绪稳定者倾向于平和的、自信的、安全的；消极情绪稳定者倾向于紧张的、焦虑的、失望的、缺乏安全感的
经验开放性	个体对新奇事物的兴趣和热衷程度	开放性非常高的人富有创造性、凡事好奇、具有艺术的敏感性；开放性非常低的人很保守，对熟悉的事物感到舒服和满足

02. “个性化”？ 智能预测的问题

——“个性化”推送干预选民意见

根据用户性格画像精准推送信息

根据上一步搜集到的信息建立“用户性格画像”，根据每个人的性格特点、关注主题、认知等不同标准，把美国民众分为了 32 种不同的性格类型，给他们传播不同的政治理念。这就使得普通选民的选择被放置于这种特定的广告推送中，选民在社交网站看到的信息都是经过“操纵”的，选民最终的政治选择也会受到影响。

03

前沿进展

Cutting-edge progressing

汇报人：刘琦玥

谈一谈

大数据时代下的伦理道德

大数据时代下的伦理道德

“推动互联网、大数据、人工智能和实体经济深度融合”

双刃剑

大数据时代下的伦理道德

中华人民共和国国家互联网信息办公室

Cyberspace Administration of China

WWW.CAC.GOV.CN

请输入检索关键词

首页 权威发布 办公室工作 网络安全 信息化 网络传播 国际交流 地方网信 执法督查 政策法规 互动中心 教育培训 业界动态 工作专题

当前位置：首页 > 正文

互联网信息服务算法推荐管理规定

2022年01月04日 10:02 来源：中国网信网 【打印】【纠错】

国家互联网信息办公室

中华人民共和国工业和信息化部

中华人民共和国公安部

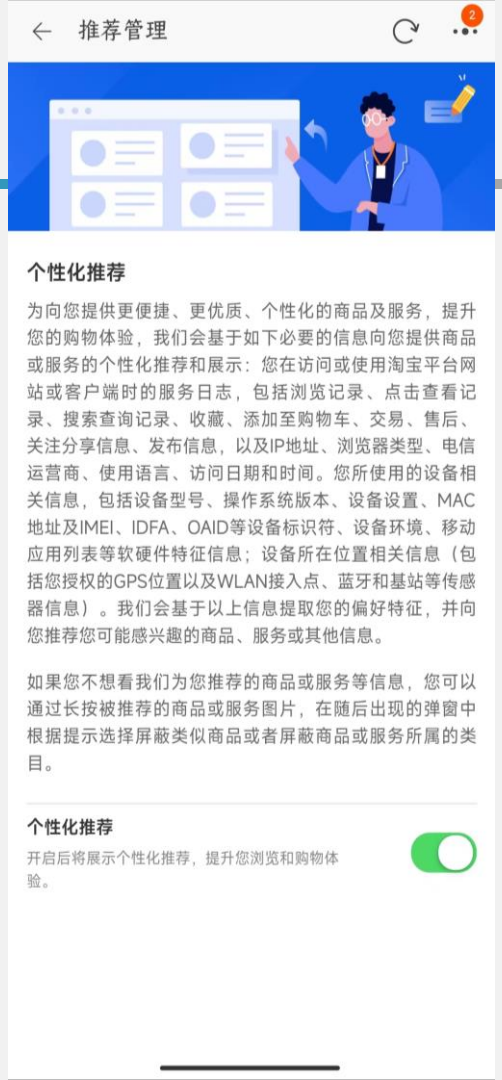
国家市场监督管理总局

令

第9号

《互联网信息服务算法推荐管理规定》已经2021年11月16日国家互联网信息办公室2021年第20次室务会议审议通过，并经工业和信息化


隐私
个人主页
谁可以看我发布的内容
所有人
谁可以看我点赞的内容
所有人
通过个人主页私信我
关闭
允许查看我的关注/粉丝列表
权限管理
系统权限管理
管理APP使用的系统权限
授权管理
管理已授权的应用权限
广告管理
管理向您展示的个性化广告类型
推荐管理
管理个性化推荐内容的展示
隐私说明
隐私政策摘要
未成年人保护
个人信息下载
我有疑问


推荐管理
个性化推荐
为向您提供更便捷、更优质、个性化的商品及服务，提升您的购物体验，我们会基于如下必要的信息向您提供商品或服务的个性化推荐和展示：您在访问或使用淘宝平台网站或客户端时的服务日志，包括浏览记录、点击查看记录、搜索查询记录、收藏、添加至购物车、交易、售后、关注分享信息、发布信息，以及IP地址、浏览器类型、电信运营商、使用语言、访问日期和时间。您所使用的设备相关信息，包括设备型号、操作系统版本、设备设置、MAC地址及IMEI、IDFA、OAID等设备标识符、设备环境、移动应用列表等软硬件特征信息；设备所在位置相关信息（包括您授权的GPS位置以及WLAN接入点、蓝牙和基站等传感器信息）。我们会基于以上信息提取您的偏好特征，并向您推荐您可能感兴趣的商品、服务或其他信息。
如果您不想看我们为您推荐的商品或服务等信息，您可以通过长按被推荐的商品或服务图片，在随后出现的弹窗中根据提示选择屏蔽类似商品或者屏蔽商品或服务所属的类目。
个性化推荐
开启后将展示个性化推荐，提升您浏览和购物体验。

关闭个性化推荐

大数据时代下的伦理道德

技术走到尽头了？

责任意识

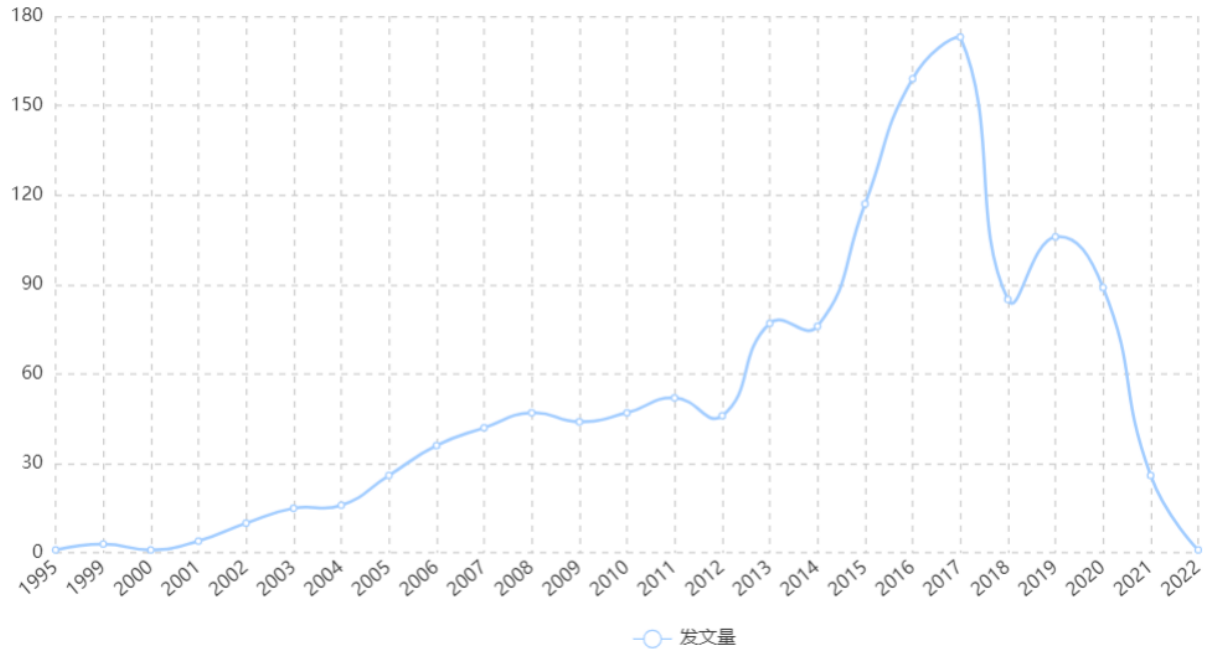
科技前沿进展

前沿进展

来自万方数据库

年份 关键词 作者 机构 学科 期刊 基金 资源类型

年份	发文量	占比
2022	1	0.08%
2021	26	2.00%
2020	89	6.85%
2019	106	8.15%
2018	85	6.54%
2017	173	13.31%
2016	159	12.23%
2015	117	9.00%
2014	76	5.85%
2013	77	5.92%

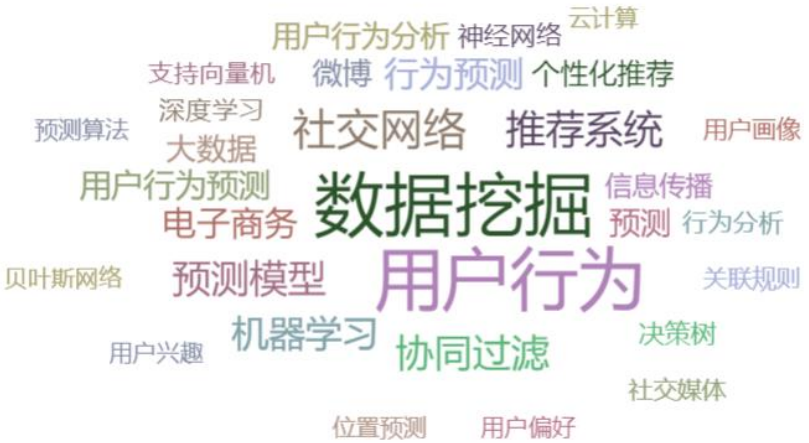


前沿进展

来自万方数据库

年份 | 关键词 | 作者 | 机构 | 学科 | 期刊 | 基金 | 资源类型

关键词	频次	百分比
数据挖掘	144	11.08%
用户行为	138	10.62%
社交网络	70	5.38%
推荐系统	58	4.46%
协同过滤	56	4.31%
预测模型	56	4.31%
机器学习	51	3.92%
行为预测	46	3.54%
电子商务	43	3.31%
用户行为预测	37	2.85%



前沿进展

近年来的研究热点：

01

数据个性化差异

Data personalization differences

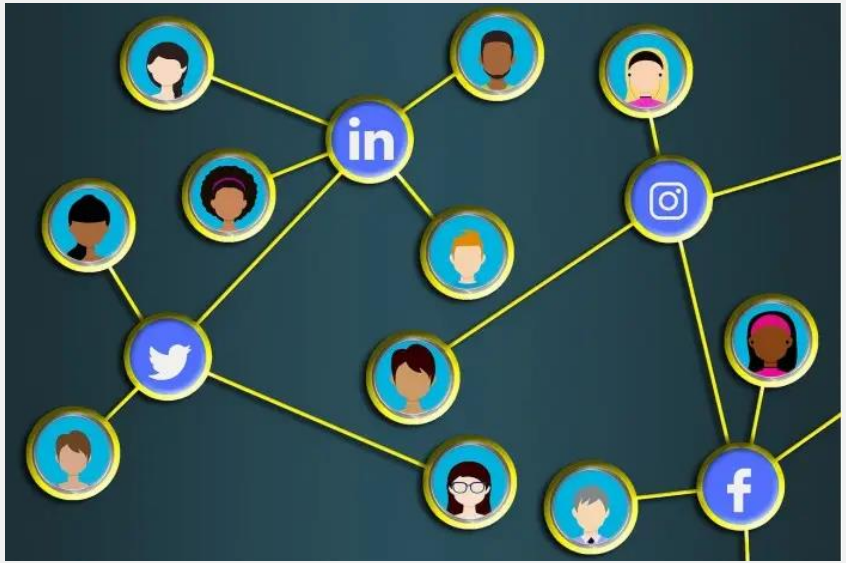
02

数据稀疏性问题

Data sparsity problem

前沿进展

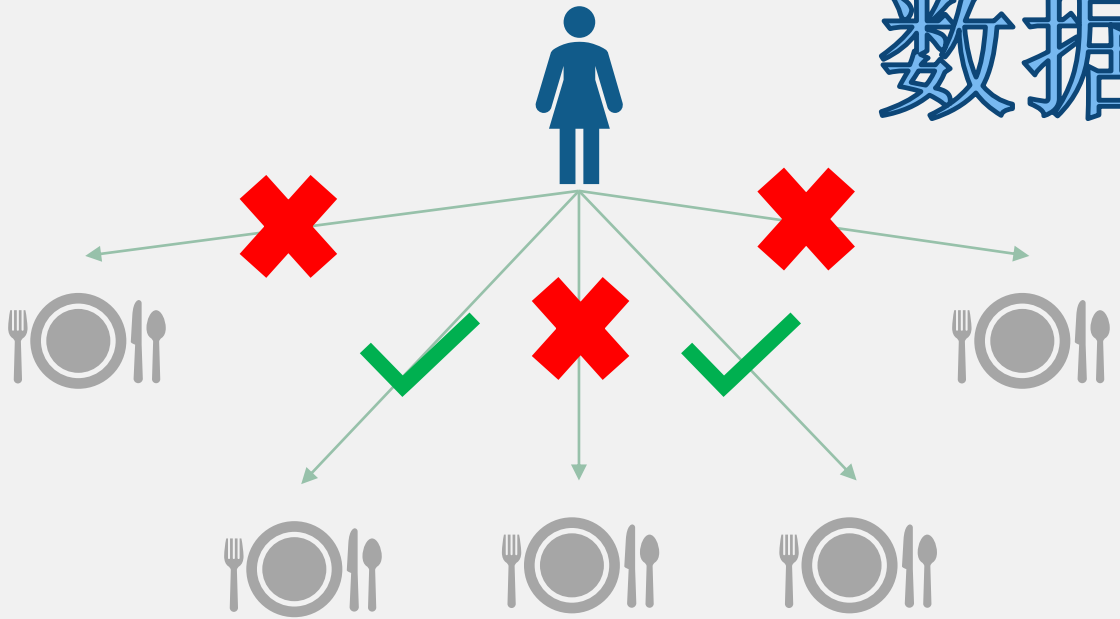
随着人们在互联网上存留信息的多样化，如用户行为历史和评论信息等，许多工作开始利用额外的用户相关信息来提高模型的预测性能，特别是当用户数据包含了用户的社交关系时。



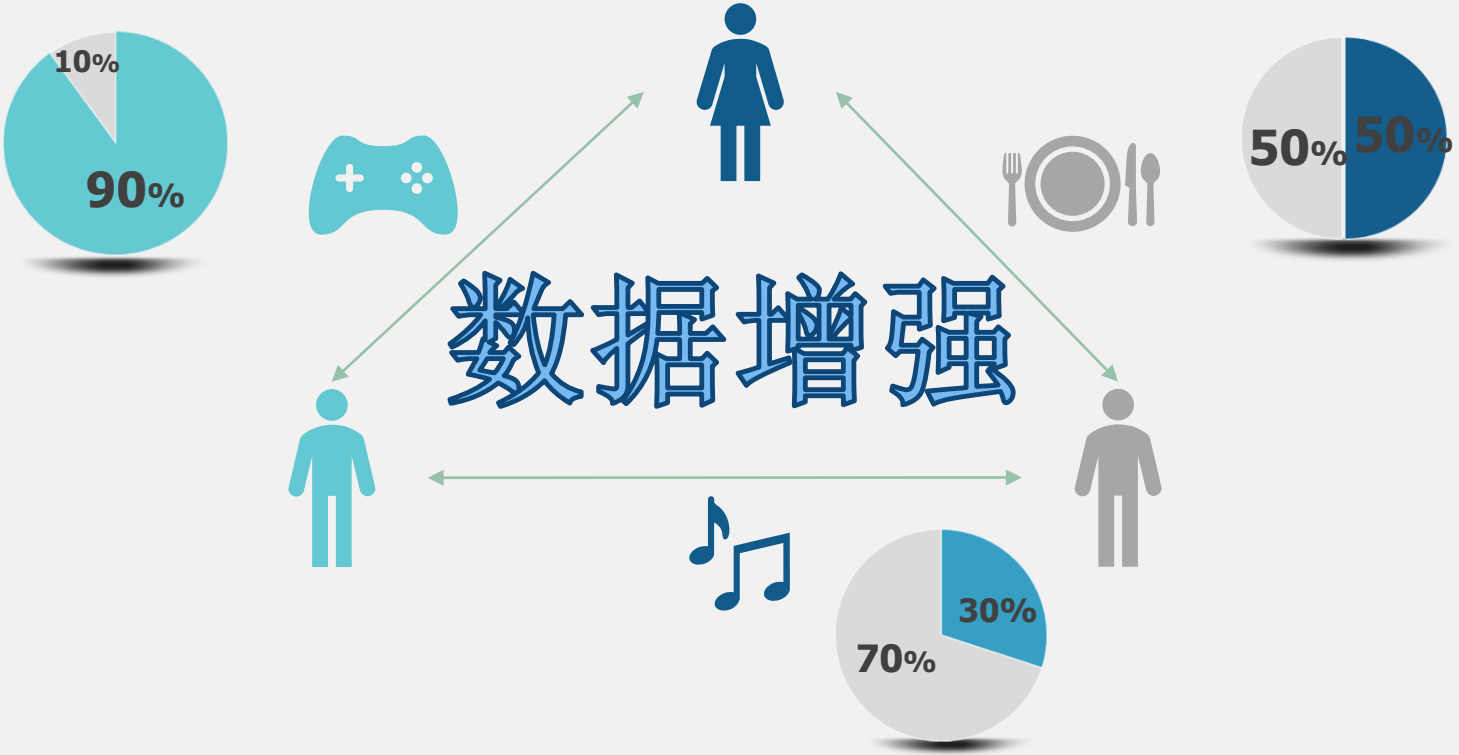
前沿进展

为了缓解**数据稀疏**问题，研究者们提出了一种融合信息网络结构的数据增强行为预测算法。针对用户当前的某一个行为，我们通过用户好友的行为信息网络来度量用户接收该好友信息的程度，并将好友的行为数据嵌入到用户数据中，从而实现对用户进行**数据增强**。

前沿进展



前沿进展



04

Demo演示

Demo

汇报人：沈雨婷

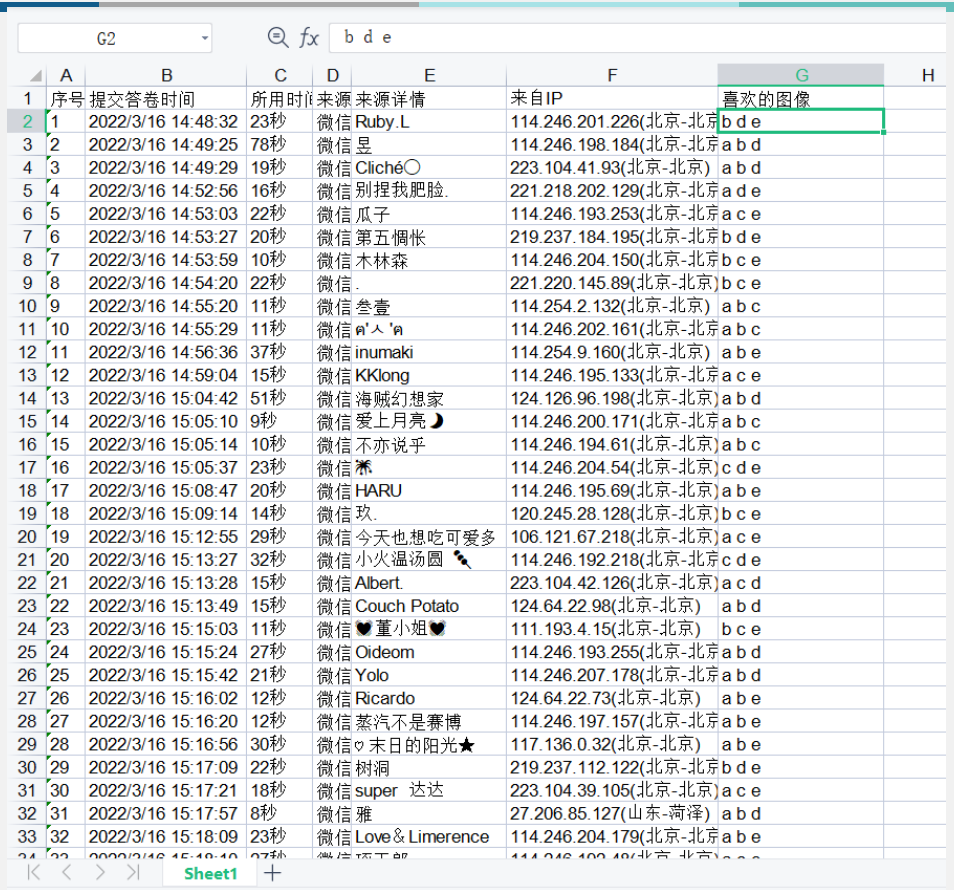
Demo

基于用户行为分析的推荐算法是个性化推荐系统的重要算法，也被称为协同过滤算法，即通过用户与网站不断互动，来不断过滤自己感兴趣的物品。

参考：https://blog.csdn.net/BGMcat/article/details/123300633

Demo

step 1 获取数据



	A	B	C	D	E	F	G	H
1	序号	提交答卷时间	所用时间	来源	来源详情	来自IP	喜欢的图像	
2	1	2022/3/16 14:48:32	23秒	微信	Ruby.L	114.246.201.226(北京-北京	b d e	
3	2	2022/3/16 14:49:25	78秒	微信	昱	114.246.198.184(北京-北京	a b d	
4	3	2022/3/16 14:49:29	19秒	微信	Cliché〇	223.104.41.93(北京-北京)	a b d	
5	4	2022/3/16 14:52:56	16秒	微信	别捏我肥脸.	221.218.202.129(北京-北京	a d e	
6	5	2022/3/16 14:53:03	22秒	微信	瓜子	114.246.193.253(北京-北京	a c e	
7	6	2022/3/16 14:53:27	20秒	微信	第五惆怅	219.237.184.195(北京-北京	b d e	
8	7	2022/3/16 14:53:59	10秒	微信	木林森	114.246.204.150(北京-北京	b c e	
9	8	2022/3/16 14:54:20	22秒	微信	.	221.220.145.89(北京-北京)	b c e	
10	9	2022/3/16 14:55:20	11秒	微信	叁壹	114.254.2.132(北京-北京)	a b c	
11	10	2022/3/16 14:55:29	11秒	微信	ค'ㅅ'ค	114.246.202.161(北京-北京	a b c	
12	11	2022/3/16 14:56:36	37秒	微信	inumaki	114.254.9.160(北京-北京)	a b e	
13	12	2022/3/16 14:59:04	15秒	微信	KKlong	114.246.195.133(北京-北京	a c e	
14	13	2022/3/16 15:04:42	51秒	微信	海贼幻想家	124.126.96.198(北京-北京)	a b d	
15	14	2022/3/16 15:05:10	9秒	微信	爱上月亮🌛	114.246.200.171(北京-北京	a b c	
16	15	2022/3/16 15:05:14	10秒	微信	不亦说乎	114.246.194.61(北京-北京)	a b c	
17	16	2022/3/16 15:05:37	23秒	微信	🎋	114.246.204.54(北京-北京)	c d e	
18	17	2022/3/16 15:08:47	20秒	微信	HARU	114.246.195.69(北京-北京)	a b e	
19	18	2022/3/16 15:09:14	14秒	微信	玖.	120.245.28.128(北京-北京)	b c e	
20	19	2022/3/16 15:12:55	29秒	微信	今天也想吃可爱多	106.121.67.218(北京-北京)	a c e	
21	20	2022/3/16 15:13:27	32秒	微信	小火温汤圆 🍡	114.246.192.218(北京-北京	c d e	
22	21	2022/3/16 15:13:28	15秒	微信	Albert.	223.104.42.126(北京-北京)	a c d	
23	22	2022/3/16 15:13:49	15秒	微信	Couch Potato	124.64.22.98(北京-北京)	a b d	
24	23	2022/3/16 15:15:03	11秒	微信	🖤董小姐🖤	111.193.4.15(北京-北京)	b c e	
25	24	2022/3/16 15:15:24	27秒	微信	Oideom	114.246.193.255(北京-北京	a b d	
26	25	2022/3/16 15:15:42	21秒	微信	Yolo	114.246.207.178(北京-北京	a b d	
27	26	2022/3/16 15:16:02	12秒	微信	Ricardo	124.64.22.73(北京-北京)	a b e	
28	27	2022/3/16 15:16:20	12秒	微信	蒸汽不是赛博	114.246.197.157(北京-北京	a b e	
29	28	2022/3/16 15:16:56	30秒	微信	♡末日的阳光★	117.136.0.32(北京-北京)	a b e	
30	29	2022/3/16 15:17:09	22秒	微信	树洞	219.237.112.122(北京-北京	b d e	
31	30	2022/3/16 15:17:21	18秒	微信	super 达达	223.104.39.105(北京-北京)	a c e	
32	31	2022/3/16 15:17:57	8秒	微信	雅	27.206.85.127(山东-菏泽)	a b d	
33	32	2022/3/16 15:18:09	23秒	微信	Love & Limerence	114.246.204.179(北京-北京	a b e	

Demo

step 2 处理数据

main.py × pre.py ×

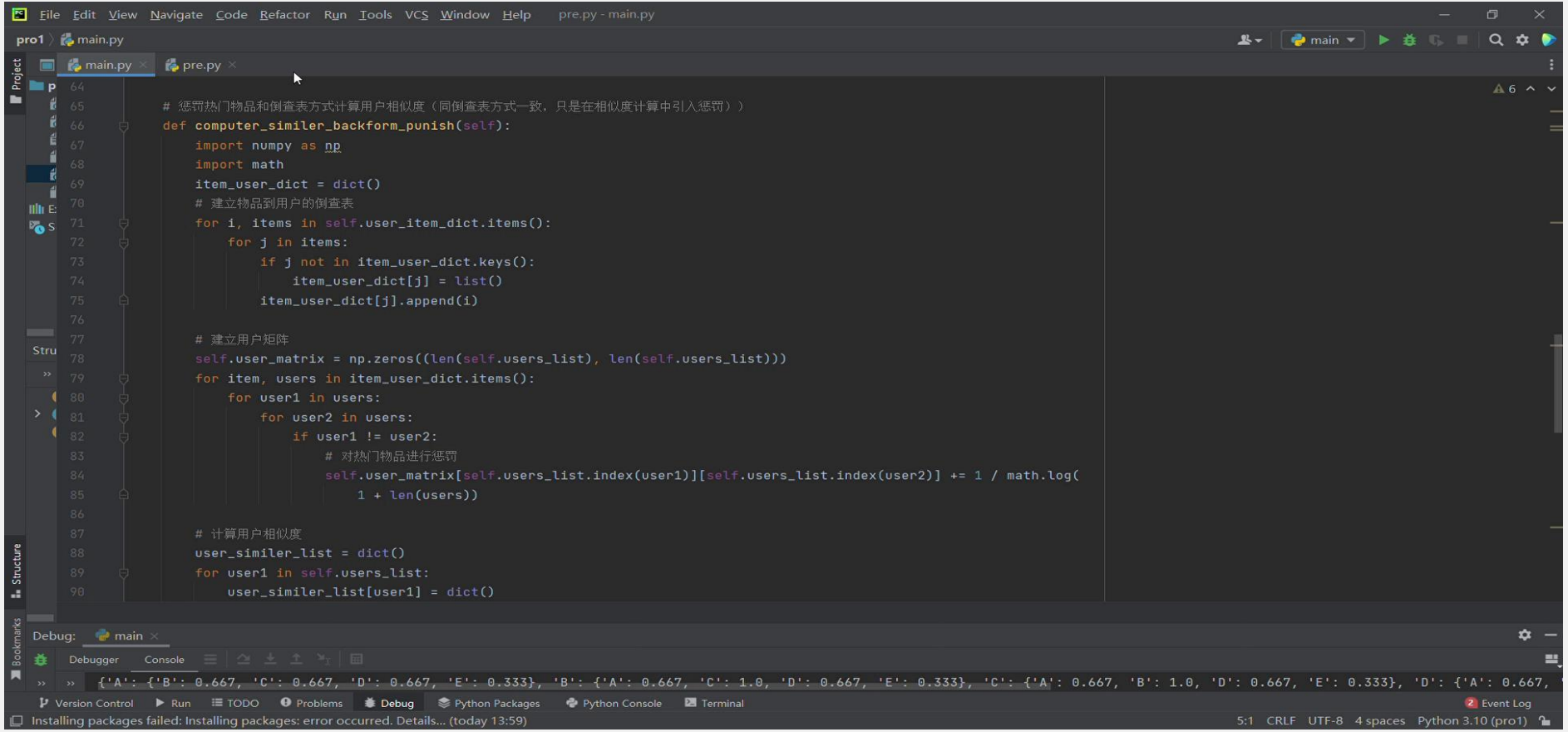
```python
from collections import defaultdict


result = []
def xlsx_to_dict():
    dic={}
    df = pd.read_excel('D:/pro1/ori.xlsx', nrows = 20)
    df = df.drop(['提交答卷时间', '所用时间', '来源', '来源详情', '来自IP'], axis=1, inplace=False)
    df["序号"] = ['id%i' % i for i in + df['序号']]
    df['like1'] = df['喜欢的图像'].map(lambda x: x.split(' ')[0])
    df['like2'] = df['喜欢的图像'].map(lambda x: x.split(' ')[1])
    df['like3'] = df['喜欢的图像'].map(lambda x: x.split(' ')[2])
    df['喜欢的图像'] = df['喜欢的图像'].str.replace(' ',',')
    df_a = df.loc[:, ['序号']]
    df_a = df_a.values.tolist()
    for s_a in df_a:
        result.append(s_a[0])
    dsn = df.loc[:,['like1','like2','like3']]
    dsn = dsn.values.tolist()
    dic = defaultdict(list)
    try:
        for key, value in zip(result, dsn):
            dic[key].append(value)
        return dic
    finally:
        print(dic)
```

pre ×

```
C:\ProgramData\Anaconda3\envs\pro1\python.exe D:/pro1/pre.py
defaultdict(<class 'list'>, {'id1': [['b', 'd', 'e']], 'id2': [['a', 'b', 'd']], 'id3': [['a', 'b', 'd']], 'id4': [['a', 'd',
```

Demo

step 3 基于用户的协同过滤算法



Demo

step 3 基于用户的协同过滤算法

```
import numpy as np


user_item_dict = {'A': ['b', 'd', 'e'], 'B': ['a', 'b', 'd'], 'C': ['a', 'b', 'd'], 'D': ['a', 'd', 'e'], 'E': ['a', 'c', 'e']}
class UserCF:
    def __init__(self, user_item_dict):...

    # 两两用户之间计算用户相似度
    def compute_similer_twouser(self):...

    #      优化后的倒查表方式计算用户相似度
    def computer_similer_backform_normal(self):...

    # 惩罚热门物品和倒查表方式计算用户相似度（同倒查表方式一致，只是在相似度计算中引入惩罚））
    def computer_similer_backform_punish(self):...

    # 计算指定用户的推荐物品
    def compute_interest_item(self, user):...
```

Demo

step 3 基于用户的协同过滤算法——用户相似度

1. 计算用户相似度

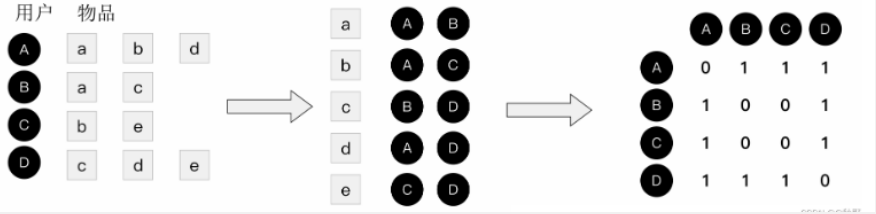
$$w_{uv} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

Jaccard公式

$$w_{uv} = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u)||N(v)|}}$$

余弦相似度

其中分子部分，即交集部分，如果两两计算时间复杂度较大，可以先建立用户倒查表，来存储对该物品产生过行为的用户列表。同时在改良方法中为降低热门物品的影响，引入惩罚项。




```python
# 两两用户之间计算用户相似度
def compute_similer_twouser(self):
    # 记录与其他用户相似度的字典
    user_similer_list = dict()

    for i in self.users_list:
        user_similer_list[i] = dict()
        for j in self.users_list:
            # 去除自身相似度
            if i != j:
                # 两用户间物品并集，并计算余弦相似度
                similer_item = self.user_item_dict[i] & self.user_item_dict[j]
                user_similer_list[i][j] = round(
                    len(similer_item) / (len(self.user_item_dict[i]) * len(self.user_item_dict[j])) ** 0.5, 3)
    return user_similer_list
```

Demo

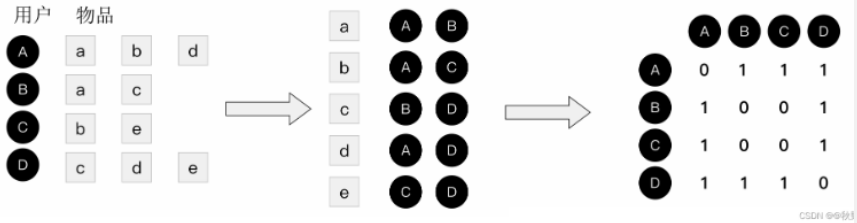
step 3 基于用户的协同过滤算法——倒查表

```python
#     优化后的倒查表方式计算用户相似度
def computer_similer_backform_normal(self):
    # 记录物品的所有用户的字典
    item_user_dict = dict()
    # 建立物品到用户的倒查表
    for i, items in self.user_item_dict.items():
        for j in items:
            if j not in item_user_dict.keys():
                item_user_dict[j] = list()
            item_user_dict[j].append(i)
    # 建立用户物品二维矩阵
    user_matrix = np.zeros((len(self.users_list), len(self.users_list)), dtype=int)
    for item, users in item_user_dict.items():
        for user1 in users:
            for user2 in users:
                if user1 != user2:
                    # 记录不同用户间共同物品的数目
                    user_matrix[self.users_list.index(user1)][self.users_list.index(user2)] += 1
    # 计算用户相似度
    user_similer_list = dict()
    for user1 in self.users_list:
        user_similer_list[user1] = dict()
        for user2 in self.users_list:
            if user1 != user2:
                # 利用二维矩阵存储的用户相似度进行计算
                user_similer_list[user1][user2] = round(
                    user_matrix[self.users_list.index(user1)][self.users_list.index(user2)] /
                    (len(self.user_item_dict[user1]) * len(self.user_item_dict[user2])) ** 0.5, 3)
    return user_similer_list
```

Demo

step 4 基于用户的协同过滤算法——惩罚项

该物品产生过行为的用户列表。同时在改良方法中为降低热门物品的影响，引入惩罚项。



$$w_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \frac{1}{\log(1 + |N(i)|)}}{\sqrt{|N(u)||N(v)|}}$$

该公式通过 $\frac{1}{\log(1 + |N(i)|)}$ 惩罚了用户 u 和用户 v 共同兴趣列表中热门物品对他们相似度的影响。

```python
# 建立用户矩阵
self.user_matrix = np.zeros((len(self.users_list), len(self.users_list)))
for item, users in item_user_dict.items():
    for user1 in users:
        for user2 in users:
            if user1 != user2:
                # 对热门物品进行惩罚
                self.user_matrix[self.users_list.index(user1)][self.users_list.index(user2)] += 1 / math.log(
                    1 + len(users))
```

Demo

step 5 基于用户的协同过滤算法——计算感兴趣程度

2. 计算用户对物品的感兴趣程度

$$p(u,i) = \sum_{v \in S(u,K) \cap N(i)} w_{uv} r_{vi}$$

其中， $S(u,K)$ 包含和用户 u 兴趣最接近的 K 个用户， $N(i)$ 是对物品 i 有过行为的用户集合， w_{uv} 是用户 u 和用户 v 的兴趣相似度， r_{vi} 代表用户 v 对物品 i 的兴趣。因为使用的是单一行为的隐反馈数据，所以所有的 $r_{vi} = 1$ 。


```python
# 计算指定用户的推荐物品
def compute_interest_item(self, user):

    user_similer_list, item_user_dict = self.computer_similer_backform_punish()
    # 本处相似用户集合大小为3，取相似度高的前三名用户
    sort_coor = np.argsort(self.user_matrix[self.users_list.index(user)])[::-1][:3]

    recommend_item = dict()
    for item in item_user_dict.keys():
        recommend_item[item] = 0
        for i in sort_coor:
            # 推荐不在指定用户物品集中，但在相似用户物品集中的物品，r=1
            if item not in self.user_item_dict[user] and item in self.user_item_dict[self.users_list[i]]:
                recommend_item[item] += self.user_matrix[self.users_list.index(user)][i]

    # 对推荐的物品按照感兴趣程度降序
    recommend_item = {k: v for k, v in sorted(recommend_item.items(), key=lambda item: item[1], reverse=True) if
                      v != 0}
    return recommend_item
```

Demo

结果输出

两两用户之间计算用户相似度:
{'A': {'B': 0.667, 'C': 0.667, 'D': 0.667, 'E': 0.333}, 'B': {'A': 0.667, 'C': 1.0, 'D': 0.667, 'E': 0.333}, 'C': {'A': 0.667, 'B': 1.0, 'D': 0.667, 'E': 0.333}, 'D': {'A': 0.667, 'B': 0.667, 'C': 0.667, 'E': 0.667}, 'E': {'A': 0.333, 'B': 0.333, 'C': 0.333, 'D': 0.667}}
优化后的倒查表方式计算用户相似度:
{'A': {'B': 0.667, 'C': 0.667, 'D': 0.667, 'E': 0.333}, 'B': {'A': 0.667, 'C': 1.0, 'D': 0.667, 'E': 0.333}, 'C': {'A': 0.667, 'B': 1.0, 'D': 0.667, 'E': 0.333}, 'D': {'A': 0.667, 'B': 0.667, 'C': 0.667, 'E': 0.667}, 'E': {'A': 0.333, 'B': 0.333, 'C': 0.333, 'D': 0.667}}
惩罚热门物品和倒查表方式计算用户相似度:
{'A': {'B': 0.448, 'C': 0.448, 'D': 0.448, 'E': 0.24}, 'B': {'A': 0.448, 'C': 0.655, 'D': 0.414, 'E': 0.207}, 'C': {'A': 0.448, 'B': 0.655, 'D': 0.414, 'E': 0.207}, 'D': {'A': 0.448, 'B': 0.414, 'C': 0.414, 'E': 0.448}, 'E': {'A': 0.24, 'B': 0.207, 'C': 0.207, 'D': 0.448}}
推荐物品:
{'a': 1.344}

'A': ['b', 'd', 'e'],
'B': ['a', 'b', 'd'],
'C': ['a', 'b', 'd'],
'D': ['a', 'd', 'e'],
'E': ['a', 'c', 'e']

Demo

结果输出

```
两两用户之间计算用户相似度：
{'A': {'B': 0.667, 'C': 0.667, 'D': 0.667, 'E': 0.333, 'F': 1.0, 'G': 0.667, 'H': 0.667, 'I': 0.333, 'J': 0.333, 'K': 0.667, 'L': 0.333, 'M': 0.667, 'N':
优化后的倒查表方式计算用户相似度：
{'A': {'B': 0.667, 'C': 0.667, 'D': 0.667, 'E': 0.333, 'F': 1.0, 'G': 0.667, 'H': 0.667, 'I': 0.333, 'J': 0.333, 'K': 0.667, 'L': 0.333, 'M': 0.667, 'N':
惩罚热门物品和倒查表方式计算用户相似度：
{'A': {'B': 0.275, 'C': 0.275, 'D': 0.278, 'E': 0.126, 'F': 0.401, 'G': 0.249, 'H': 0.249, 'I': 0.123, 'J': 0.123, 'K': 0.249, 'L': 0.126, 'M': 0.275, 'N'
推荐物品：
{'c': 0.556}
```

'A': ['b', 'd', 'e'], 'B': ['a', 'b', 'd'], 'C': ['a', 'b', 'd'], 'D': ['a', 'd', 'e'],
'E': ['a', 'c', 'e'],'F': ['b', 'd', 'e'], 'G': ['b', 'c', 'e'], 'H': ['b', 'c', 'e'],
'I': ['a', 'b', 'c'], 'J': ['a', 'b', 'c'], 'K': ['a', 'b', 'e'], 'L': ['a', 'c', 'e'],
'M': ['a', 'b', 'd'], 'N': ['a', 'b', 'c'], 'O': ['a', 'b', 'c'], 'P': ['c', 'd', 'e'],
'Q': ['a', 'b', 'e'], 'R': ['b', 'c', 'e'], 'S': ['a', 'c', 'e'], 'T': ['c', 'd', 'e']

参考文献
References

Below are some of our references.

References 01

黄旭彬. 一种基于用户行为预测的智能家居系统的实现[J]. 智能计算机与应用,2020,10(7):311-313,317. DOI:10.3969/j.issn.2095-2163.2020.07.075.

References 02

王吉鳖. 基于数据的APP用户行为预测算法研究[D]. 浙江:浙江大学,2019.

References 03

冯辉,邓明,陈宝国. 基于大数据平台的用户行为预测系统设计[J]. 赤峰学院学报（自然科学版）,2020,36(12):17-21. DOI:10.3969/j.issn.1673-260X.2020.12.006.

References 04

傅晨波,夏镒楠,岳昕晨,等. 一种融合信息网络结构的数据增强行为预测算法[J]. 小型微型计算机系统,2022,43(3):568-573. DOI:10.20009/j.cnki.21-1106/TP.2020-0919.

感谢聆听