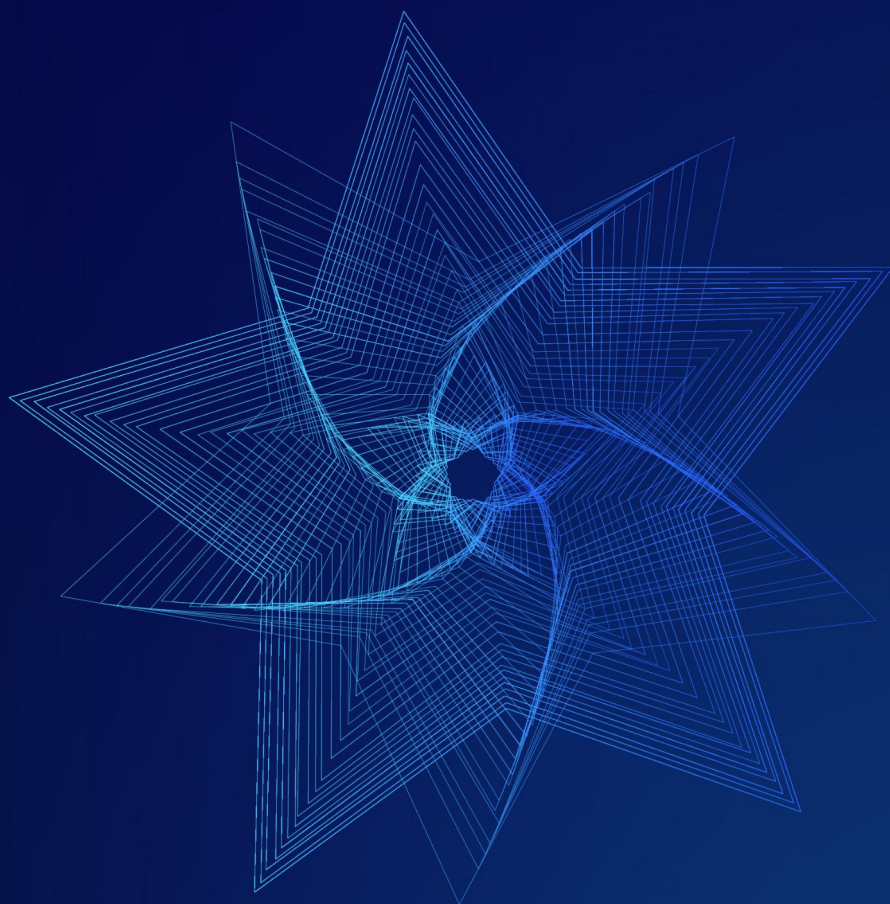


智能风控 典藏版合集

DataFunTalk系列技术合集：智能风控篇



本书内含：

DataFunTalk精选最新智能风控技术文章合集
21篇，380+页，8W字

DataFunTalk 成就百万数据科学家！

关于 DataFunTalk：

DataFunTalk 专注于大数据、人工智能技术应用的分享与交流，发起于 2017.12，至今已在全国 7 个数据智能企业和人才集中的城市举办了近 100 场线下技术分享和两场千人规模的行业峰会，邀请超过 600 位工业界专家和 40 位知名学者参与分享，近 30000 人次从业者参与线下交流。合作企业包含 BAT 等大厂，以及知名互联网公司和数据智能创业公司。其运营的公众号 DataFunTalk 共生产原创文章近 500 篇，超百万阅读，10 万+精准粉丝。

配套福利：

本电子书合集，为更好的方便大家结合视频和文章进行学习，特为大家提供了 DataFunTalk 视频回放年卡**专属优惠福利**，感兴趣的小伙伴，欢迎通过下页海报二维码，领取专属优惠券，成为我们的视频回放年卡会员，**大部分文章都可结合视频进行学习**：



DataFun视频回看年卡

内容简介

近3年累计大数据，算法实战案例300+

每年至少新增案例分享200+

专属交流群，配套PPT下载

可在公众号发布内推信息，助力招聘

其他权益

推荐算法

精排、粗排、召回、排序模型、推荐系统架构

广告算法

点击率预估、定向广告、广告冷启动、展示广告

搜索算法

Query理解、语义计算、搜索排序/召回

知识图谱

大规模知识图谱构建、电商知识图谱、金融知识图谱

用户画像

用户画像方法论、标签生产引擎、用户生命周期管理

大数据

数据架构、数据仓库、数据治理、数据应用、OLAP

~~原价199元/年~~
年度合集专属价**133元/年**





高级私人定制服务

职业 “问” 道

一流公司高级专家为你答疑解惑

想找心仪的工作？想升职加薪？

专家为你构建能力框架

想解决项目中的难题？想顺利推进工作？

专家为你制定落地方案

想让公司的营收进行指数级增长？

专家为你提供发展之道



扫码了解服务详情



目录

模型可解释性在保险理赔反欺诈中的实践	7
图算法在网络黑产挖掘中的思考	21
联邦学习与安全多方计算	36
58 同城风控平台演进	59
度小满金融大数据风控模型实践	70
Fraudar 算法在京东关系网络反欺诈中的应用	85
携程金融自动化迭代反欺诈模型体系	100
微众银行在联邦推荐算法上的探索及应用	113
机器学习中的隐私保护	144
关系图谱在贝壳找房风控体系的应用与实践	162
新一代联邦学习技术及应用实战	178
构建端到端的联邦学习 Pipeline 生产服务	212
风控建模流程：以京东群体感知项目为例	244
信贷业务风控策略简介	266
机器学习在信贷风控建模中的优势和挑战	280
度小满智能获客系统实践	297
金融风控反欺诈之图算法	307
机器学习在反欺诈中应用	324
混合、安全对抗下人工智能设计原则	336



自动化特征工程和自动建模在风控场景的应用.....	349
爱奇艺流量反作弊的“术”与“道”	366



模型可解释性在保险理赔反欺诈中的实践



分享嘉宾：张洪涛

编辑整理：马韶光

出品平台：DataFunTalk

导读：真实的理赔产品中，绝大部分客户是就医或者患病诊断患病之后去找保险公司进行申请理赔，但是其中有一小部分客户他会制造假的就医记录或者带病投保到保险公司骗取保险金，怎么样保证正常投保客户能够正常理赔的保险权益，同时避免骗保客户给公司带来的经济上的损失成为了一个十分关键的问题。近几年，随着 AI 的蓬勃发展 and 数据的不断积累，从算法技术来讲，很多行业在欺诈风险识别中表现出了非常优异的效果，产生了一些非常好的价值。但由于算法本身属性的原因，模型的结果却难以解释，但这在真实的业务场景中却非常关键。这次带来的分享内容，就是我们在实际的保险理赔反欺诈场景中的一个模型，可解释性的一些探索经验，希望能够给大家带来一些启发，或者一些其他的帮助。



今天的介绍会围绕下面四点进行展开：

- 模型可解释的整体背景
- 目前学术界和工业界现有的一些模型解释方法，例子以及对应原理
- 模型可解释性在实际的场景中的一个具体的应用和实施方案
- 对模型可解释性的简单的展望

01

模型可解释性的整体背景

1.可解释性背景

模型的解释性，在模型的整个生命周期中具有重要作用。

优化方法

模型解释，是一种重要的模型badcase分析手段，优化模型效果。

可信度

模型可解释，提升模型预测结果的可信度，进而影响推广使用。

指导性

模型可解释，对业务开展和决策具有明确的指导性作用。

然而，由于算法模型本身的黑盒属性，预测结果难以解释。

 2020 年终·大会
DATAFUNTALK

软件工程学上我们经常用到一个术语叫软件的生命周期，这里把它用在模型上，按照模型的生命周期来看，将模型的可解释性总结为三个方面，或者说三个不同时期的作用。



- 模型的开发和构建。企业模型解释在优化模型的期间，是一个优化模型的一个非常重要的手段。在实际的模型构建的过程中，这种 bad case 分析寻找模型优化方向还是一个比较困难的问题，如果模型可解释，可以对出错的样本采取针对性的措施对模型进行优化。
- 我们的模型试运行上线期间。模型的可解释性能够提升模型的可信度，同时有利于业务的推广。
- 模型推广期间，模型预测真正人融入到具体的业务环节流程之中。之后，我们希望这个模型能够可以解释模型预测值的解释内容，能够对后续的一些业务上的处理环节带来指导性的作用。

02

模型解释性方法

1. 模型解释性方法



2.1模型解释性方法



- 全局的解释方法(全局就是我们考察更偏重对模型整体的一个预测的解释)：第一个特征权重，每个特征的权重代表着特征对预测结果的一个影响程度，本质上其实这个权重系数就是一种显示性。第二个信息增益，可以计算出某个特征对预测结果带来的信息量。第三个特征重要性，特征重要性很大程度上就可以解释模型预测的一个判断依据。
- 局部的解释方法（对这个单条的预测进行解释）：第一个 LIME，它本质上是用线性模型在一个局部的样本空间上进行一个模拟。第二个 DeepLIFT，计算每个特征值的一个基准值，然后计算某个特征取值相对于基准值的变动对于预测结果带来的一个提升和影响。第三个 Shap，它本质上是基于博弈论的一种计算方法，计算也是计算特征的贡献。

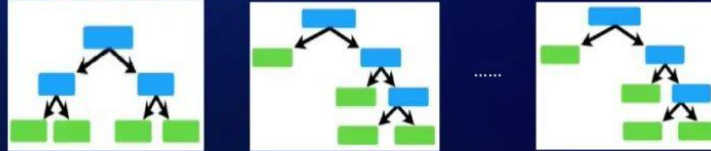
2. 模型解释性方法-特征重要性



2.2模型解释性方法-特征重要性

定义：全局可解释性是指能够基于完整数据集上的预测结果和特征之间的条件交互来解释和理解模型决策。解释整个模型行为。

举例：xgboosting特征重要性。



Weight : the number of times a feature is used to split the data across all trees.

Gain , Cover.



2020 年终·大会
DATAFUNTALK

从定义上看，全局可解释性是指能够基于完整数据集上的预测结果和特征之间的条件交互来解释和理解模型。简单地理解来，它就是解释整个模型的行为。举个常见的特征重要度的算法，是每个特征被引用的次数，对次数的值进行一个排序，次数越多，排序越靠前，对应特征越重要，这样就作为模型一种非常直观的解释方法。

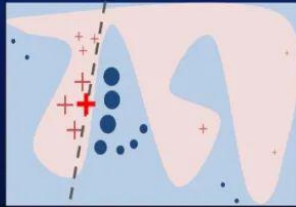
3. 模型解释性方法-lime



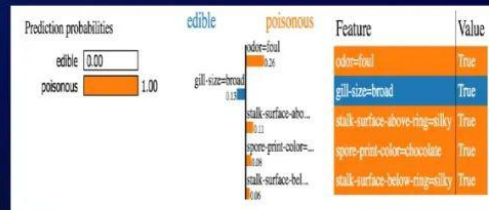
2.3模型解释性方法-lime

定义：专注于该数据点并查看该点周围的特征空间中的局部子区域，并尝试基于该局部区域理解该点的模型决策。解释单个预测。

举例：Lime方法。



某个样本附近生成采样数据，训练线性模型，辅助解释



Lime : "Why Should I Trust You" Explaining the Predictions of Any Classifier



2020 年终·大会
DATAFUNTALK

lime 方法其实是非常流行或者非常常见或者经常被大家拿过来讨论的一种方法。它是一种局部解释方法，局部解释性专注于该数据点，并查看该点周围特征空间中的局部次区域，并尝试基该局部及区域去理解该点的模型决策。简单理解为它是解释单个预测样本。以 ppt 中的红叉对应的样本点为例，在选取的样本点的附近选取一定数量的样本点，利用这些样本点重新训练一个简单的模型，如线性模型，然后利用这种解释性较好的方法来解释预测样本。

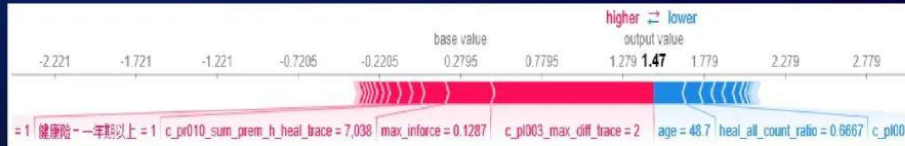
4. 模型解释性方法-shap



2.4模型解释性方法-shap

定义：专注于该数据点并查看该点周围的特征空间中的局部子区域，并尝试基于该局部区域理解该点的模型决策。解释单个预测。

举例：Shap贡献值。



将某一特征与其他所有的特征子集进行博弈比较，计算其对于其他特征子集对预测结果影响。

$$\text{预测值} = 1 / (1 + \exp(-\sum(\text{贡献值})))$$

Shap : A Unified Approach To Interpreting Model Predictions



2020 年终·大会
DATAFUNTALK

修改方法中将某一特征与其他所有特征子集进行博弈比较，计算其对于其他特征子集对预测结果的影响。预测值和各个特征的贡献值之间存在着这样的映射关系，预测的所有的贡献值求和，代入如上图所示方程式中，得到对应的一个预测样本。其中红色代表对于预测结果具有最大的优先贡献，蓝色的与之相反，对应的是负向的贡献，框的长度代表贡献的绝对值的大小。

5. 模型解释性方法-选型



2.5模型解释性方法-选型



实际的使用过程当中，还需要面对一个选择的问题。两个方面需要考虑：

- 解释方法的适用范围：在实际场景中，其实我们能希望某一种解释方法与模型无关，或者说至少适用于实际的我们使用的模型算法。
- 解释方法的运行效率：在实际场景中，需要根据场景对运行效率的要求做出适当的选择。对于那种实时服务的响应要求的服务，或者多长时间之内必须要反馈结果，那么这个时候我们就需要考虑这个方法的解释方法的一个运行效率。

综合以上因素考虑，Shap 方法具有一定的综合性优势。理赔反欺诈实践中也采用了这样上的这样一种方法。

03

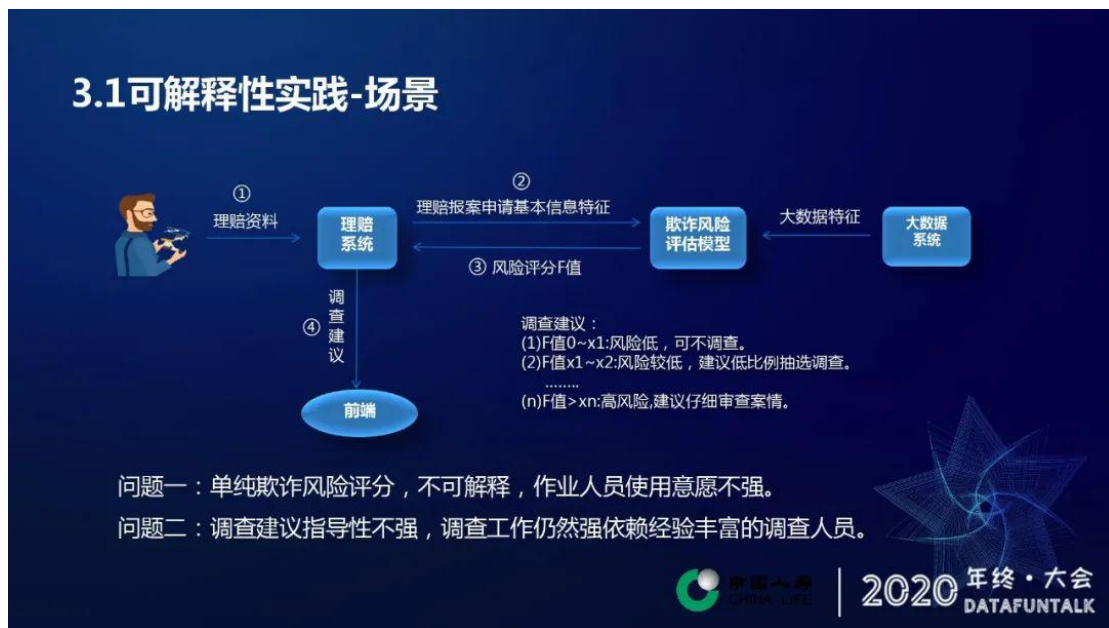
可解释性实践

14

峰会回看，扫码领取专属优惠



1. 可解释性实践-场景



保险理赔反欺诈的一个具体应用场景：

- 客户会向理赔系统提交理赔申请
- 理赔系统将理赔相关信息会传入反欺诈模型接口
- 接口根据理赔信息，一些基本的信息，以及在我们大数据系统中的一些既往历史数据，然后进行整合，进行一个综合的模型预测。

遇到的两个问题

- 一个单纯的欺诈风险评分是不可解释的。
- 模型预测的结果对于调查建议的指导性不强。

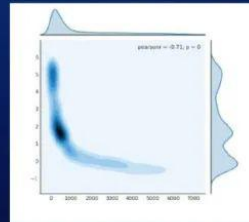
2. 可解释性实践-方法



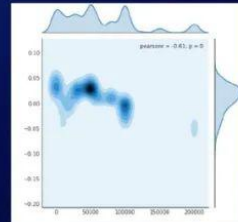
3.2可解释性实践-方法

对历史理赔欺诈案件特征值（横坐标）和特征贡献shap值（纵坐标）进行统计分析。

举例：



- (1) 该特征值与欺诈风险负相关；
- (2) 取值为800附近，欺诈高发。



- (1) 该特征值与欺诈风险负相关不明显；
- (2) 取值为5000附近，欺诈高发。

分析整理出与欺诈风险具有明显相关关系的特征，及对应的规律。供理赔审核员参考。



2020 年终·大会
DATAFUNTALK

针对上述问题，采用 shap 方法进行解决。对历史理赔欺诈案件特征值和特征贡献的 shap 值进行了一下统计分析。由上述 ppt 中左边图看出，横坐标代表某一个特征的取值，纵坐标的是特征的贡献值，图中的每一个点代表是一个欺诈样本，我们通过这样一个热图可以发现从这张图上发现两点内容，一个是特征与其欺诈评分或者欺诈程度是呈一个负相关的，因为我们明显能看到大概有一个随着特征值的增加，有一个递减的这样一个过程，第二个特征特征值在取值在小于 1000800 附近，这个地方就是它的热度是最高的，说明在附近欺诈案件是非常高发的，因为这个颜色是最重的，欺诈样本的密度也最高。从右图中可以看出特征与欺诈程度没有明显的相关性，但在特征值在取值在 5000 附近的时候，欺诈案件是非常高发的。这两个规律可以反馈给我们的业务人员去使用，或者是做一些启发性的东西，方便他们进行后续的使用。



3.3可解释性实践-方法



除了一维特征，也可以和业务人员共同进行各个特征交叉情况下对多维特征进行 shap 值分析。举个简单例子，上述图中是两个类别特征的交叉分析，其中，每个特征都只有 0，1 取值。从图中可以看到，两个特征变量都取 1 时欺诈风险较高。我们可以将分析结果我们会反馈给我们的理赔作业人员，有由于作业人员来根据实际的工作经进行一个验证，然后去整理出具体的书面的表达话术。另一方面这个时候我们的理赔作业人员也有可能自发地去总结一些他自身的经验，然后通过这样一个反馈给我们的算法工程师，不断的循环往复，从而得到更丰富的欺诈经验。

3. 可解释性实践-应用



3.4可解释性实践-应用



2020 年终·大会
DATAFUNTALK

我们可以从上面看这个例子，最上面的表格是我们理赔反欺诈模型考察理赔申请的所有特征，在特征输入模型还有上和价值分析的之后，会得到第二行这样一个结果，通过 shap 分析和风险评估预测会得到我们的欺诈值期价值显示是 0.85，代表被欺诈的风险是 0.85，然后同时通过 shap 方法计算出所有的特征的贡献值。然后我们拿到计算结果之后，会做一定程度的筛选。

根据业务规则，筛选出欺诈值较高风险的样本，筛选出来之后，我们再去筛选它的特征，根据贡献值去做，筛选出具有对特征值具有正向贡献的几个特征值，然后拿到这几个特征及其特征取值进行检索。右边这张图是根据分析做出来的经验规则，比如说最简单的开始，可能从一味的去考虑特征一小于某一个数的时候，它可能就是一种欺诈的迹象。这个时候我们的理赔作业人员总结出来，这样一个经验和话术就会显示在这里，然后扩充



到二维，比如特征一小于 Y 特征 M 大 P 或者说其他更高维，特征一小于 X 特征二小 P...特征 N 等于 R 的时候，这样的话也有对应的一个话术显示，通过这样一步的检索过程，就是我们把这些能够碰撞上的规则以及经验显示到给我们前端从作业人员做参考，指导他们进行后续的业务开展。

04

可解释性展望



简单展望：

- 图技术：图这种数据结构的话，它具有比较天然的可解释性的优势，比如说它有自己的实体，有自己的属性，实体属性之间它有相互的之间的一个关系。当然这两种技术的话可能是差别比较大，找到这



样一种桥梁，能够让他们两个联合起来，或者说一种方式能够相互配合起来，也是一个比较困难的事情。

- 模型蒸馏：模型蒸馏也是被广泛采用模型整理的一个整体思路就是用一个结构简单的模型在保证准确率下降不是太大，或者说在我们一个可接受的范围之内去尽量的去简化这模型，把这个模型变得更简单，因为我们更加简单的模型的话，其实它是更容易去解释的，同时也提高了运行效率。
- 领域知识：在短期内与领域专家和领域的领域知识相结合，设计一套比较可行的落地方案，仍然是一种比较有效的选择。我们能够用这种方式快速落地，快速的实现模型可解释性。

今天的分享就到这里，谢谢大家。



图算法在网络黑产挖掘中的思考

分享嘉宾：Harry 高级研究员
编辑整理：蔡鑫隆
出品平台：DataFunTalk

导读：虚拟网络中存在部分黑产用户，这部分用户通过违法犯罪等不正当的方式去谋取利益。作为恶意内容生产的源头，管控相关黑产用户可以保障各业务健康平稳运行。当前工业界与学术界的许多组织通常采用树形模型、社区划分等方式挖掘黑产用户，但树形模型、社区划分的方式存在一定短板，为了更好地挖掘黑产用户，我们通过图表征学习与聚类相结合的方式进行挖掘。本文将为大家介绍图算法在网络黑产挖掘中的思考与应用，主要介绍：

- 图算法设计的背景及目标
- 图算法 GraphSAGE 落地及优化
- 孤立点&异质性
- 总结思考

01

图算法设计的背景及目标

1. 图算法设计的背景

在虚拟网络中存在部分的黑产用户，这部分用户通过违法犯罪等不正当的方式去谋取利益，比如招嫖、色情宣传、赌博宣传的行为，更有甚者，如毒品、枪支贩



卖等严重的犯罪行为。当前工业界与学术界的许多组织推出了基于图像文字等内容方面的 API 以及解决方案。而本次主题则是介绍基于账号层面上的解决方法，为什么需要在账号层面对网络黑产的账号进行挖掘呢？

用户黑产挖掘

学界 | 几张贴纸就让神经网络看不懂道路标志，伯克利为真实环境生成对抗样本

学界 | 神奇的面包机！谷歌造出对抗样本的实体版

学界 | 继图像识别后，图像标注系统也被对抗样本攻陷！

修改一个像素，就能让神经网络识别图像出错

- 恶意账号是网络黑产的源头
- 账号行为对抗门槛高
- 提前圈定可疑账号



原因主要有三：

- ① 恶意账号是网络黑产的源头，在账号层面对网络黑产的账号进行挖掘可以对黑产的源头进行精准地打击；
- ② 账号行为对抗门槛高，用户的行为习惯以及关系网络是很难在短期内作出改变的，而针对单一的黑产内容可以通过多种方式避免被现有的算法所感知，虽然黑产用户可能不懂算法，但其可以通过“接地气”的方式来干扰算法模型，譬如在图片上进行简单的涂抹，在敏感处打上马赛克，在图片处加上黑框，通过简单的对抗手段会对基于黑产内容的算法产生较大的影响；
- ③ 可以防范于未然，通过账号层面的关联提前圈定可疑账号，在其进行违法犯罪行为之前对账号进行相应的处理以及管控。



具体通过什么方式挖掘黑产账号？

首先，简单介绍下在推荐场景中应用。比如广告推荐，通常上，广告商会给予平台方用户的用户标签，用户存在用户标签之后，平台方则会将相关类别的用户找出，然后将广告推送给对应的用户；另一种方式是广告方提供种子包给平台方，平台方会找到相似的用户，然后将广告推送给相关的用户，常见的应用场景有 Facebook look like、Google similar audiences。

在黑产场景中与推荐场景中的应用类似，主要分为两个任务场景：

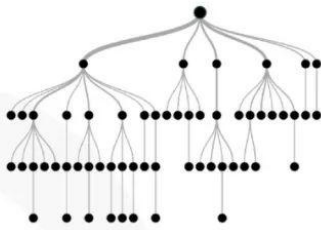
- ① 找出目标恶意类别用户。比如需要找出散播招嫖信息的用户，则给定该类用户招嫖的标签，类似于一个用户定性的问题；
- ② 黑产种子用户扩散，即利用历史的黑产用户进行用户扩散以及用户召回，可以通过染色扩散以及相似用户检索等方式完成。

针对恶意用户定性的传统方法，通常采用树形模型，比如说 XGboost、GBDT 等。这类算法短板显而易见，其缺乏对用户之间的关联进行考虑；另外一种用户召回方式为用户社区划分（相似用户召回），其中比较常用的社区划分算法有 FastUnfolding、Copra 等。这类算法的缺陷也相当明显，其由于原本社区规模小，所以最终召回的人数也少。且会存在多个种子用户在同一个社区的情况，难以召回大量可疑用户。



传统方法

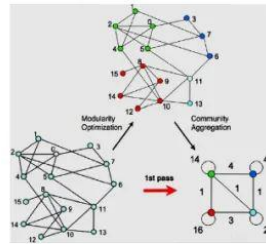
➤ 恶意用户定性



Tree-Based Model: Xgboost, GDBT, etc.

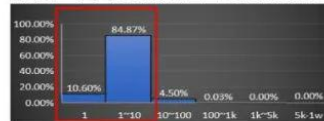
短板：缺乏对用户之间关联的考虑

➤ 用户社区划分



FastUnfolding, Copra, etc

短板：社区规模极小，应用效果差



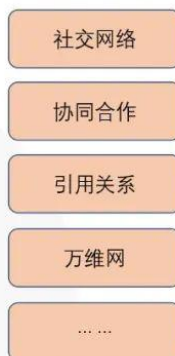
$$Q = \sum_c [\frac{\sum_{in} p}{2m} - (\frac{\sum_{tot} p}{2m})^2]$$

$$p = \frac{2e}{n(n-1)} - \frac{2E}{N(N-1)}$$

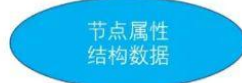
因此，通过图表征学习与聚类相结合的方式召回。通过图表征学习将图结构的节点属性以及结构特征映射到一个节点低维空间，由此产生一个节点特征，然后再去进行下游的任务，如用户定性即节点分类等。其中，图表征学习的关键点在于在进行低维的映射当中需要保留原始图的结构和节点属性信息。

图表征学习

真实的图结构



图表征学习



下游任务



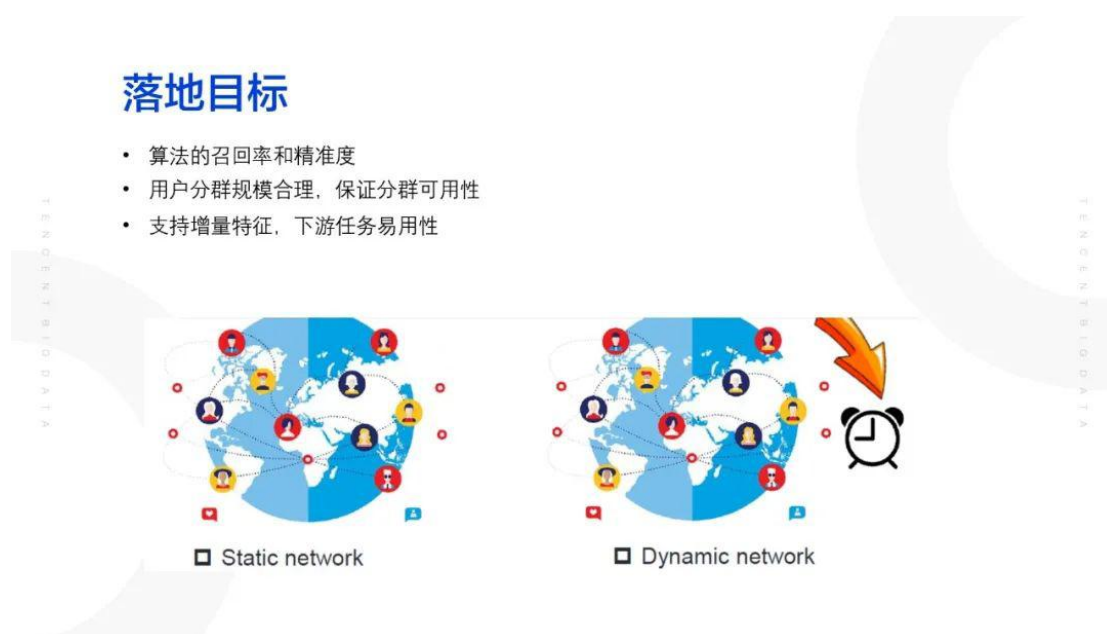
关键点：保留原始图的结构和节点属性信息



2. 图算法设计的目标

- ① 算法的覆盖率和精准度；
- ② 用户分群规模合理，保证分群的可用性；
- ③ 支持增量特征，下游任务易用性。

由于业务场景更多为动态网络，当新增节点时，如果模型支持增量特征，则不需要重复训练模型，可以极大的减少开发的流程，节省机器学习的资源，缩短任务完成的时间。



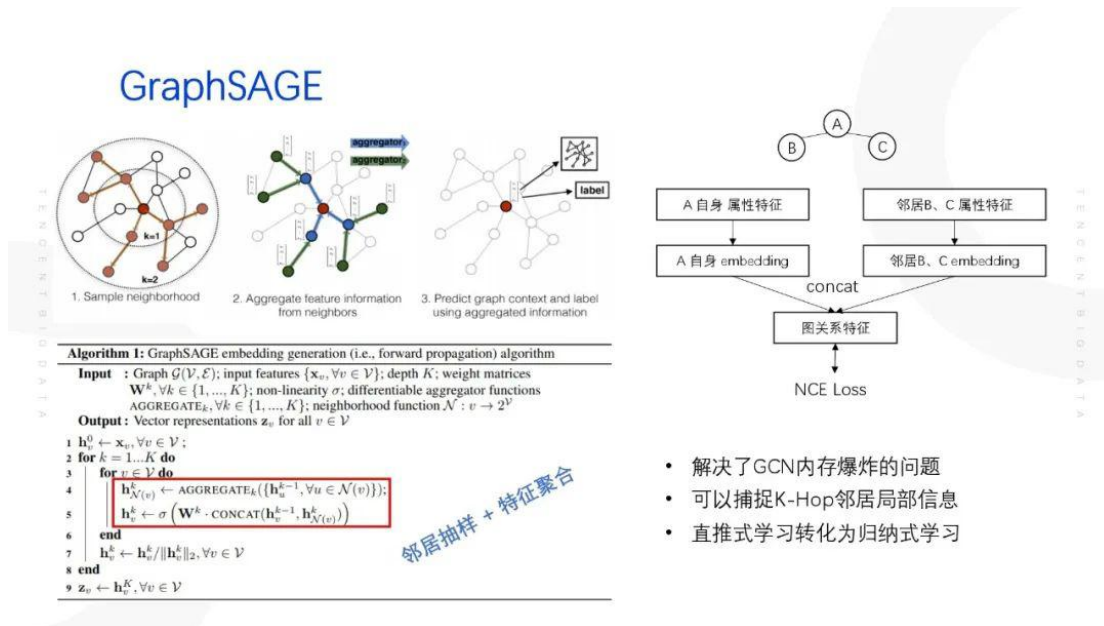
02

图算法 GraphSAGE 落地及优化

1. GraphSAGE 核心思想



GraphSAGE 核心思想主要为两点：邻居抽样；特征聚合。



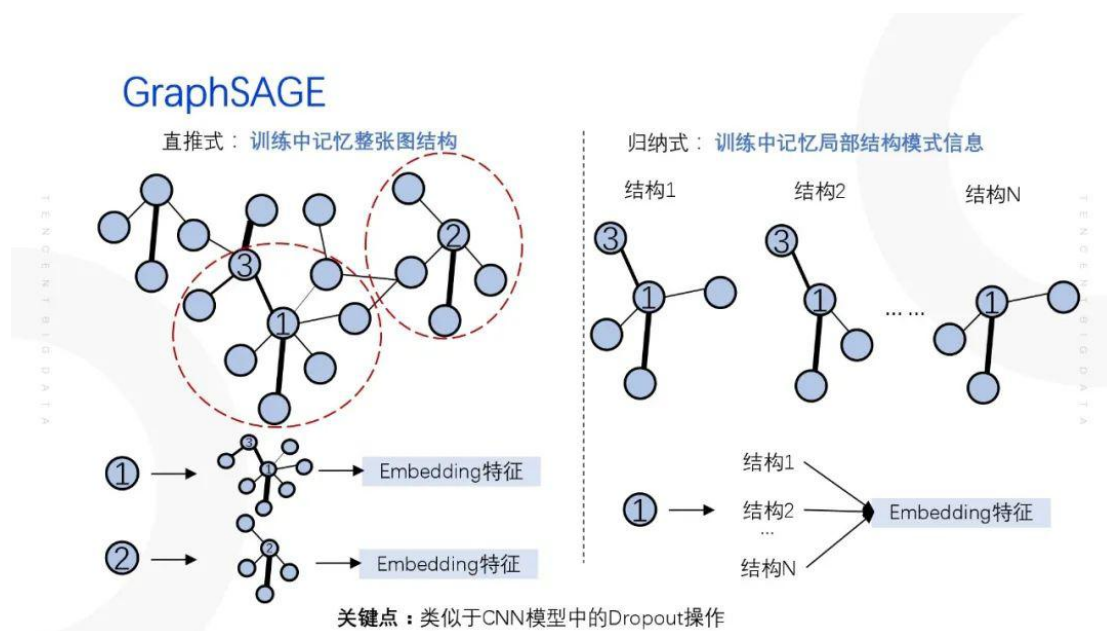
GraphSAGE 的聚合过程实际是节点自身的属性特征和其抽样的邻居节点特征分别做一次线性变换，然后将两者 concat 在一起，再进行一次线性变换得到目标节点的 embedding 特征。最后利用得到的目标节点的 embedding 特征进行下游的任务，训练的方式的可以采用无监督的方式，如 NCE Loss。

2. GraphSAGE 的优点

GraphSAGE 通过邻居抽样的方式解决了 GCN 内存爆炸的问题，同时可以将直推式学习转化为归纳式学习，避免了节点的 embedding 特征每一次都需要重新训练的情况，支持了增量特征。为什么通过邻居随机抽样就可以使得直推式的模型变为支持增量特征的归纳式模型呢？在原始的 GraphSAGE 模型（直推式模型）当中，节点标签皆仅对应一种局部结构、一种 embedding 特征。在 GraphSAGE



引入邻居随机抽样之后，节点标签则变为对应多种局部结构、多种 embedding 特征，这样可以防止模型在训练过程过拟合，增强模型的泛化能力，则可以支持增量特征。



3. GraphSAGE 的缺点

- ① 原 GraphSAGE 无法处理加权图，仅能够邻居节点等权聚合；
- ② 抽样引入随机过程，推理过程中同一节点 embedding 特征不稳定；
- ③ 抽样数目限制会导致部分局部信息丢失；
- ④ GCN 网络层太多容易引起训练中过度平滑问题。

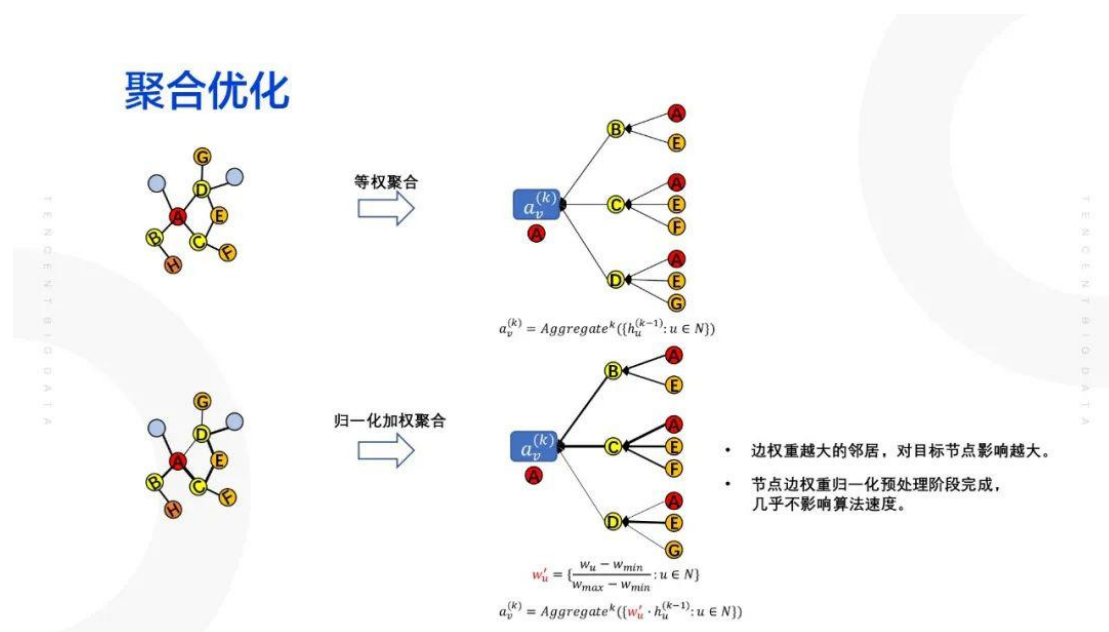
4. GraphSAGE 的优化

为解决上述 GraphSAGE 存在的缺点，对 GraphSAGE 进行优化。

- ① 聚合优化



解决等权聚合的问题。相对于直接将邻居节点进行聚合，将边权重进行归一化之后，点的邻居节点的特征进行点燃，最后再进行特征融合。这样做的好处主要有两点：边权重越大的邻居，对目标节点影响越大；节点边权重归一化在预处理阶段完成，几乎不影响算法速度。



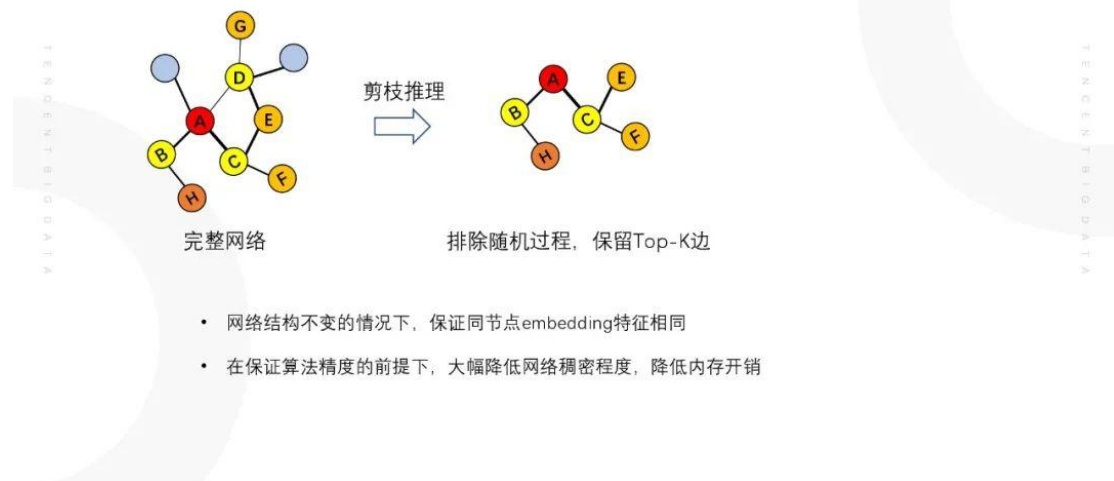
② 剪枝优化

解决 embedding 特征不稳定的问题。在训练的过程希望通过引入随机过程防止模型出现过拟合的现象，但是在模型的推理过程是想要去掉这样一个随机过程。直接对原始网络进行剪枝操作，仅保留每个节点权重最大的 K 条边，在模型进行推理的时候，会将目标节点所有的 K 个邻居节点的特征都聚合到目标节点上，聚合方式同样为加权的方式。这样做的好处主要有两个点：在网络结构不变的情况



下，保证同节点 embedding 特征相同；在保证算法精度的前提下，大幅度降低图的稠密程度，降低内存开销。

剪枝优化



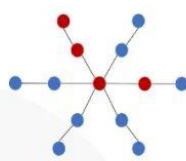
③ 采样优化

解决局部信息丢失以及训练过平滑的问题。主要通过 DGL 的抽样方式代替原有的抽样方式，具体的做法为：提前将每一个节点的属性特征与它所有的邻居节点的属性特征的均值进行 concat，这样可以使得每一个节点初始状态下已经包含了周围一些邻居节点的一些信息，通过这种方式，在采样相同节点的前提下，可以获得更多的局部信息。一般情况下，GCN 模型采用两层网络模型，当增加至第三层的时候则将存在内存爆炸的问题；当增加至第四层时，则将出现过平滑的问题，将导致特征分布去重，这样则导致节点没有区分性。而采用 DGL 采样，通过采样两层 GCN 模型而实际上采样了三层，而且不会出现过平滑问题。



采样优化

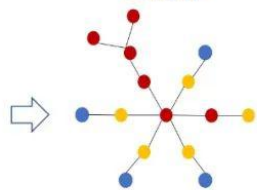
随机采样



● 不采样
● 采样

- 采样数阶受内存限制严重
- 采样存在信息损失

DGL采样



● 不采样
● 采样
● 弱采样

- 拓展采样数阶
- 降低采样信息损失

- 整体特征事先做加权预处理，预处理时间短
- 目标节点含所有邻居节点特征，提升局部信息
- N层模型可以捕获N+1阶邻居信息，可以抑制训练过度平滑的问题
- 聚类准确率提升**3~4个百分点**

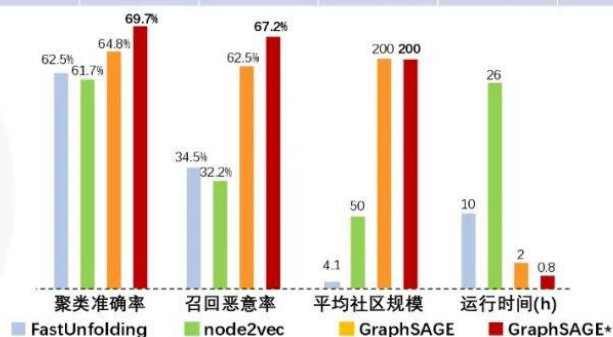
5. 效果评估

效果评估的指标主要有两个：聚类（社区）准确率；召回恶意率。相对于原有的 fastunfolding 以及 node2vec 从聚类准确率、召回恶意率、平均社区规模、运行时间作一个横向对比：

效果评估

• 数据量级：6千万节点，70亿边

	优化策略			应用效果	
	Weight	Prune	DGL	Clus AUC	Rec AUC
				64.18%	62.53%
	✓			65.78%	63.10%
	✓	✓		66.87%	65.43%
	✓	✓	✓	69.71%	67.18%



03

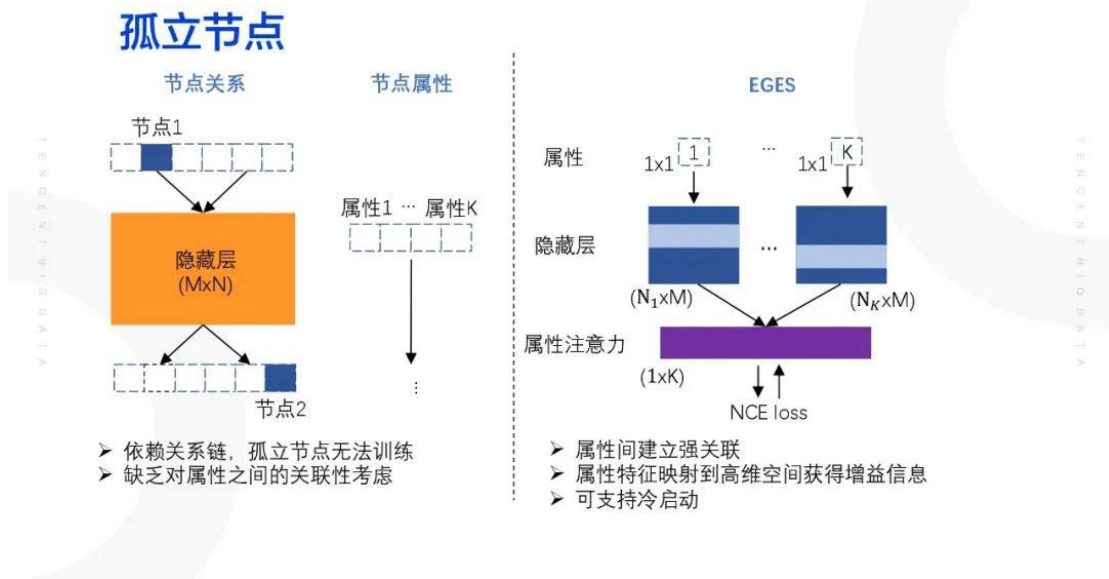
孤立点&异质性

1. 黑产挖掘场景中的孤立点的解决思路

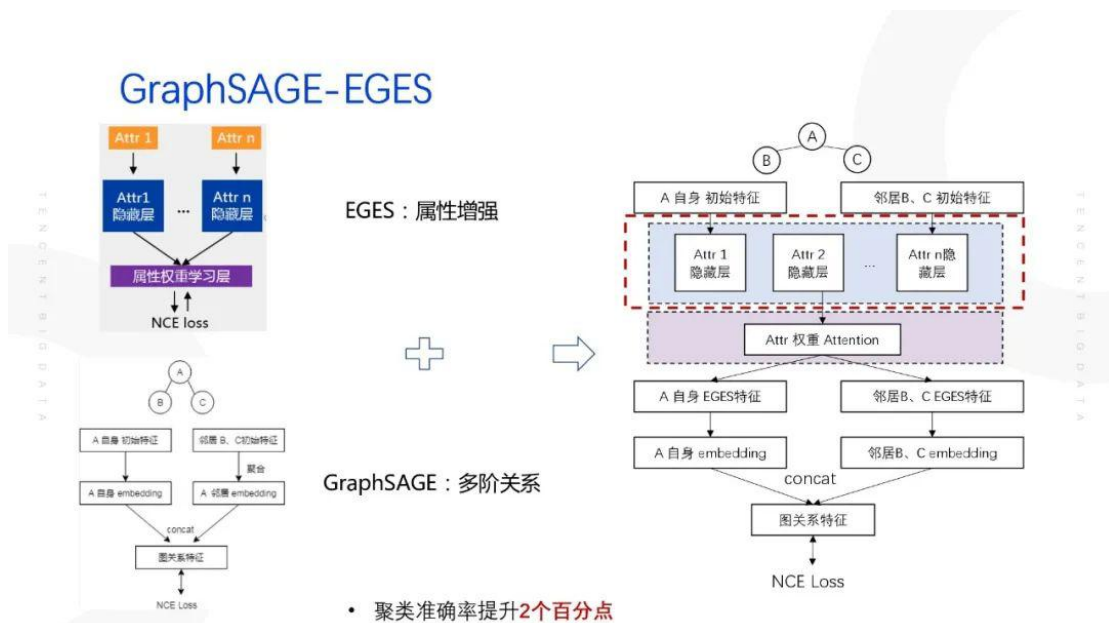
黑产用户在被处理后，通常会快速地申请新的账号或使用备用账号，因为在对黑产的挖掘过程中就不可避免地会出现孤立点，类似在推荐算法中的冷启动问题。

以 node2vec 算法为例，算法通常会通过游走去构造训练的节点段，那么如果孤立节点没有连边的话，节点是无法出现在训练集当中。为了解决该问题，引入一个解决推荐系统冷启动的算法——EGES，将每一个节点的属性特征映射到一个 embedding 特征，然后将每一个属性的 embedding 特征置于注意力层进行处理，比如将 N 个随机特征通过注意力加权，可以获得最终的一个节点层面的 embedding 特征，新增的节点将不再依赖于关系网络以及用户的一些交互行为，新增的节点可以通过自身的属性特征就直接获得我们的 embedding 特征，不需要考虑用户关系从而解决孤立点的问题。





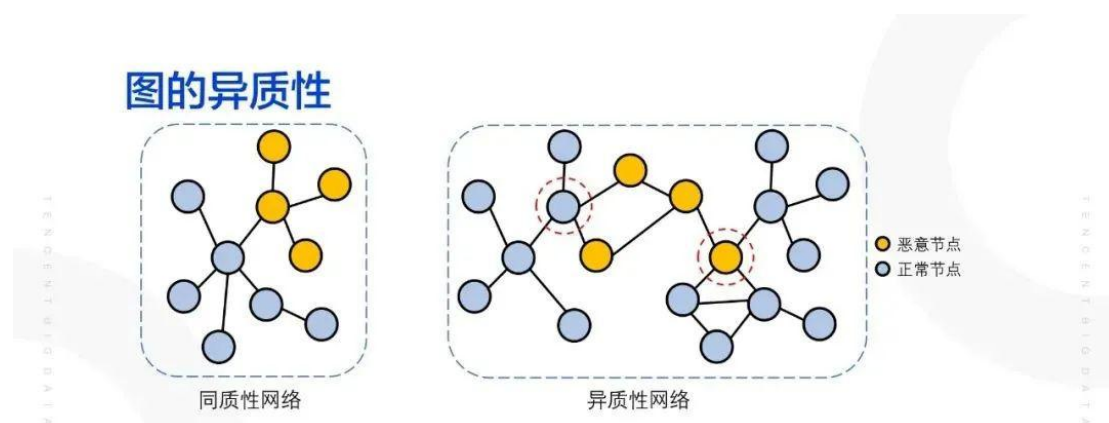
在具体落地过程中，提出了 GraphSAGE-EGES 算法，实际上是综合了两种算法的优势，GraphSAGE 的节点本身的初始特征将其替换成了 EGES 增强之后的属性特征，通过此类方式，最终的算法框架如下图所示：



此类算法可以提升聚类准确率 2 个百分点。

2. 黑产网络中异质性的解决思路

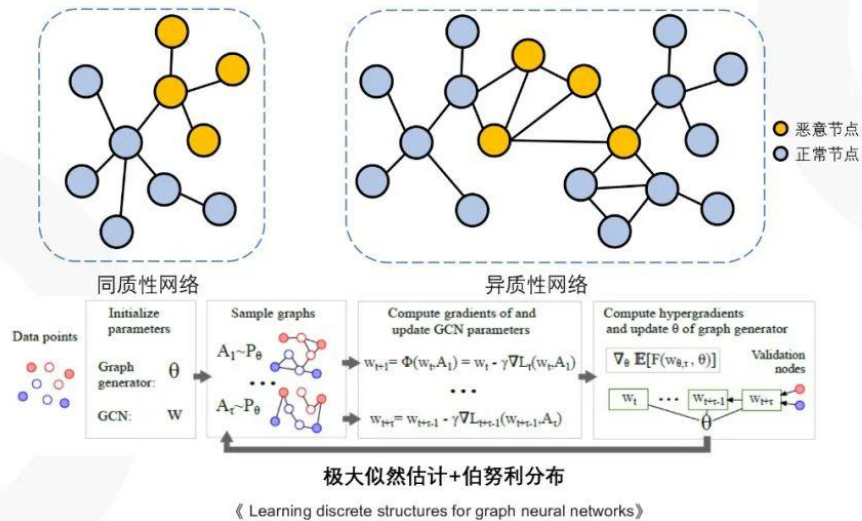
在正常的网络结构当中，一个用户的一阶邻居基本上都是同一类的用户，比如说在学术引用当中，一篇数据挖掘的论文，引用其的论文也多是与数据挖掘相关的。这一类的网络称之为同质性网络。但在黑产的关系网络当中，图的异质性就非常高了，黑产用户不仅仅与黑产用户相关，其也可以与正常用户建立关系，这种特殊的网络结构就会存在一些弊端，以下图异质性网络为例，圈住的正常节点的一阶邻居节点一半为恶意账号，算法进行预测、聚类时，该节点很多概率会被判定为恶意账号。圈住的恶意节点的一阶邻居 3 个皆为正常账号，算法进行预测、聚类时，该节点则大概率被判定为正常节点，导致算法的精度下降。



为了解决上述问题，需要去考虑网络的结构是否合理。为了构建合理的网络结构，需要将恶意账号与正常账号之间存在的联系剔除掉，并将恶意账号之间的联系进行一定的增强。



图结构学习



当网络结构合理时，算法进行预测、聚类时会更加准确，因此引入图结构学习的概念，尝试用 LDS 算法解决这类问题。

LDS 算法的思想：在训练 GCN 模型的参数的同时对网络的结构进行调整，在最初的时候给予一个网络结构（邻接矩阵），先固定 GCN 的模型，然后训练邻接矩阵，通过几轮迭代之后再固定邻接矩阵，再训练 GCN 模型，通过几轮迭代之后，可以得出一个合理的网络结构。总的来说，这个算法实际上就是一个极大似然估计以及伯努利分布的问题。在 LDS 算法学习邻接矩阵的时候实际就是学习两个点的邻边是否存在，实际上为一个 0-1 分布。最终通过网络结构以及节点的标签去预估在当前数据标签的情况下，更应该得到什么样的一个网络结构，以上即为该算法的核心思想。

实际上，在许多业务场景当中会存在许多不合理的图结构，甚者在某些业务场景中不存在关系信息，这样的话，在最初达不到完整网络的情况时，通常会使用 KNN



的方式对网络进行初始化，然后再去学习一个更加合理的网络结构，最终达到一个更好节点预测、聚类的目的。

04

总结思考

下面分享几点在算法落地以及算法选择中的一些工作总结与思考：

- ① 针对图算法这块，特征工程和图的构建方式是非常重要的。如果图的结构不合理的话，即使算法模型再强大、特征工程处理得再好，算法训练出的结果也不是最终理想的效果；
- ② 多数业务场景的区分度是不一样的，不存在一个普适的算法可以解决所有业务场景存在的问题，如上述的 FastUnfolding、node2vec 在某些特定的业务场景下效果可以比 GraphSAGE 的效果更好，所以在面临具体问题的时候，需要结合场景作算法选择以及优化；
- ③ 在工业界落地的算法通常比较直接、明了，这样的算法往往效果更好。

今天的分享就到这里，谢谢大家。



联邦学习与安全多方计算



分享嘉宾：洪澄博士 阿里巴巴 高级安全专家
编辑整理：徐华卿
出品平台：DataFunTalk、AI 启蒙者

导读：联邦学习和安全多方计算是当前跨机构数据协同的两类主流技术，本文将从基本思想、安全性、性能等多个方面介绍二者的区别，并介绍阿里在安全多方计算方面的最新成果。

01

联邦学习的发展历史

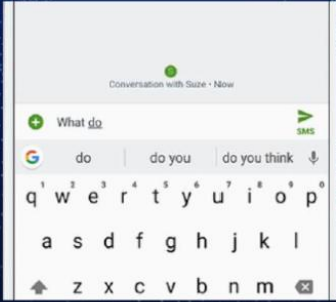


1. 联邦学习

DATAFUNCON
2020大数据 AI 的最新技术实践

阿里安全
ALIBABA SECURITY

- 联邦学习 (Federated learning , FL) 由Google于2016年提出
- 初衷是用于解决多个移动设备的分布式建模问题



例：Google Gboard安卓输入法预测

- 为了智能预测下一个词，需要针对大量用户的输入历史数据进行训练
- 设计目标：避免直接收集用户的输入历史，尽量在端上训练

联邦学习在 2016 年由谷歌提出，因为 Google 有安卓系统，需要解决多个安卓设备的分布式建模问题。其中，主要是针对输入法的建模，比如客户在安卓输入法中输入单词 “what”，或许他可能想继续输入 “do you think”，Google 输入法如果能自动联想出来，用户体验就会变得比较好，但是自动联想功能需要大量的用户数据才能学习出来，怎么获得这些用户数据呢？

一个比较粗暴的做法是用户输入了什么字就把这个字全部收集到云端上，但这种做法无疑是对用户隐私的一种破坏。由于谷歌崇尚不作恶，怎样在不收集用户输入文字的前提下，从而预测出用户接下来需要输入的文字？因此，产生了联邦学习。

2. 联邦学习用于多移动端分布式建模





联邦学习的设计优点就是用户数据尽量不离开用户自己的安卓设备，用户尽量在本地完成一部分的训练，然后把训练的梯度传到谷歌的云端，这样谷歌只看见一个梯度，它并没有获得这个用户的设备以前的聊天内容，这样在设计上有一种 privacy by design 的设计优点。有很多这样的安卓设备，比如：Parameter Server 设备是谷歌的云端服务器，它开始会有一个全局的初始化模型，云端服务器会把模型推到各个设备上，然后各个设备基于本地的数据来优化模型，得到一个更新的梯度，把这个更新的梯度发给服务器，服务器收到这么多梯度之后，会更新全局模型，然后发到这些设备上，这些设备又迭代，直到这个模型在某种程度上收敛为止，这就是联邦学习最开始的一个雏形。

3. 国内联邦学习与谷歌联邦学习的区别

区别一：



DATAFUNCON
2020大数据 AI的最新技术实践

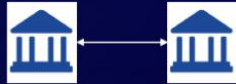
阿里安全
ALIBABA SECURITY

- 2018年国内开始引入Federated learning概念，主要区别1：

Google FL：主要面向
海量（百万+）移动设备的合作



国内主要是cross silo FL：
少量（如2个）机构之间的合作



大概在 2018 年左右，国内开始引入联邦学习概念，与谷歌的联邦学习相比有了一些发展和改变。两者主要的区别是谷歌的联邦学习主要是面向海量移动设备的数据之间的合作，但是国内主要是机构之间的合作，被称为 cross silo FL，一般都是两个或者三个机构之间的合作。但是，目前的应用主要以信贷或者广告为主，例如：两个或多个机构一起判断用户的信用，从而决定要不要借钱给他，或者要不要给他推一些广告。这种情况下参与方的数目实际上跟 Google 的联邦学习相比是有很大的降低的。

区别二：



DATAFUNCON

2020大数据 AI的最新技术实践



- 2018年国内开始引入Federated learning概念，主要区别2：

Google FL：主要面向数据的**横向**分割

	特征1	特征2	特征3	特征4
id 1	绿色	绿色	绿色	绿色
id 2	绿色	绿色	绿色	绿色
id 3	白色	白色	白色	白色
id 4	白色	白色	白色	白色

国内FL：主要面向数据的**纵向**分割

	特征1	特征2	特征3	特征4
id 1	绿色	绿色	白色	白色
id 2	绿色	绿色	白色	白色
id 3	绿色	绿色	白色	白色
id 4	绿色	绿色	白色	白色

Google 有很多的设备，每个设备上都有自治的一些样本，也就是说数据在多个参与方之间，它是横向分割的，比如说这个绿色的在一个设备上，这个白色的在另一个设备上就是横向的分割，每个都有一个完整的样本。

但是国内经常使用的联邦学习，主要是面向数据的纵向分割的。以信贷为例，其通常都是针对一个人的不同特征并把它们组合起来做联邦学习。比如说特征 1 与特征 2 在一个机构，特征 3 与特征 4 以及 label 是在另外一个机构，也就是说它主要是面向数据的纵向分割。当然横向分割这种应用国内同样存在，但是用的比较多的或者说比较赚钱的，还是在这种纵向的分割法上。

02

联邦学习面临的安全挑战



谷歌原版的联邦学习有什么样的安全挑战？而在国内，会面临什么样的新的安全挑战？

1. 谷歌原版的联邦学习的安全挑战

DATAFUNCON
2020 大数据 AI 的最新技术实践

阿里安全
ALIBABA SECURITY

- 梯度与原始数据的关系
 - 梯度 ∇W 的定义：本质上是一个函数

$$\nabla W_{t,i} = \frac{\partial \ell(F(\mathbf{x}_{t,i}, W_t), \mathbf{y}_{t,i})}{\partial W_t}$$

Client (x,y) → ∇W → Parameter server

- 已知梯度，如何求原始数据？
 - 攻击方法1：对于简单的F（如Logistic regression），可以直接解方程组（LHCH19）
 - 攻击方法2：对于复杂的F（如CNN），可以用ML方法求近似解（MSCS19, ZLH19）

► MSCS19: Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning, S&P 2019
► ZLH19: Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients, NIPS 2019
► LHCH19: Zhaorui Li, Zhicong Huang, Chaochao Chen and Cheng Hong. Quantification of the Leakage in Federated Learning, FL-NeurIPS 2019

首先是原版横向跨设备的联邦学习。因为它设计上只传梯度，梯度本质是一个函数，它是根据初始的模型以及本地的数据算出来的一个函数，那么这个函数可能是跟原数据是相关的，不能说有梯度就算不出原数据了，那多大程度上相关呢？其实算出来是有一定的难度，但是有一些学者也能算出来，比如说假设我们训练的模型是一个简单模型，比如逻辑回归，我们有了一堆梯度跟原始数据的这种关系，可以通过解方程组把这个未知数解出来的，这是我们在 NIPS 联邦学习 workshop 上的一个工作。如果这个模型比较复杂，解方程组算就变得不现实了。这时有一些其他的方法，比如我们用 machine learning 的优化方法来反向的优化求得一个近似的解，可能求不到精确的结果，但可以取到一个大致差不多的结果，这里有一个去年的 NIPS 的文章，它可以反向的从梯度求出人脸，然后这个



人脸可能只有若干个像素的区别。所以我们看到如果不保护这个梯度的话，本质上还是能推出原始数据的。

谷歌的解决方法：

① 加差分隐私

DATAFUNCON
2020大数据 AI的最新技术实践

阿里安全
ALIBABA SECURITY

- 如何防止从梯度反推原始数据
 - 方法1：加差分隐私，但是准确率会下降
 - 对于输入法这类产品来说或许可以接受
 - 但不适用于准确率是关键因素的产品

Client (x,y) → VW + noise → Parameter server

Inset image: Google search interface showing a keyboard and search results.

Google 应对的方法，主要通过加差分隐私，也就是说 client 上传到云端的梯度，它不直接上传，而是加一个 noise，但是准确率会下降。准确率的下降，对于 Google 输入法 somehow 是可以接受的，因为输入法的 Top3 顺序换了一下，或者推荐的东西错了一点，对于用户体验可能差别不大，但是对于我们这种用在广告或者信贷场景下，准确率差 1%就可能差很多很多钱，所以对我们来说加差分隐私不是一种能够接受的方案。

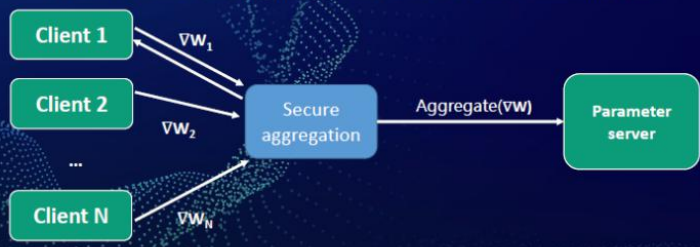
② secure aggregation



DATAFUNCON
2020大数据 AI的最新技术实践

阿里安全
ALIBABA SECURITY

- 如何防止从梯度反推原始数据
 - 方法2：Secure Aggregation
 - Server只能看到聚合之后的梯度，无法了解具体某个client的梯度



The diagram illustrates the Secure Aggregation process. On the left, three green boxes represent 'Client 1', 'Client 2', and 'Client N'. Arrows labeled ∇W_1 , ∇W_2 , and ∇W_N point from each client to a central blue box labeled 'Secure aggregation'. An arrow labeled 'Aggregate(∇W)' points from the 'Secure aggregation' box to a green box on the right labeled 'Parameter server'.

但是 Secure aggregation 只适用于Client数目较多的场景

DATAFUNCON
2020大数据 AI的最新技术实践

阿里安全
ALIBABA SECURITY

- Secure aggregation的局限性
 - 如果参与方过少（例如2个），Secure aggregation并不能保护梯度
 - Client 1拿到新一轮的W，减去自己的梯度就可以推出Client 2的梯度了



The diagram illustrates a limitation of Secure Aggregation with two clients. On the left, two green boxes represent 'Client 1' and 'Client 2'. Arrows labeled ∇W_1 and ∇W_2 point from each client to a central blue box labeled 'Secure aggregation'. A green box on the right is labeled 'Parameter server'. An arrow labeled 'W' points from the 'Parameter server' to the 'Secure aggregation' box. Another arrow labeled 'W' points from the 'Secure aggregation' box to 'Client 1'. A red callout box with a red arrow pointing to the 'W' arrow between the 'Secure aggregation' box and 'Client 1' contains the text: '可以根据W和 ∇W_1 推出 ∇W_2 '. An arrow labeled 'Aggregate(∇W)' points from the 'Secure aggregation' box to the 'Parameter server'.

Google 还有一种方案叫做 secure aggregation，也就是说要通过 secure 的方法把这些多梯度聚合在一起，最后效果就是 Server 只看到了 n 个梯度聚合在一起的结果。但是，不知道某个具体的 client 梯度是多少的，从而导致了 Server 要攻击某个 client 的概率非常的低，但是我们观察到 secure aggregation 只适用于



client 数目比较多的情况。我们可以假设只有两个 client，那么这个 aggregate 的结果就是两个梯度的和，通过第一个 client 可以推出第二个人的梯度，所以参与方至少要三个人以上，而且这些参与方之间还不能够合谋，所以说这是 secure aggregation 的局限性。

2. 联邦学习应用面临的新安全挑战

讲解完横向联邦学习的问题之后，接下来了解下国内引入新的联邦学习应用后，会面临什么样的新的安全挑战。

① 参与方过少带来的问题

DATAFUNCON
2020大数据 AI 的最新技术实践

阿里安全
ALIBABA SECURITY

- 参与方过少（例如两方合作）带来的问题 – 续
- 半同态加密保护参数：只能实现单向保护

例：Alice拥有解密能力

```
graph LR; Alice[Alice] -- "Encrypt(vw)" --> Bob[Bob]; Bob -- "Encrypt(w)" --> Alice; Alice -- "解密" --> W[w];
```

✓ Alice的参数对Bob保密
❗ 但Bob的参数无法对Alice保密

我们经常遇到 cross silo FL 参与方很多情况下都是两个，由于参与方过少会引来新的安全问题。



我们传的梯度是可以用半同态对它进行加密的，例如：Alice 把它的梯度用半同态加密，然后传给 Bob，这样是没问题的。Alice 的参数确实是对 Bob 保密的，但是 Bob 在这个加密的数据上运算完之后他是需要传回给 Alice，Alice 最终需要解密，或者说每一个 round 都需要解密，每一个 round 中 Bob 的参数实际上是被 Alice 知道的。因为参与方只有两个，Alice 得到两个人的计算结果，她肯定是可以从这中间推断出 Bob 的信息的。也就是说，在这种同态加密保护梯度中，只有一方是受益的，另一方他其实没有受益，跟普通的联邦学习是一样的。就是说半同态加密参数只能实现单向的防护。

② 纵向 FL 带来的问题

- 怎样对齐样本？

DATAFUNCON

2020大数据·AI的最新技术实践

- 纵向FL带来的问题-1
 - 为了实现纵向FL，需要首先按id对齐数据
 - 对齐过程是否符合隐私政策？
 - 即使使用PSI（隐私求交）技术，也只能保护“不在交集内的用户身份”，但是在交集内的用户身份**必然泄露**
 - 例：商家A知道了“用户1也在商家B那注册了”
 - 用户1未必同意这个信息被A知晓

	特征1	特征2	特征3	特征4	标签
User 1	Green	Green	Grey	Grey	Grey
User 2	Green	Green	Grey	Grey	Grey
User 3	Green	Green	Grey	Grey	Grey
User 4	Green	Green	Grey	Grey	Grey

商家A持有 商家B持有

纵向的联邦学习又带来了一个新的问题——怎么对齐样本？例如：不安全的方法跟安全的方法，无论怎么对齐，其都是要按照主键对齐的。在对齐之后，不可避免的泄露了一个信息，对齐的用户都是谁？可能没对齐的用户呢？我们是可以用



PSI 这种方法来保护它的。一旦建模，就不可避免的要把这些数据提取出来，也就是说只要在交集里面的那些用户，就会不可避免的泄露了，我们可以再往里面加入假数据等等，但毕竟它在里面就是在里面了。比如说 A 公司跟 B 公司合作，他们之间想进行一个聚合，可能 A 公司的用户并不想把我是 A 的注册用户这个信息告诉 B，也就是说对齐这个东西它的 somehow 是在一个灰色地带，所以严格来说如果要对齐的话，应该用户显式的点击同意，我同意 A 把我的信息授权给 B，所以纵向的样本对齐问题是一个老大难的问题，虽然现在可能大家都在做，但如果监管严格了，这个问题，我们需要一起来想怎么处理。

- 无标签方

DATAFUNCON

2020大数据 AI的最新技术实践



- 纵向FL带来的问题-2
 - 纵向FL必然存在无标签方，而无标签方难以进行特征工程
 - 如何让无标签方进行特征工程又能保护数据隐私？
 - 已经脱离联邦学习的范畴
 - 需要定制化的安全解决方案

	特征1	特征2	特征3	特征4	标签
User 1					
User 2					
User 3					
User 4					

商家A持有

商家B持有

没有标签，怎么做特征工程？

纵向的联邦学习肯定有一个人是无标签方，无标签方他可能需要做特征工程，他不能直接把这个特征直接传给别人或者直接进行联邦学习，那么有些特征工程是需要用到这个标签的，所以它怎么用呢？这也是一个难题。



实际上这个特征工程本身就是一个特定的算法，跟 Google 的横向联邦学习已经没有关系了，我们需要定制一种方案，比如说我们就是要算那个 WOE。那我们就要定制一个方案来安全地算这个 WOE，这也是第二个难题，也就是说纵向的联邦学习带来了很多新的我们以前传统的联邦学习没有遇到过的问题。

DATAFUNCON

2020大数据 AI 的最新技术实践



- 举例：计算 WOE (Weight of Evidence)
 - WOE 定义：某个特征箱体内的 $\ln(\text{反例总占比} / \text{正例总占比})$
 - 若拥有“年龄”一方不拥有标签（样本是正还是负），则难以正常计算 WOE

	正样本数	负样本数	反例总占比	正例总占比	WOE
0-18岁	100	50	10%	10%	0
18-40岁	500	100	50%	20%	0.92
40-60岁	300	150	30%	30%	0
>60岁	100	200	10%	40%	-1.39
总数	1000	500			

上图是 WOE 的例子，WOE 它是要计算这个特征的重要性，比如说我想把年龄分成不同段，比如 0~18 岁等这样几个段，那么每个段段内都存在正样本数与负样本数。那么，这个 WOE 就是把反例总占比比上正例总占比，然后求一个 $\log$ ，这个数越大，说明这个特征这个分段对这个模型越重要，也就是它的判别度越高，我们最后就可以给它加一些分，这个分总可能比较好。但是，对绿色的参与方来说，他是不知道那个标签的（假设标签是在另一方），那他怎么知道这正样本数跟负样本数呢，所以他是没办法知道的。



所以，怎么计算 WOE 也是一个难题，这也是纵向联盟带来的新的难题。

03

安全多方计算解决方案

1. 安全多方计算

DATAFUNCON
2020 大数据 AI 的最新技术实践

阿里安全
ALIBABA SECURITY

- 安全多方计算(Secure Multiparty Computation , MPC)
- 可证明安全
 - 严格的安全定义：除最终的训练结果之外，不泄露任何数据内容
 - Semi-Honest model
 - Malicious model

除最终的计算结果之外，
一切中间结果都是加密状态，永不解密

a
↕
b

什么是安全多方计算？怎么用它来解决这些难题？

安全多方计算是一个密码学的定义，它叫 secure multi party computation MPC，它是可证明安全的，也就是说它有一个严格的安全定义，双方想计算什么东西，除了这个计算的结果之外，中间的任何步骤都是不泄露任何数据内容的。

比如说 a 和 b 想一起算个 $f(a,b)$ ，双方就真的就只知道 $f(a,b)$ ，其他任何东西，都是零泄露的。当然它里面有细分，比如说有 semi honest model 跟 malicious model，这个就是具体技术问题，就不细讲了。



2. 举例子说明安全多方计算到底怎么做？

DATAFUNCON
2020大数据 AI的最新技术实践

阿里安全
ALIBABA SECURITY

- 例：Alice和Bob分别拥有数据a,b，希望联合计算机器学习模型 $F(a,b)$

Alice a

Bob b

比如说 Alice 跟 Bob，他们分别拥有数据 a 和 b，他们想进行一个联合的机器学习 $f(a,b)$ 。这里我们不管它是纵向横向总之它就有一堆数据 a，它有一堆数据 b 就对了。安全多方计算 MPC 有很多种，我们这里是用基于秘密共享的例子，就是说用秘密共享的 MPC 方法怎么做这个建模。



DATAFUNCON
2020大数据 AI的最新技术实践

阿里安全
ALIBABA SECURITY

- Step 1 : 随机拆分

Alice: $a-r, r$

Bob: $b-r', r'$

r, r' 是随机数

DATAFUNCON
2020大数据 AI的最新技术实践

阿里安全
ALIBABA SECURITY

- Step 2 : 交换分量
 - 得到秘密分享状态的a和b
 - 单方视角下都是乱码，只有双方同意的情况下才能复原

Alice: $a-r, b-r'$

Bob: r, r'

r, r' 是随机数



DATAFUNCON
2020大数据 AI的最新技术实践

阿里安全
ALIBABA SECURITY

- Step 2 : 秘密分享状态下进行计算
 - 加法：A和B各自本地将“密文”相加即可得到 $a+b$ 的“加密”版本
 - 其他操作：乘法、比较、除法...
 - $\{+, -, *, \dots\}$ 构成整个机器学习算法

Alice

$a-r+b-r'$

Bob

$r+r'$

首先， a 跟 b 会把他自己的这个数据进行一个随机拆分，比如 a 有一堆数据，生成了一堆随机数， a 减去这个随机数，这个 r 是他本地生成的随机数，同理， Bob 他也会本地生成随机数 r' ，那这个 r 跟 r' 先不告诉对方，那我就把这个数据分成了两份，任意一份单拎出来看好像都是个 nonsense 的 garbage，因为它是随机的嘛，它减去随机的也是个随机的，然后，他们两个人可以交换一下这个分量，比如说 Bob 把这个 $b-r'$ 发给对方， $Alice$ 把这个 r 发给 Bob 。之后，我们称这个数据集现在处于一个秘密共享的状态，也就是说单方视角上他们看到的都是乱码，只有双方同意的情况下，把这两个数据拼到一起，他才能知道最终的数据是什么。那么这个秘密共享状态下的数据集，它的优点就是它还是能够计算的。

我们怎么算 a 加 b ？其就是本地把这两个分量相加。比如 $Alice$ 算出了 a 加 b 减去这两个东西， Bob 就把这两个东西加起来，可以看到这两个东西如果拼在一起的话，它是可以得到 a 加 b 的。同理，我们也可以在秘密共享的状态下做 a 乘 b 、



a 除 b, a greater than b 等等，协议会复杂一点，但是都是能做的。然后这些操作它构成了整个机器学习的算法，比如说我可以在上面算一个 $f(a,b)$ ，然后得到 $f(a,b)$ 的秘密共享状态，我们两个人再商量一下，把这个拼起来，发现了 $f(a,b)$ 是多少，同时中间的任何中间结果都是秘密共享状态的，都是零泄漏的。

3. WOE 为例子，我们怎么来无泄漏的计算这个 WOE 呢？

DATAFUNCON
2020 大数据 AI 的最新技术实践

- 安全多方计算可以无泄露的计算 WOE
 - 秘密共享状态下向量内积计算正负样本数
 - 秘密共享状态下计算除法得到 WOE
 - 除 WOE 之外没有任何信息泄露
- 对比：使用半同态计算 WOE 的方案会泄露每个分箱的样本数目

	正样本数	负样本数	反例总占比	正例总占比	WOE
0-18岁	100	50	10%	10%	0
18-40岁	500	100	50%	20%	0.92
40-60岁	300	150	30%	30%	0
>60岁	100	200	10%	40%	-1.39
总数	1000	500			

A: X (Feature to be binned)

id	set_1
eg	x1, x2, x5...
id	set_2
.....	
id	set_j
.....	
id	set_k

B: Y (Labels)

id1	y1
id2	y2
...	...
idj	yj
...	...
idn	yn

1. $(id, Enc(y), Enc(1-y))$

2. {
 Encode(id, set_i)
 sum(Enc(y)),
 sum(Enc(1-y))
}

3. $npos_j = Decy(sum(Enc(y)))$
 $nneg_j = Decy(sum(Enc(1-y)))$

Compute iv result in B

1. $dispos_j = npos_j / pos_total$, $disneg_j = nneg_j / neg_total$
 2. $Woe_j = 100 * \log(dispos_j / disneg_j)$
 3. $IV = \sum_j (dispos_j - disneg_j) * \log(dispos_j / disneg_j)$

因为 WOE 就是一个正负样本的比值，正负样本我不知道，但是知道标签的那一方可以发一个秘密共享的向量过来。比如，正样本的就是 1，负样本的就是 0，他把这个向量以秘密共享的方式发过来，我自己的这个向量跟这个秘密共享的向量进行一个乘法，得到一个秘密共享的这个结果，这个秘密共享的结果就是这个正样本的数。但是，它是秘密共享状态的，所以我也不知道它是多少。之后，我可以进行一个秘密共享的除法，可以再次进行一个秘密共享的 log。最后，如果我



要是必要的话，我就把这个数据复原出来，比如算出 WOE 是 0.9，然后这个过程中任何数据都是没有泄露的，除了你要计算的那个 WOE 最终的结果。如果我们不用安全多方计算，用其他的自设方法来算 WOE 呢？比如说我们用半同态来算这个 WOE，那边把加密的 0 跟 1 发过来，这样会泄露我每个分箱的样本数目，比如我 0~18 岁有 150 个人，这个数据有样本有标签的一方，不可避免的被他知道了，这个泄漏虽然少，但是中间肯定是有泄漏的。

对于这两个方法，因为我们安全多方计算的除法跟向量内积还是比较高效的，所以这个方法还是比较好的。

4. 安全多方计算不需要“数据对齐”就可以建模

DATAFUNCON

2020大数据 AI 的最新技术实践



阿里安全
ALIBABA SECURITY

- 安全多方计算不需要“对齐数据”就可以建模
 - 秘密共享状态下进行匹配，各机构不泄露自己的客户信息
 - 交集也是秘密共享状态，不泄露交集内的用户身份
- GDPR第5条(b)
 - 对个人数据的处理不当违反最初收集该数据时的初始目的（对齐数据过程是存在风险的）
 - 若为统计用途，则可以超出该初始目的（可以建模）

	特征1	特征2	特征3	特征4	标签							
商家A持有	User 1	XX	XX	0	0	MPC匹配， 谁都不知道匹配上的 是User 2	User 2	特征1	特征2	特征3	特征4	标签
	User 2	XX	XX	0	0			XX	XX	YY	YY	YY
商家B持有	User 2	0	0	YY	YY							
	User 3	0	0	YY	YY							

商家A持有 商家B持有

<https://engineering.fb.com/open-source/private-matching/>

然后来到比较关键的数据对齐方面，虽然有 PSI 的数据对齐，但交集里面的用户身份是不可避免的泄露，不过我们有方法可以在秘密共享的状态下进行匹配。



比如说商家 A 持有用户 1 与用户 2，商家 B 它持有用户 2 与用户 3，然后他们可以把他们所有的数据都以秘密共享的形式分成两份。大家有 4 个秘密共享的数据，谁也不知道哪个是谁，然后在这个秘密共享状态下可以进行匹配，得到一个秘密共享的结果。从 4 行得到了 1 行，但是大家只看见了 4 行变成 1 行，谁也不知道这一行是 user2，最后得到了秘密共享状态下的 user2，然后秘密共享状态的数据是可以进行 MPC 建模的。这样完美的保护了用户的隐私，谁也不知道这是 user2，user2 呢也没有让任何人知道她是 A 的客户还是 B 的客户，那么这样做有什么好处呢？

我们可以下结论说我们这样做是合规的。例如：我们以 GDPR 为例子，其第 5 条规定：对个人数据的处理不应当违反最初收集该数据时的初始目的，意思就是：用户让你干什么你就可以干什么，用户没答应干什么你就不能干什么。严格来说对齐数据的处理这个过程，用户是没有同意商家 A 把我是你的注册用户这个信息告诉商家 B 的，所以，这个过程 somehow 是存在风险的。但是 GDPR 也规定，统计用途是可以超出这个初始目的，很明显建模是一个统计性的。比如，他在这个交集上建出一个模型，这个肯定是一个统计性的模型，也就是说秘密共享状态下的数据对齐是合规的，这是安全多方计算的一个优势。

具体的算法比较密码学，大家可以参考一下 Facebook 最近发的一个 blog，上面的方法就是在秘密共享状态下进行数据对齐，这是安全多方计算解决的第二个数据挑战——怎么对齐数据。

5. 安全多方学习缺点



DATAFUNCON

2020大数据 AI的最新技术实践

阿里安全

ALIBABA SECURITY

- 在LR等模型方面，安全多方计算的性能完全可以满足业务需求
 - 20000样本，100特征，LR建模耗时：秒级~分钟级

NIH

National Institute of General Medical Sciences

PlatON

NIGMS and PlatON Award for 2019 IDASH Genome Privacy & Security Competition

Cheng Hong
Zhicong Huang
Wen-jie Lu

Li Ma
Fuping Qu

Alibaba Gemini Lab, Alibaba Group

in recognition of your contribution to

Trust & Secure Collaborative Privacy of Machine Learning Model

October 24, 2019

Team	Affiliation	Training time cost(s)		Accuracy(%)	
		BC-TCGA	GSE2034	BC-TCGA	GSE2034
Gene X	University of Washington	130.985	1161.29	100	76.087
V for Victory	Alibaba security	37.61 (11.631+25.979)	21.62 (5.814+16.006)	100	70.402
		6.183	6.897	100	66.102
		13.231	16.129	100	61.917
Monse	Art Financial Services Group	69.112	81.713	100	64.407

注：组委会复议时发现 University of Washington 的准确率有误，实际不到68%

安全多方计算有什么缺点呢？它的缺点就是它性能肯定是低于联邦学习的，为什么说？

因为联邦学习中每个 round 总有一部分是可以本地算的，不需要网络，然后算完之后再交互一次。但是安全多方计算，他每一个操作都需要交互，例如：每一个乘法，每一个比较都需要双方的交互，也就是说它的性能可能是比较弱的。但是，目前在 logistic regression 这种简单模型下，它的性能经过我们的优化已经是完全可接受了。比如说万级样本百级特征可以 10 秒钟跑完，我们去年参加了一个 iDASH 的安全多方计算比赛，他的题目是：有三个医院，每个医院是有一些病人的数据，他们规定这个病人的数据是严格不能够传给别的医院的，他们三个医院想合作在这个数据上进行一个建模，也就是说判断某些基因的人可能/不可能得某些病，这样数据越多建模是越准确的。但是，由于合规问题，医院之间不能互传数据，所以比赛要求要使用安全多方计算来实现医院之间的联合建模。



我们是取得了这个比赛的冠军，我们是唯一一个准确率超过 70% 的队伍，我们的性能也非常好，是 20 秒，这也是这个比赛 6 年以来第一次有中国队获得冠军。

总的来说，安全多方计算在某些场景是完全可用的，但是我们不能否认它对复杂模型，比如说 XGBoost 等等，多方安全计算的性能还是存在明显的瓶颈，这方面 Secure Boost 的性能就会比它强很多。

04

总结



DATAFUNCON
2020 大数据 AI 的最新技术实践

阿里安全
ALIBABA SECURITY

- 国内流行的 FL 方案与 Google 的经典 FL 方案存在很大不同
 - 技术角度：国内 FL 多为纵向，Google FL 为横向；
国内 FL 多面向少量机构的合作，Google FL 面向海量终端的合作
 - 风险角度：国内 FL 面临的安全挑战更多
- 在设计 FL 解决方案时，安全性上必须慎之又慎
 - 必须详细说明对各参与方提供了何种安全保障，存在何种信息泄露，能抵抗何种攻击
- 安全多方计算（MPC）解决方案有其独到的安全优势
 - 有广阔的应用前景

我们国内在流行的联邦学习方案与 Google 最初发明的联邦学习方案之间存在很大的不同：



技术角度来讲，国内的联邦学习主要类似于 cross silo 的这种联邦学习，它是纵向的，Google 的联邦学习主要是横向的，国内联邦学习主要是面向少量的两个或三个机构的合作，Google 联邦学习面向海量的安卓终端之间的合作。

这个变化肯定带来了新的安全挑战，当我们要面对这些安全挑战时，我们设计上必须慎之又慎，我们要设计完善的解决方案。如果我们为了性能进行了妥协，这是无可厚非的，但是我们一定要详细的对参与方说明，我们这个算法解决方案是解决了什么样的安全问题？提供了何种安全保障？比如说对谁可能会产生何种信息泄露，能抵抗何种攻击，让用户有这样一种知情权。

安全多方计算解决方案，它是有其独到的安全优势，因为其中间没有结果泄露，那么它是有更广阔的应用前景。我们也很高兴看到 FATE 他也现在已经提出了 SPDZ 的安全多方计算 protocol，我们相信它可以进行更好的后面的迭代开发。



DataFunTalk 成就百万数据科学家！

这是我们阿里安全双子座实验室，这个平台的 logo 叫可用不可见，就是 BLINDFOLDED COMPUTING，闭着眼睛计算，并看不到数据的内容。

今天的分享就到这里，谢谢大家。



58 同城风控平台演进



分享嘉宾：张月 58 同城 风控架构师

编辑整理：黄乐平

出品平台：DataFunTalk、AI 启蒙者

导读：58 同城的风控业务从 2010 年开始搭建平台，到 2012 年正式成立信息质量部，经历过了 10 年的发展，三代系统变革，目前已经是全集团最重要的中台之一，肩负了每天百亿级别数据的风险控制。本次分享将主要解析在不同业态、不同年代下，风控治理的不同思路，进而打造出不同平台的过程。

01

58 做风控的起因

59

峰会回看，扫码领取专属优惠

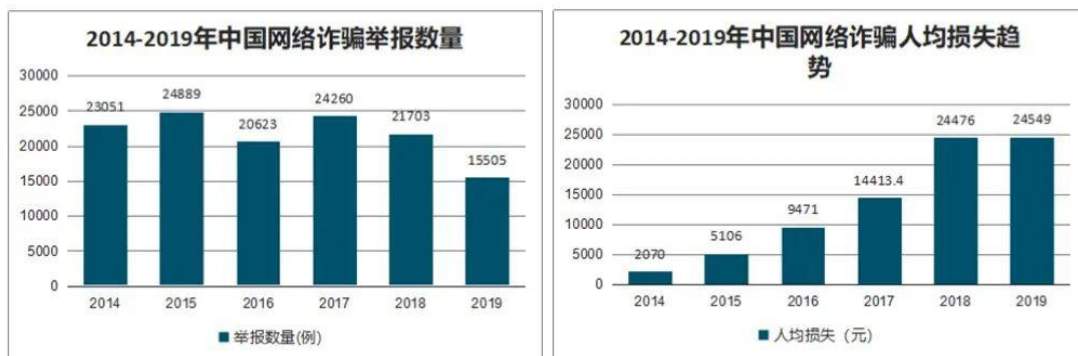


一个分类信息平台，其主要是以内容和流量为核心，包括上下游两大类用户：B 端用户（供应端）——生产内容以及 C 端（消费端）用户——使用内容。在信息的产生和消费两端，都会面临许多问题，诸如诈骗、羊毛党、刷单、刻章、办证、发票、微信吸粉、广告、水贴等一系列非正常流量行为。

面临的问题：

- 诈骗是最为严重的一种行为，是红线以下，对个体伤害巨大。从 2014-2019 年中国网络诈骗举报数量以及中国网络诈骗人均损失趋势两个统计表可以看到，随着国家的管控力度逐步增大，诈骗数量也呈现逐年下降的趋势，但对于个体伤害而言，却是逐年增加的。我们能从中看出，一方面人们的收入水平逐年上升，另一方面，诈骗的手段逐渐更新迭代，隐蔽性增强，更多高知分子也逐步被卷入骗局。

58 Why



- 羊毛党主要指的是借助各类营销活动薅羊毛，诸如之前的某多多被曝出现重大 Bug，用户可领 100 元无门槛券，吸引了众多羊毛党“薅羊毛”，网传“一夜损失了 200 亿”；更有甚者，羊毛党自发组织形成团体，可以搜到很多以“羊毛”为关键词的 QQ 群，仅在 2019 年，58 利用技术手段挽回的各类羊毛党损失就超过千万级别。

58 Why



- 刷单是电商平台面临的头疼难题，这里会分为两类情况，一种是为各电商平台商家刷量，赚取收益，另一类则回归到诈骗渠道，诈骗贪小便宜的各种“兼职”人员。



58 Why

诈骗
红线以下，个体伤害巨大

羊毛党
借助各类营销活动薅羊毛

刷单
电商平台的头等难题



- 刻章、办证、发票为平台的灰色地带，这其中良莠不齐，正规的诸如招聘类目下的财务人员会有开具发票的情形，但类似代开发票这种行为已经触碰红线了。
- 微信吸粉是各类恶意行为中套路最丰富的，诸如房产类目会存在假冒房东发假房源信息，吸引房客咨询，实际为了吸粉的营销活动。
- 广告水贴是各类 UGC 平台面临的普遍问题，行为上也和正常用户趋紧，治理难度较高。

02

58 风控发展历程

58 风控发展历程主要包括 4 个阶段：



58 历史



- 第 1 阶段：雏形阶段——微量业务，封闭系统
- 第 2 阶段：发展阶段——少量业务，增强运营
- 第 3 阶段：中台转变——能力复用，业务自治
- 第 4 阶段：生态建设——服务思维

第一阶段：微量机审+人工审核



58 手撕XX阶段——微量机审+人工审核



第一阶段平台业务较少，主要为 58 信息、企业发布以及简历发布，采用人工审核配合部分机器审核的手段。此阶段主要规范了系统的边际，包括 UGC 产生后的一系列流程操作。经过在线检测后，分成了两个分支，其一经过离线检测，其二经过人工审核，并行操作，两者的结果最终通过判断执行模块进行审核执行操作。

为了加强对外部情势的感知，主要主动采用“卧底”黑产群，通过购买各类黑产软件，探索平台漏洞，对系统进行更新迭代。

暴露问题：

该阶段对于系统中的“变”和“不变”部分缺乏抽象，主要是采用硬编码的方式，这使得开发频次及成本较高，策略上线周期上，开发速度远远无法跟上黑产破解的速度。

第二阶段：配置化机审+人工审核



58 小米加步枪——配置化机审+人工审核



第二阶段主要是在前一阶段的基础上，明确了策略层的运营能力，以特征为基础，规则和条例为抓手，同时也增加了部分算法的能力，诸如图像算法、文本算法以及行为聚类等。在此基础上形成了三大模块，包括简化特征开发的平台用于风险发现、可运营的策略管理用于风险评估以及集中化的风险处理。

暴露问题：

- 并购：58 和赶集风控系统同时存在，人员重叠，维护代价高；
- 业务增长：风控场景快速增加，技术和运营成本的不断增加；
- 新思想：友商风控平台的快速崛起，学习借鉴友商思路；
- 中台：部门定位的变化，逐步考虑中台化。

第三阶段：机审人审融合、分场景治理



58 多兵种综合作战——机审人审融合、分场景治理



第三阶段目前运行至今，主要包括三方面，包括

- 自助式开发：改变原仅有风控部门独立开发的模式，提供业务侧参与风控工具开发的能力，同时能够进行整个开发后的测试、部署、上线工作。
- 一站式运营：提供了一整套的运营方案，运营人员可以快速完成业务注册。同时积极提供自助接入场景的能力，在新场景接入时提供基础的通用能力，助力业务快速落地构建防御体系；同时，也能够提供一些专家经验，输出前期总结经验，避免反复造轮子。
- 业务隔离：改变了前期多个业务共用一套风控模型的方案，第三阶段实现了业务间进程级别的隔离以及微服务的拆分，避免业务间问题的相互影响；同时增加了熔断降级等各类服务治理能力，避免上下游数据服务造成的影响。



第三阶段整体实现了毫秒级的响应速度、目前已支撑千级别业务场景、全场景内容服务覆盖、上万节点集群规模、百亿量级离线分析能力以及全自助平台管理。

第四阶段：专家指挥、智能审核

第四阶段为正在实现中以及对未来的规划方面，包括：

- 隔离方面，目前业务间的隔离还没有做到极致，后期会对数据库、人工审核、配置中心等进行隔离；
- 以算法为中心，逐步提升系统自动运营水平。

整体而言，风控的能力提升有很大的空间，这对技术储备要求很高，需要多职能跨部门、跨组织合作，从而实现高并发、高可用需求下的快速风控能力。当然，风控其实是一个不断对抗的过程，版本的迭代需要足够快，以交付为目标，避免过度设计导致长时间的对抗真空期。

03

后续答疑

① 能否举一个 58 同城黑产攻防案例？

答：常见的攻防案例一般是敌方攻击、我方防守，对抗的过程。

当时有遇到一个案例，房地产业务因为流量大，关注度很高，有利用工具批量发布信息的行为。开始阶段能够从中发现，其发布内容行为与正常人发布有一定的偏差，常见的手段就是通过模拟批量 Post 请求提交数据，但这种情况下上报的数据



中缺乏正常用户的行为数据，很容易可以识别到为非真人发布而进行拦截。后续，对方也会逐渐升级手段，寻找可突破点，诸如采用按键精灵，通过浏览器进行提交，用模拟点击的方式。我们解决方案则是通过寻找相似点以及用户行为时间序列来发现非真人操作，之后采用动作随机化，这时从单一用户的角度已经很难着手，我们开始从单一用户转向群体用户行为的分析，挖掘某时间段内存在大量异常行为的账户，通过诸如验证码、认证等手段增加对抗的成本，黑产也在升级过程中采用包括打码平台等方式提高刷帖成功率，我们则增加验证的方式，用更复杂的人机交互等，可以看出这过程是不断对抗，提升作恶成本的过程。

② 详细介绍下当前风控架构？

答：最上层为业务层，包括 58 信息、企业发布、简历发布、同镇信息、交友直播、微聊信息、房产三网、部落视频等上千级别业务；针对每个业务集合，我们会提供一站式运营平台，包括完整的工具集、处理集、人工审核布局以及运营能力服务；在此基础上，针对多业务间并存的服务能力进行抽象，整合出包括基础工具、风险处理集合、数据增广、行为聚类、文本算法、图像算法等通用化服务能力。

③ 风控团队如何对接业务侧，如何说服业务侧接入，投入产出比如何计算等？

答：早期所有的风控均处于风控部门，既承担裁判员的职责，又做运动员的工作，即做风控并评判风控收益，会存在与业务侧产生冲突矛盾的地方。这其中包括业务侧不认可风控的输出能力，认为引入后为带来大量的误杀，影响业务发展等各类问题。后期通过中台能力的搭建以及与业务部门责任共担，实现了双方合作共赢。投入产出比可以通过对线上数据的巡查，以及数据部门的分析，因其中包含



DataFunTalk 成就百万数据科学家！

了部分业务因素，我们也会采用三方数据、申诉反馈等综合收集信息客观反映风控能力。

今天的分享就到这里，谢谢大家。



度小满金融大数据风控模型实践



分享嘉宾：严澄 度小满金融 风控模型负责人
编辑整理：黄乐平
出品平台：DataFunTalk

导读：金融是 AI 赋能传统行业的重要赛道。本次分享的主要内容为金融大数据风控模型在度小满金融的实践。主要介绍金融大数据风控模型的主要技术方法与在应用层面的主要问题，并结合新冠疫情背景下，探讨下风控模型的发展。

01

金融大数据风控模型的技术方法

1. 风险管理中的金融科技



风险管理中的金融科技



风险管理中的金融科技主要包括两个方面：

一方面是传统金融的风险管理，包括：

- A 卡 (Application Scorecard , 申请评分卡)
- B 卡 (Behavior Scorecard , 行为评分卡)
- C 卡 (Collection Scorecard , 催收评分卡)

三张评分卡，是传统金融科技中的一部分。

另一方面是信息技术，包括：

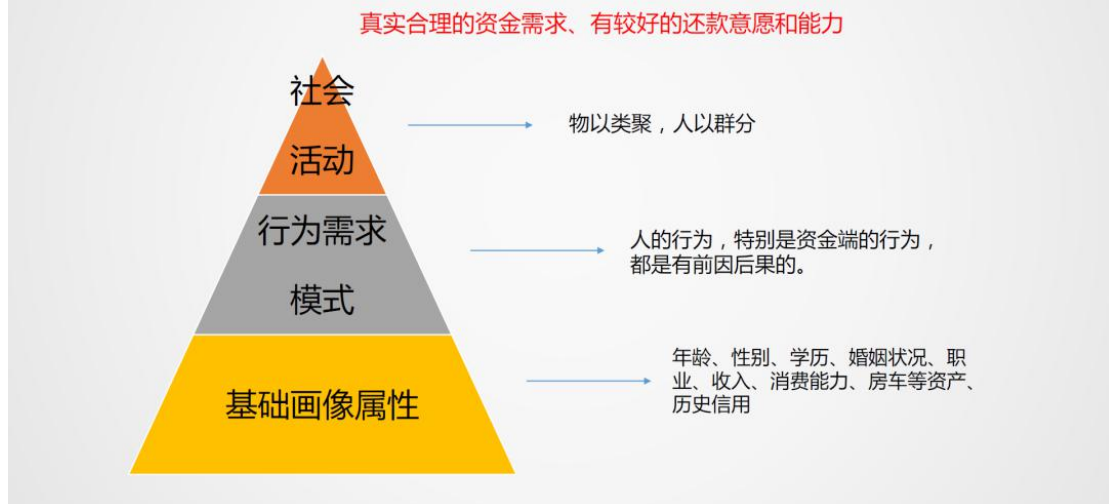
- 算法算力的大幅度提升：A (Artificial Intelligence , 人工智能)
- 用户的行为数据数字化的存储和挖掘基础：B (Big Data , 大数据)
- 强大的资源服务共享能力：C (Cloud , 云服务)

利用信息技术能力能够有效提升传统风险管理评分卡建模的效果。

2. 度小满信贷风险



信用风险



在度小满信贷业务发展过程中，积累了大量数据和模型相关的实战经验。下面主要介绍关于信贷风险模型在度小满的实战经验。如何去识别信用风险，其核心关键点在于识别借款用户真实合理的资金需求以及评估用户是否拥有较好的还款意愿和能力，主要包括三个方面：

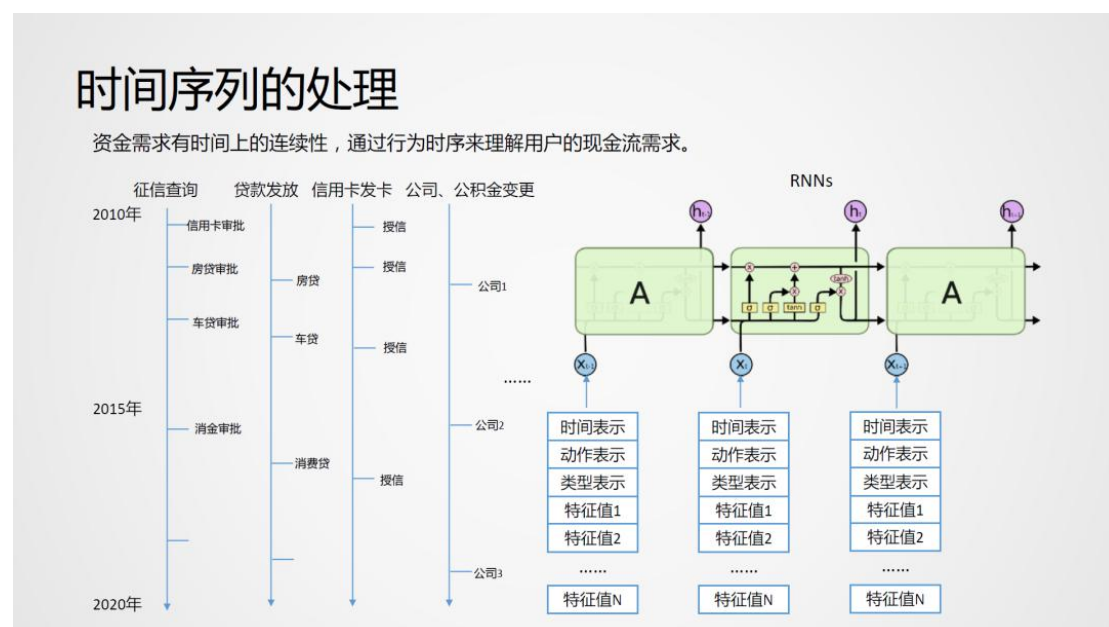
- 底层需要的是用户基础画像信息，主要包括用户的年龄、性别、学历、婚姻状况、职业、收入、消费能力、房车等资产以及相应的历史信用信息，金融相比于电商等领域对基础画像的准确度等更为严格，因其涉及了用户真实的还款能力。
- 中层为用户的基础行为需求模式，主要是用户当前资金端行为往往与前一段时间内的行为存在较大的相关性，通过这些行为可以预测用户的真实资金需求以及未来的还款表现。



- 顶层为用户的社会活动，即“物以类聚，人以群分”，通过关联网络中的聚集性发现欺诈团伙，同时可以发现与还款意愿差关系紧密的人逾期的风险更高，而与还款行为好的用户关系密切的人逾期风险相对较低。

核心而言，利用用户的基础画像、行为需求，外延到用户社会活动去挖掘用户是否有真实合理的资金需求，是否有较好的还款意愿和还款能力，从而建立区分度良好的风险模型。

3. 时间序列的处理：贷前



信贷业务通过用户授权获取征信报告，基于征信报告了解用户的信用历史，通过分析用户的行为时序来理解用户的真实现金流需求。

- 常见的征信查询包括信用卡审批、贷款审批等，此类查询表征了用户在该时刻的资金需求，通过征信报告中贷款发放情况匹配贷款查询申请时间，可以分析用户资金的用信行为。传统金融行业常利用诸如基于不同时间滑动窗口的加工逻辑方式去进行统计，包括过去一个月三个月、六个月、十

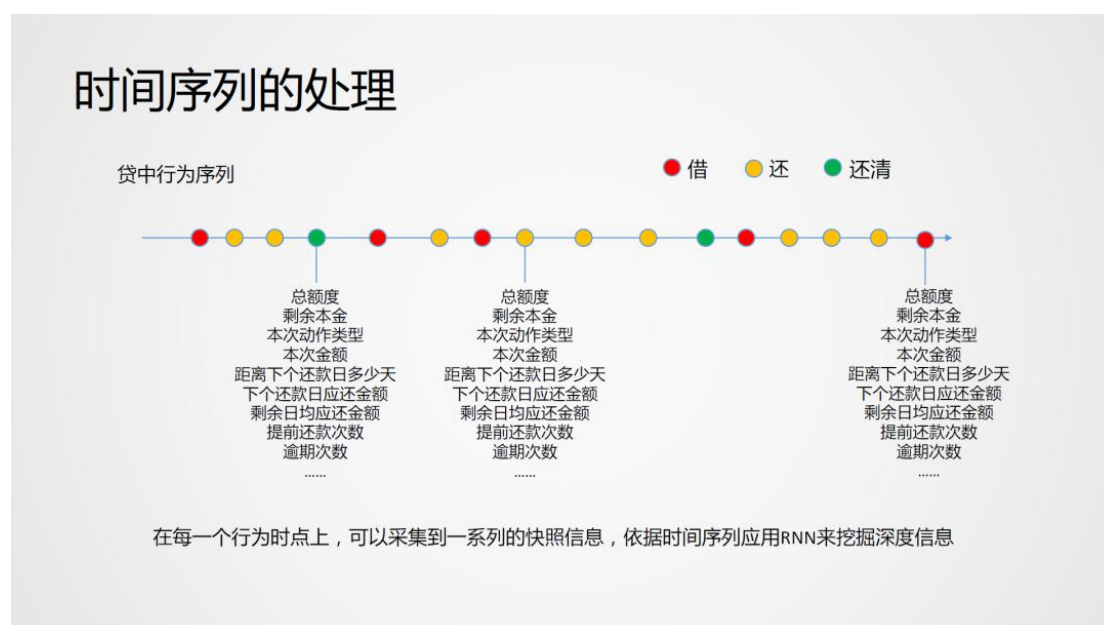


二个月，二十四个月征信报告查询次数等、过去一个月、三个月贷款发放笔数等指标。

- 征信报告还包括用户的公司地址变更、公积金变更等信息，将上述信息基于时间轴进行划分，就可以对用户在一段时间内的信贷需求和用信情况进行刻画分析。我们利用深度神经网络去进行分析，通过记录时间点、该时间点的动作、该动作的类型以及该动作的多个特征值，将每一项信息转为 Item 作为输入提供给 LSTM 单元，获取该单元的输出同时又作为下一个时间序列的输入，不断学习从而获得更好的信息表达。

相比传统方案，基于这类机器学习的方案能够带来 KS 2 个点的提升。

4. 时间序列的处理：贷中

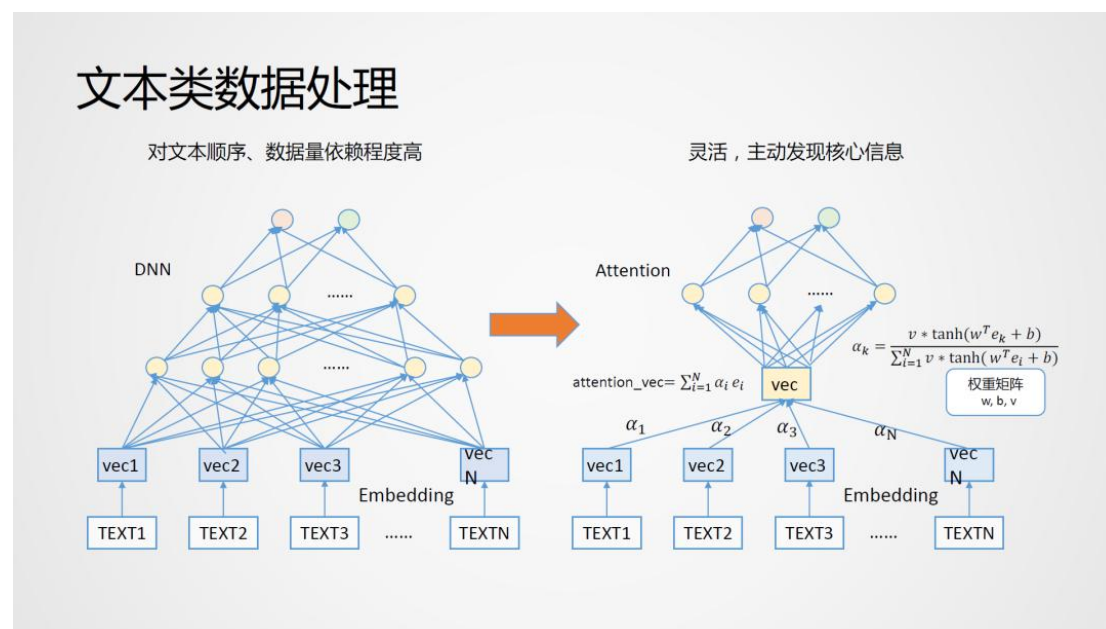


信贷贷中行为数据主要用于 B 卡的建模，在整个客户生命周期线上，用户存在不断借钱、还钱的循环往复行为。基于此，在每个动作发生的时间切片上，可以生成诸如总额度、剩余本金、本次动作类型、本次金额、距离下个还款日多少天、



下个还款日应还金额、剩余日均应还金额、提前还款次数、逾期次数等特征，将这些特征组织成 item，通过各类 RNN 网络能够显著提升 B 卡的能力。

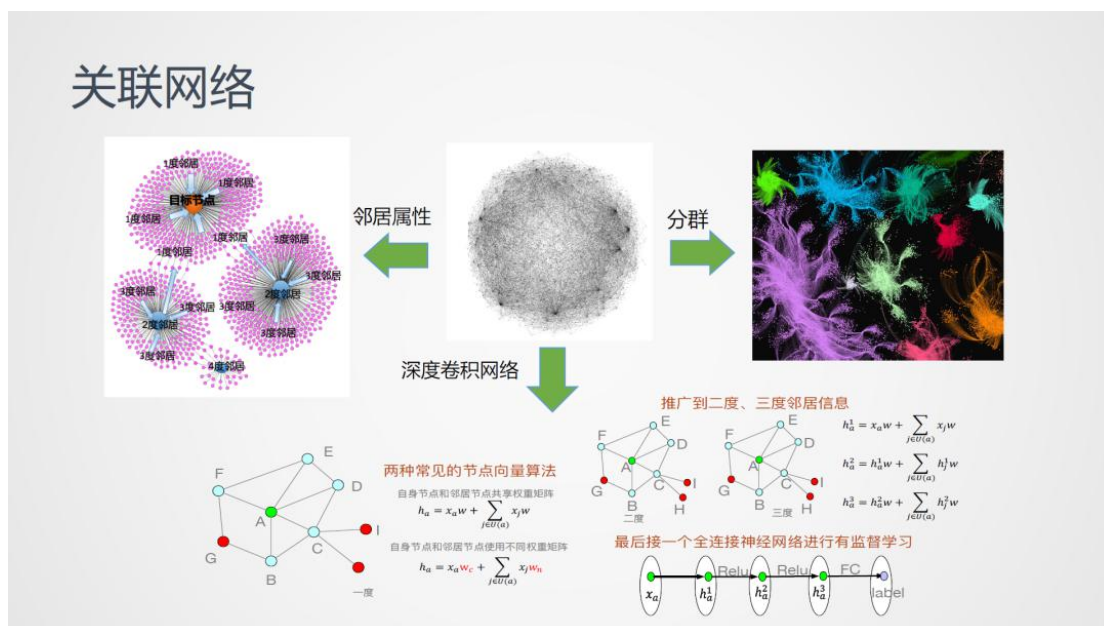
5. 文本类数据处理



互联网行为数据除了结构化数据外，还存在许多非结构化数据，诸如文本类数据。传统金融的很多数据都是数值类的数据，类似收入水平等变量，这些数据一般具有置信度高、分辨率强的特点，而互联网数据则有数据规模大，数据杂乱，且分辨率较弱的特征。传统文本处理的方式一般对文本特征进行排序，构建全连接层深度网络 DNN，这种处理方式强依赖于文本的顺序，会存在稀疏性、泛化能力弱等缺陷。我们参照 Attention 机制的核心思想，关注每个信息单元在整体中的重要性 α ，由于 α 是基于信息单元集合实时计算得到，依赖于 Input 但与 Input 的顺序无关，所以特别适用于那些原本就没有顺序依赖关系的文本内容集合。基于此框架，该方案无需关注文本特征的顺序，这对于整个模型的特征处理逻辑上而言是非常友好的，通过不断提供新的数据，能够让模型网络更加灵活高效。



6. 关联网络



图在金融科技侧的应用十分广泛。对于图在度小满的落地应用，和传统构建图的方式相同，通过构建密集网络主要包括三个方面的输出：

- 为构建该网络的邻居属性，包括一度、二度、三度网络，这些网络都是可见的；
- 为对图网络的分群，利用节点之间的紧密型基于网络的分类去计算群内的相关参数特征信息；
- 为对图进行深度卷积神经网络，利用一度用户特征结合用户自身特征进行卷积，然后推广到二度、三度邻居信息，在此基础上构建全连接层进行有监督学习。

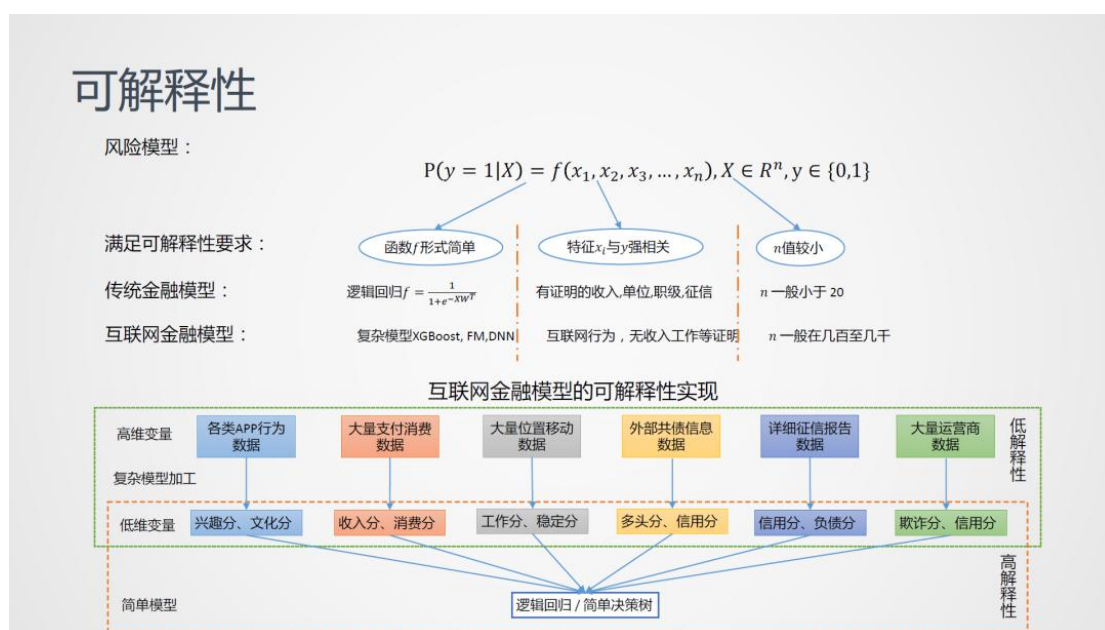
单一应用方向对识别信用风险可能较弱，但通过三种方式的组合，能够显著提升模型的风险识别能力。



02

金融大数据风控模型应用层面的问题

1. 模型可解释性



信用风险的核心即构建一个二分类的模型，而可解释性是策略同学应用模型的一个重要诉求。一个可解释的模型主要有以下三个方面的特点：

- 该模型的函数表达简单，一般为逻辑回归；
- 输出特征 X 与预测 Y 变量强相关，符合普遍认知；
- 核心模型变量不需要特别多，一般不超过 20 个；



而常见的互联网金融模型方案为 XGBoost ,它的特点是模型非线性 ,特征规模大 ,解释性差。为了实现模型的高可解释性 ,我们常用的方案是将底层的高维变量按其来源进行分类 ,先利用复杂模型进行加工形成子模型特征 ,主要有以下几个方面 :

- 利用用户的各类 APP 行为数据输出兴趣分、文化分等 ;
- 利用用户的大量支付消费数据输出收入分、消费分等 ;
- 利用用户的大量位置移动数据输出工作分、稳定分等 ;
- 利用用户的外部共债信息数据输出多头分、信用分等 ;
- 利用用户的详细征信报告数据输出信用分、负债分等 ;
- 利用用户的大量运营商数据输出欺诈分、信用分等 ;

基于上述多个子模型评分 ,再利用逻辑回归或简单决策树构建最终的模型。由于每个低维变量评分和人的认知接近 ,且与风险有很好的线性相关性 ,整个模型就有了很好的解释性。同时 ,对模型质量的监控也变得更加容易了。当模型出现问题时 ,可以很容易定位到可能出现问题的子模型分 ,然后再依据子模型评分去寻找对应的底层数据的异常。

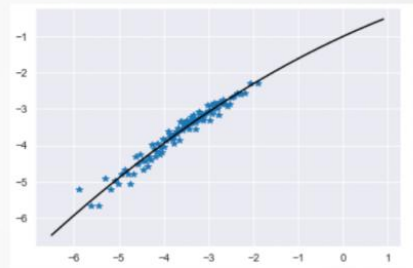
2. 概率标准化



概率标准化

不同概率水平的样本进行训练(模型迭代更新)，得到的模型预测均值概率差异较大；需要修正对齐真实不良率。

1. 按预测值排序分段（如100段）
2. 对每段的真实逾期率取logit
3. 对每段内的所有预测值取logit后求平均
4. 尝试一次或二次曲线拟合



信用评分是模型和策略、外部机构交互的“标准语言”，参照主流的转换方式

$$\text{信用评分 } S = 400 - 35 / \ln 2 * \ln \frac{p}{1-p}$$

经过这样标准化以后，模型的升级就对上游透明了

不同概率水平的样本进行训练，得到的模型预测均值差异较大，这对策略应用的同学带来很大挑战。基于此，我们需要对不同的模型修正对齐真实的不良率，具体步骤如下：

- 按预测值排序分段，如 100 段；
- 对每段的真实逾期率取 logit；
- 对每段内的所有预测值取 logit 后求平均；
- 尝试一次或二次曲线拟合该散点图；
- 参考诸如 FICO 的评分方式进行转化信用评分。

经过上述标准化后，模型与样本数据集或样本不良率无关，模型的升级就对策略、业务同事更加透明。

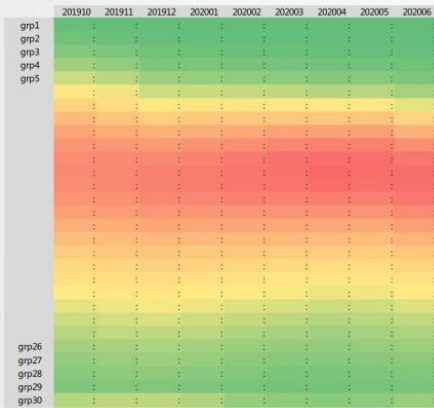
3. 评分稳定性

评分稳定性包括分布稳定性、性能稳定性以及微观层面的稳定性三方面：



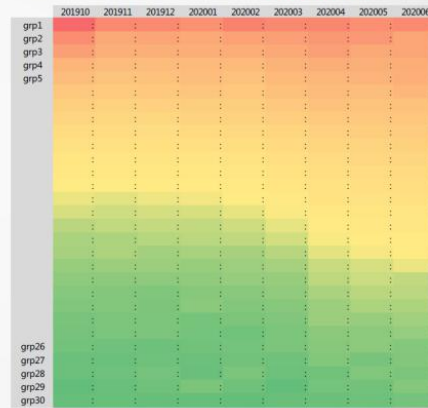
稳定性

分布稳定性：各个分数段的人数占比各月稳定



注：色阶图每个格子是当月分组人数的占比值

性能稳定性：各个分数段对应的人数不良率各月稳定



注：色阶图每个格子是当月分组真实的人数不良率

- 分布稳定性：指的是将用户评分分段排序后按月进行统计，符合正常的正态分布情况，各个分数段的人数占比各月较为稳定，即整个客群的质量未发生较为明显的变化；
- 性能稳定性：指的是将用户不良率分段排序后按月进行统计，能够满足各个分数段对应的人数不良率稳定；

稳定性

微观层面的稳定性：



模型对用户短期的行为反应过于灵敏，虽然整体客群排序稳定，但对于具体用户而言，评分波动过大，策略难以管理

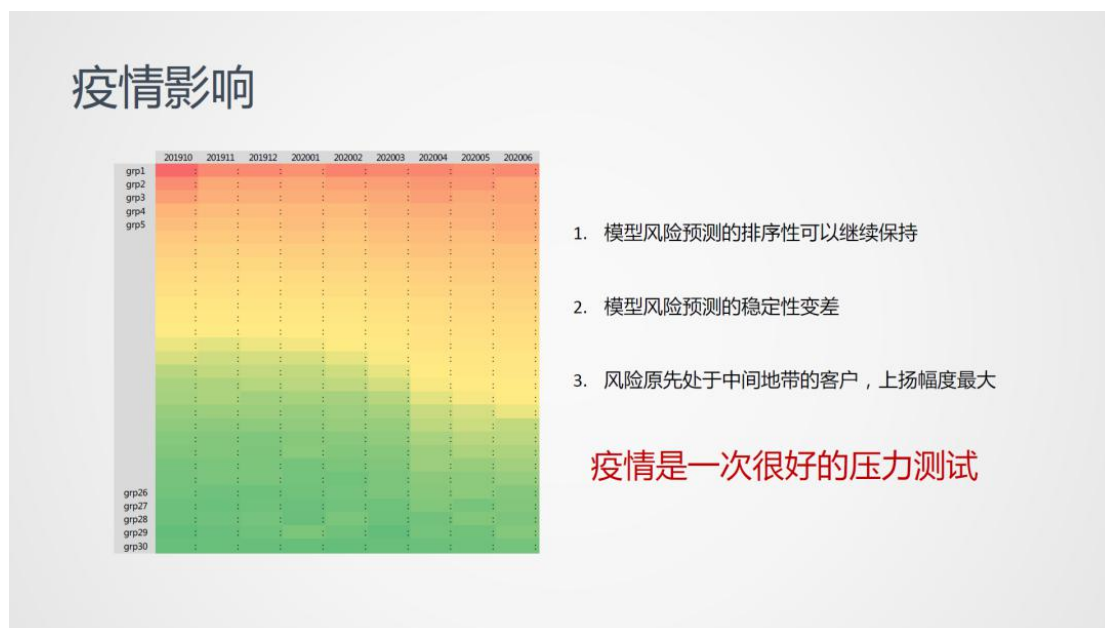


- 单一用户评分的稳定性：若用户存在新的借款、放款行为，则未来贷款风险会提高；若用户还清所有贷款，余额为 0，则未来一段时间内风险会降低。模型对于用户短期行为反应较为灵敏，虽然整体客群排序稳定，但对于单一具体用户而言，各月之间的评分波动较大，策略就会难以管理。从策略层面上而言，模型不论是整体还是局部方面，都希望其稳定性较高，以便于对用户的评价相对稳定，这样才能更好地进行客户管理。

03

新冠疫情背景下，风控模型的一些探讨

1. 疫情影响



新冠疫情是对业务、策略、模型一次很好的压力测试；以前面的用于评估稳定性的图示来看，主要表现在受疫情影响的多个时间窗口下绿色、黄色下沉，该图示包含了以下三个方面的信息：

- 模型风险预测的排序性可以继续保持（色阶图的颜色块顺序依然保持）
- 模型风险预测的稳定性变差（同样高度位置的颜色不一致了）
- 风险原先处于中间地带的客户，不良率上扬幅度最大。（中间地带的绿色变成了黄色（黄色的风险值>绿色））

疫情归根到底其实是对客户、风险模型、策略的一场考验。

2. 疫情对模型的影响和挑战

疫情对模型的影响和挑战

$$p = P(y = 1|X) = f(x_1, x_2, x_3, \dots, x_n)$$

$p \uparrow$ $X \rightarrow$

多头类变量，疫情前后相同X值对应的风险水平显著上升。

1. 宏观环境的影响是否有特征变量可以表达？
2. 未来模型训练的样本中，是否需要包含疫情下表现的样本？
3. 当前模型应用中，如何基于疫情的发展进行修正？

疫情并不会改变用户的特征 X ，但在疫情下同样 X 对应的 Y ，即风险水平则会显著上升。在实际业务场景中，我们观察到诸如多头类变量最为明显。有一些思考和探索供参考：



- 宏观环境的影响是否有特征变量可以表达？疫情必然对用户的收入会造成一定影响，那么是否有很灵敏的变量可以反映这部分信息？比如行职业、复工复产数据对收入进行修正？
- 未来模型训练的样本中，是否需要包含疫情下表现的样本？当前训练的样本为非疫情下的样本，随着时间推移样本的更新，模型就面临是否需要包含疫情期下表现的样本。，这块需要评估下疫情对整个金融环境的影响是否可逆，若疫情为短期，可以考虑直接跳过这些样本数据；若疫情为长期，则需要将疫情作为常态进入模型的训练中了；
- 当前模型应用中，策略如何基于疫情的发展进行修正？这需要模型和策略基于疫情发展的长期或短期性在实践中进行探索迭代。

04

问答环节

1. 什么特征可以反映疫情下的宏观经济情况？

答：这是一个极为有挑战性的问题，在我们实际业务应用发展中，复工指数对整个资产模型的贡献度较高，其实际与用户的收入存在较为强的联系。从隐私的角度很难拿到用户的收入情况，在实际应用中，我们通过位置迁移来推测用户的复工状态。

2. 对比高维变量入模效果与分开计算模型效果？

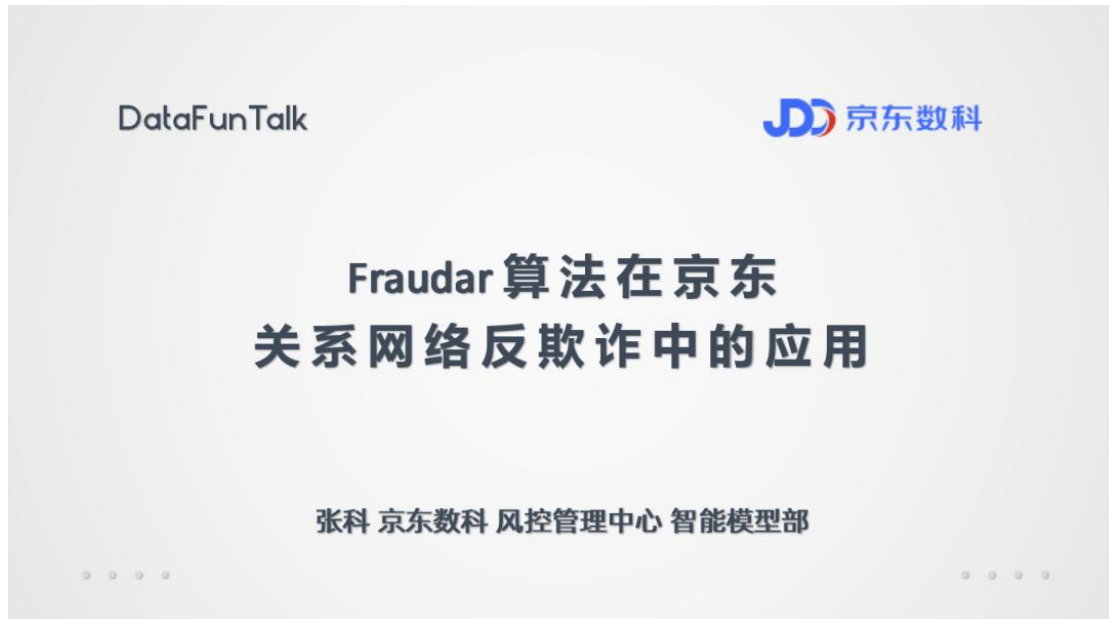


答：高维变量入模和分开入模 KS 偏差幅度在 0.5%左右，并不能说高维变量直接入模就一定好于分开入模，不同模型有 ± 0.5 的不同表现。从整体的角度来看，我们认为结果差距并不是特别大。高维变量入模缺点在于参数较多，对数据质量监控有一定的难度，发现问题较难且可解释性较差。

今天的分享就到这里，谢谢大家。



Fraudar 算法在京东关系网络反欺诈中的应用



文章作者：张科 京东数科
内容来源：京东数科风险算法与技术

导读：一年两度的 618 和双十一，是电商和剁手党在仲夏和金秋的两场盛宴，也是刷单党的两场狂欢。竞争与压力之下，刷单成为了新入商户和低信用商户在官方选择之外速度最快的店铺成长方式。它们按订单价格及数量向“刷单军团”支付刷单佣金来提升虚假销量和好评，从而获取更好的信誉度和搜索排名。对于时间充裕的社会群体如学生党和宝妈而言，刷单也正好可以成为其动动手指就能月入千元的低风险副业。可以说电商的发展，滋生并带火了一个由出资店铺、刷单中介、各级代理、刷手、空包物流组成的刷单产业。但其繁荣的背后，本质仍是恶意炒作信用和获取利益的不正当竞争行为，这一方面侵害了其他商家的交易公平，导致了“刷单找死”，不刷单等死’的劣币驱良币行业现状，另一方面也会



误导消费者对商品和店铺的选择。为了行业的长远发展，电商平台反刷单欺诈，势在必行。



对于刷单欺诈，平台可以从订单、商品、店铺、用户、设备、物流等多个维度进行分析识别。但此类分析的弊端在于随着欺诈手段的不断变化，如刷单行业从机刷向人刷的升级、浏览下单方式的改变，都会导致已有的经验指标和模型特征的失效。这使电商平台在反欺诈的对抗过程中，总存在滞后风险。如果我们以一个更高的维度抛开细节来看问题，或许可以得到一个更好、更稳定的解决方案。刷单场景里，变的是欺诈手段和交易细节，不变的是交易结果。众多的刷手和出资店铺在一段时间内的交易关系形成了一张网络，这是电商平台在整体交易网络中



的一个子网。如果我们可以有效的把欺诈子网从整体网络中识别出来，问题就得到了定位和解决。



图 1 复杂关系网络

01

关系网络与二部图

不同于对个体自身特征的分析，网络提供了对多个对象的关系之间另一种看问题的视角。把对象看做结点，把交互看成边，对象间的发生的各种关联自然会构成一张关系网络。从图论的角度出发，根据结点属性的不同可以把网络分为同构图和异构图。同构图是由同一种结点组成的关系网络，如家庭亲属关系、社交好友关系、论文间的引用关系等。历史上对于同构图的网络表示有很多研究，早在十九世纪就形成了几何拓扑学这一数学分支。在现代的同构关系研究中也逐步提出了基于网页链接的谷歌 PageRank 网页评级、基于结点关系紧密度的 Louvian 社



区发现等重要算法。不过异构图在生活中的表现更为广泛，异构图是由不同属性的结点组成的关系网络，如由买方卖方以及中介组成的交易网络、由大 V 和其关注者组成的关注网络、由手机号\地址\证件号\IP 地址组成的复杂关系网络等。二部图（也叫二分图）是异构网络的一种，它由两类结点组成，并且同类结点之间通常没有关联。前述的刷单欺诈，即是以出资店铺和刷手这两类结点组成的交易关系二部图。

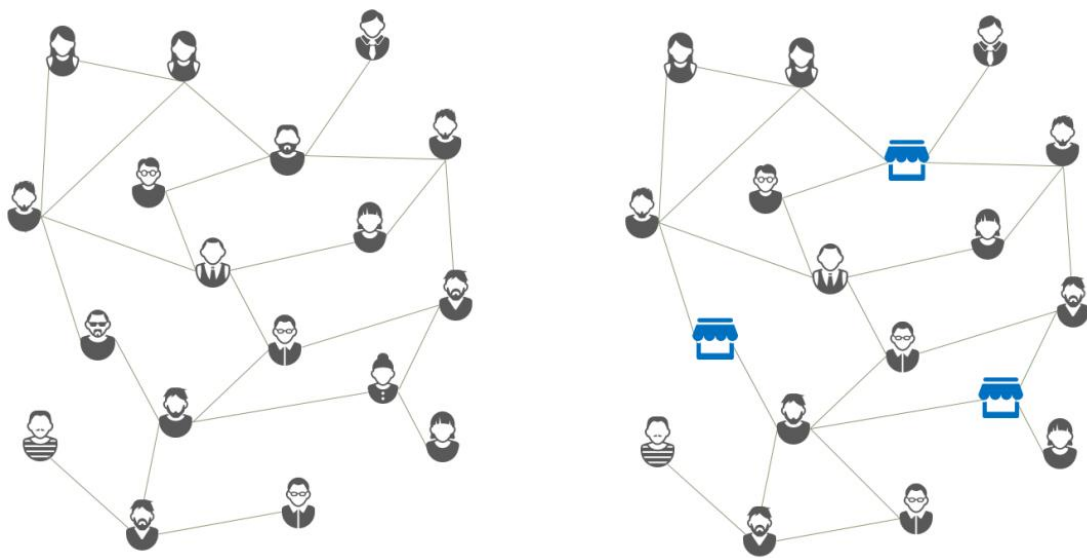


图 2 同构图（左）与异构图（右）

02

二部图下的欺诈

两类结点组成的欺诈场景还可以举出很多例子，如电商场景下用户对商户的薅羊毛、刷好评，如社交场景下水军账号的虚假关注、转发，又如消金场景下用户与商户勾结对平台的消费贷套现欺诈。这些行为都会使两类结点之间出现异常的连



接分布，从整体网络看来其呈现出了一张致密的双边连接子图，且该子图内的结点与图外结点联系相对较少。我们把这种大量的、同步的非正常关联行为模式称之为 Lockstep，即在本不应出现聚集行为的二部图自然关系网络中，出现了双边聚集性行为。

只要能把欺诈行为总结成一种模式，自然可以将其分离出来。但是欺诈者往往会对自已做出某种伪装以使自己看起来有向好的一面，意图绕过风控系统。如在刷单欺诈场景下，为了尽量贴近真实用户的购买习惯，刷单平台会对刷手提出一系列要求，比如要求货比三家、要求最低浏览时长、要求滚动浏览高度及停留时间、要求对于正常热销商品做一定购买等，这些行为都会导致风控经验指标和模型特征的部分失效。在二部图交易网络中，对于正常热销商品的购买体现为刷手为自已增加了一些优异的边连接，使自己看起来更像一个正常的消费者结点。我们需要一种能从这种复杂关系网络中对抗伪装、抽丝剥茧的提取出异常致密子图的算法。接下来对症下药引入 Fraudar。Fraudar 算法来源于 2016 年的 KDD 会议，并获得了当年的最佳论文奖。



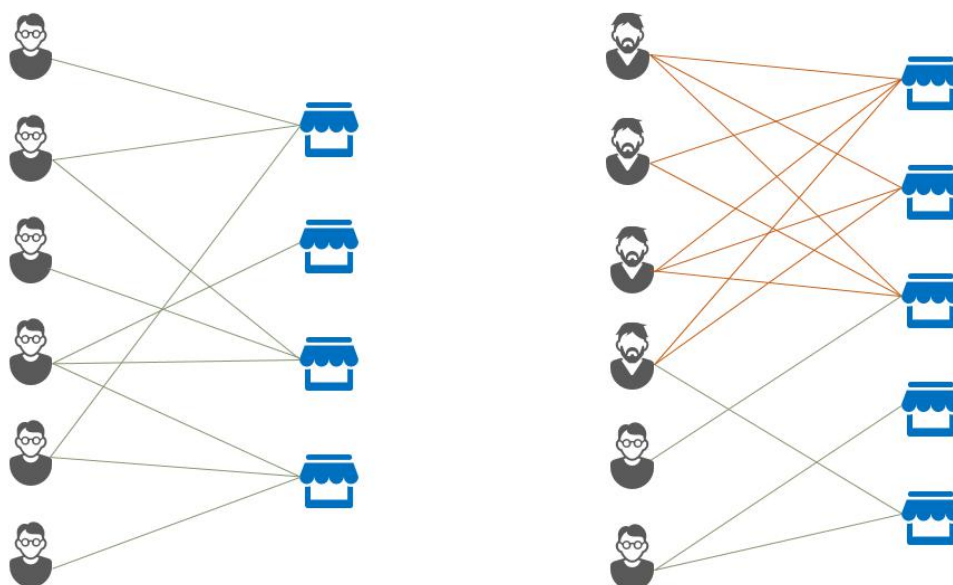


图 3 二部图下的正常连接模式（左）和 Lockstep 欺诈连接模式（右）

03

Fraudar 算法介绍

简单来说，Fraudar 定义了一个可以表达结点平均可疑度的全局度量 $G(\cdot)$ ，在逐步贪心移除可疑度最小结点的迭代过程中，使 $G(\cdot)$ 达到最大的留存结点组成了可疑度最高的致密子图。接下来我们稍微细化一下算法过程，以刷单交易场景为例（定义二部图结点集合 $S=[A,B]$ ，其中 A 、 B 分别代表消费者和店铺的结点集合），看 Fraudar 是如何从交易网络剥离出刷手和其出资店铺的。

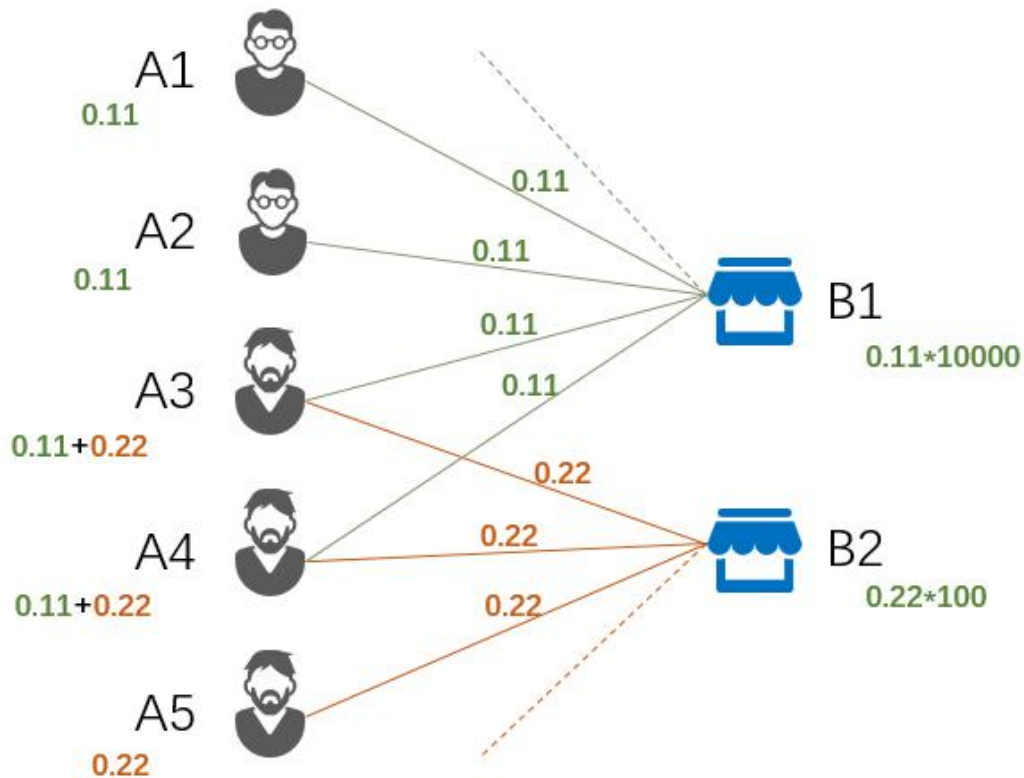
1. 定义全局度量 $G(s)$

全局度量的定义公式非常简单： $G(S) = F(S)/|S| = (F(A)+F(B))/(|A|+|B|)$ 。其中 $F(\cdot)$ 是结点可疑度的总合， $|S|$ 是当前的结点数，所以 $G(s)$ 可以理解成网络结构中



每个结点的平均可疑程度。当网络中增加一个高可疑度的结点时，其带来的 $F(S)$ 增加百分比大于 $|s|$ 带来的增加， $G(S)$ 增大；当增加一个低可疑度的结点时，其带来的 $F(S)$ 增加百分比小于 $|s|$ 带来的增加， $G(S)$ 减小；所以 $G(S)$ 是全局可疑度的一个有效表达。结点总可疑度 $F(s)=F(A)+F(B)$ ，是两类结点可疑度的总和。而结点可疑度又是其所连接的边的可疑度的总和。Fraudar 对边可疑度的定义准则是 '与 B 类结点连接的边越多，其可疑程度越小，即根据连接数降权。一个有效的边可疑度计算公式为 $1/\log(x+5)$ ，其中 x 是边的数量，实际含义是交易量越大的店铺，其交易可疑程度越小，因为大概率是真热门店铺。下图展示了客户与热门店铺和刷单店铺的交易网络中，初始各边及结点的可疑度计算，计算过程为 B 类结点边数量的确定->边可疑度计算->A/B 类结点可疑度计算。





B1为热门店铺，共1w条边，边可疑度： $1/\log(10000+5)=0.11$

B2为刷单店铺，共100条边，边可疑度： $1/\log(100+5)=0.22$

图 4 网络初始边和结点可疑度计算示意

2. 结点的迭代移除

初始可疑度确定完毕后在模型迭代过程中，Fraudar 贪心的逐步移除图所有二类结点中可疑度最低的结点。因为网络规模通常较大，如电商场景下每天会产生数百万的消费者结点和数十万的店铺结点，如果每一步迭代都遍历全部结点来定位最小值，将是非常大的开销。作者以空间换时间，为两类结点分别构建了用于快速搜索的优先树。优先树是一棵二叉树，以二部图中的 A 类或 B 类结点作为叶结点，并让父结点记录其子结点中的最小值。这样从根结点记录的全局最小值出发，



可以快速定位到该最小值所对应的叶结点，然后将其从二部图中删除，并更新网络可疑度、更新优先树。

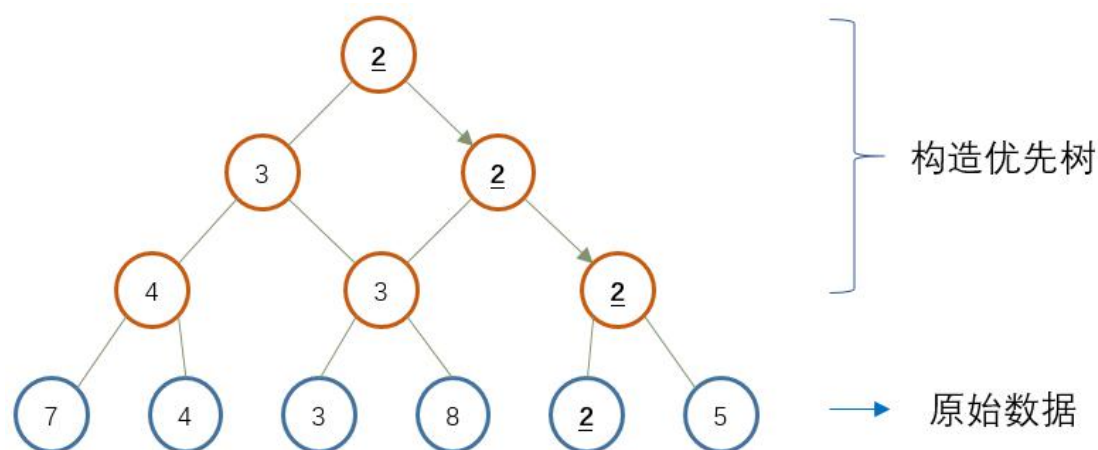


图 5 基于优先树的最小值快速定位

3. 结点可疑度更新

迭代过程对结点的删除改变了网络结构，每次迭代后都需要在保持边可疑度不变的前提下，更新剩余有关结点的可疑度。迭代初始，A 类结点中正常的低活消费者因为边连接少、可疑度低而被首先移除，如下图中的 A1 结点。同时 B1 商户因与 A1 关联的边连接被删除，其可疑度也被更新降低。如此往复，低可疑度的 A 类用户结点逐步减少、正常 B 类商户结点的可疑度也逐步降低而被删除，网络剩余结点的全局平均可疑度 $G(\cdot)$ 逐步增大，直到最后一个结点被移除而归为 0。



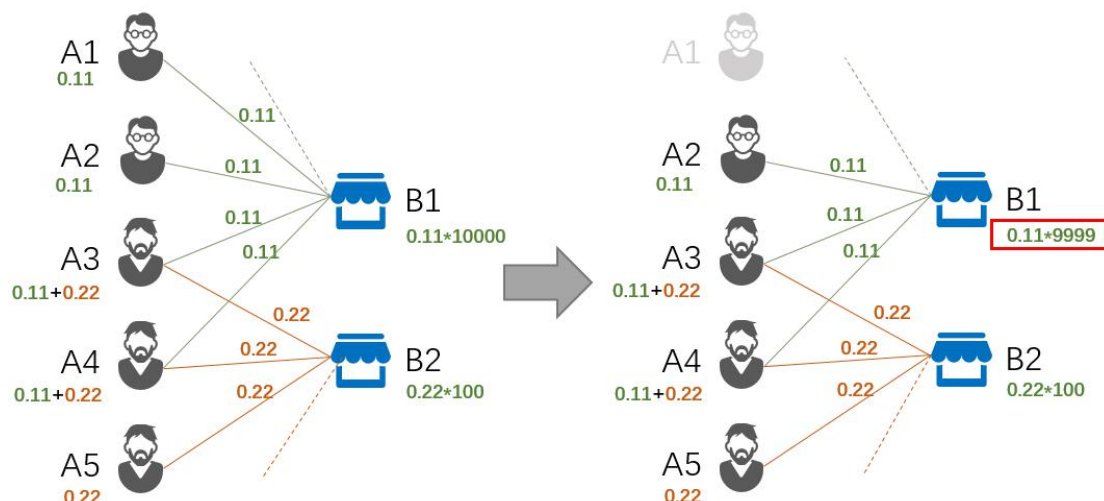


图 6 结点移除导致剩余网络可疑度更新

4. 回溯最大全局度量

全局平均可疑度 $G(S)$ 在结点的贪心迭代移除过程中会取得最大值，当全部结点迭代移除完毕时，回溯此过程中使 $G(\cdot)$ 达到最大的迭代，此时对应的留存结点即我们的目标结点，他们之间的关系网络是整个网络的最可疑致密子图。该子图中两类结点之间关系紧密，且都与外部结点连接相对稀疏，这在公共电商平台下是违反社会关系规律的一种连接模式，可以认为是拥有极大刷单概率的可疑群体。全图的一次迭代可以获得一个高危群体，将该群体的结点从图中删除后再运行一次迭代，又可以获得另一个次高危群体，如此往复可以根据业务需要得到多个群体，再进行后续排查分析。



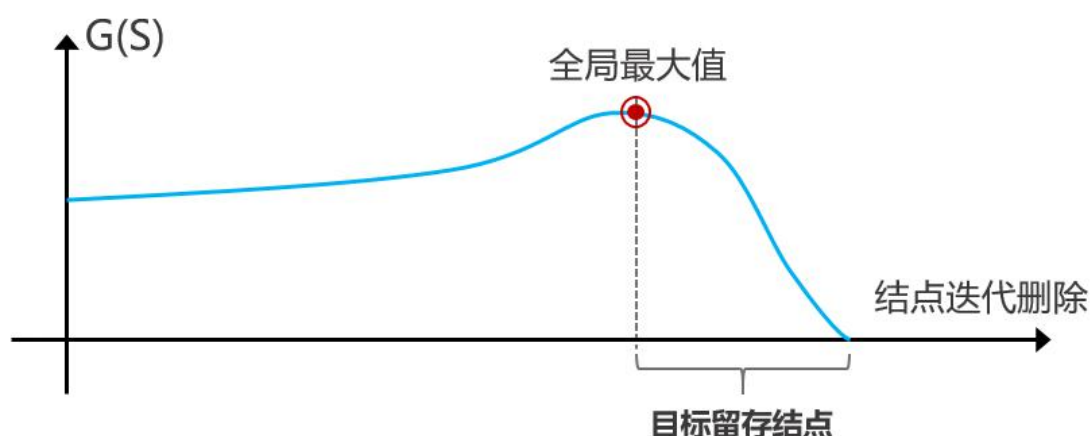


图 7 全局平均可疑度 $G(S)$ 得分曲线

04

可疑致密子图分析

从模型有效性来说，Fraudar 有着和其它无监督算法如聚类和异常检测共同的缺点，即模型输出结果的可解释性差、稳定性差，需要结合统计性指标做出二次判断。但专家判断需要大量的人工干预，且不具备泛化能力，有效性完全取决于专家业务能力。例如当我们从二部交易网络中提取致密子图时，除了刷单，可能还提取出了多个羊毛党或黄牛网络，这些拥有类似连接关系的欺诈模式都会被抓取出来，只有经过充分的二次定性才能进行后续的案件处理。当然，为了增强模型输出的可信度，也可以根据先验的业务知识对图做预剪枝，同时可以有效的减少数据量和运算量。如刷单场景下我们预先可以把所有高信用的白名单店铺从结点中移除，只保留更危险的低活店铺来提高准确度。在实际中为提高社群的挖掘准确性，我们开发了一套无监督社群发现和 supervised 模型协同训练的模型框架。



05

社群发现与有监督的协同训练

基于关系网络的无监督社群发现考量了各结点间的拓扑关系，但仅以交易或评论、关注等交互行为的频繁程度来判断风险有些片面，没有考量到结点自身的属性和行为特征。这些特征如消费者的注册时长、历史活跃度、设备稳定性，如店铺的漏斗转化率、交易增长率等都是对欺诈程度的一个有效评价。高活跃高信用的消费用户，一般是不会参与到刷单和薅羊毛等活动中去的。而有监督算法虽然使用了高维特征来对样本做刻画，但这些特征并不能很好归纳其邻接以及更远的网络关系。为将两类互补信息做有效结合，我们使用了一种两类模型协同训练的方法，如下：

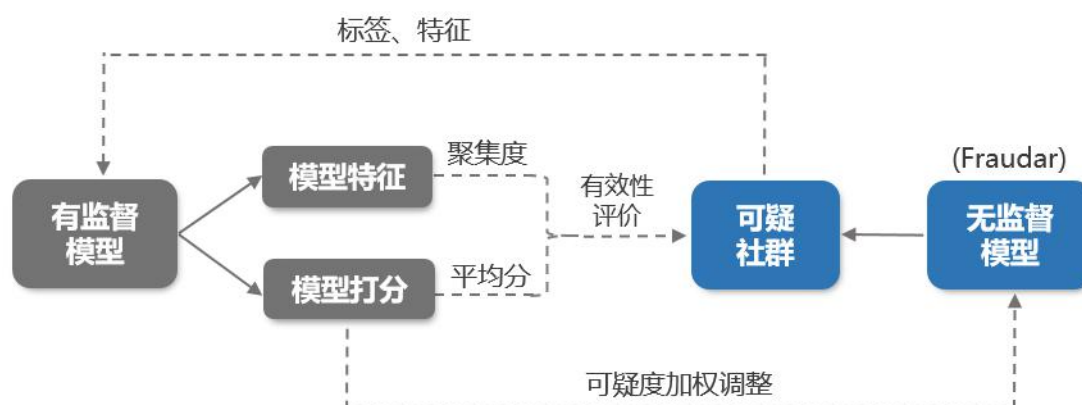


图 8 社群发现与有监督模型协同训练

在建模初期可以获得的数据包括少量的欺诈标签、用户结点特征、商户结点特征、以及用户和商户的交易关系组成的边。我们用两类结点的关联关系构建了交易二



部图，以全局可疑度作为度量标准进行 Fraudar 无监督社群挖掘；用欺诈标签、用户结点特征和商户结点特征这些单点的数据进行有监督建模，得到单点的欺诈概率。在此过程中，一方面，有监督模型的结果对无监督做指导：其模型打分结果可以对结点可疑度做加权调整，使结点可疑度能够从拓扑信息和单点行为属性上综合表达其欺诈程度；也可以通过计算输出社群的模型平均分和在特征空间类内聚集度作为社群挖掘质量的有效性评价。另一方面，有监督模型也从无监督模型收到了正向的反馈：通过对有效社群的分析获得了更多风险标签和特征构建上的指导。二者综合利用各方数据信息互相协同形成模型闭环，通过多轮迭代不断提升社群挖掘的数量和有效性。在建模初期标签稀缺时，一个简化的替代版本是以专家经验构建的多维线性打分模型来替代有监督模型，在经过几轮迭代积累足够的标签后再进行有监督学习。

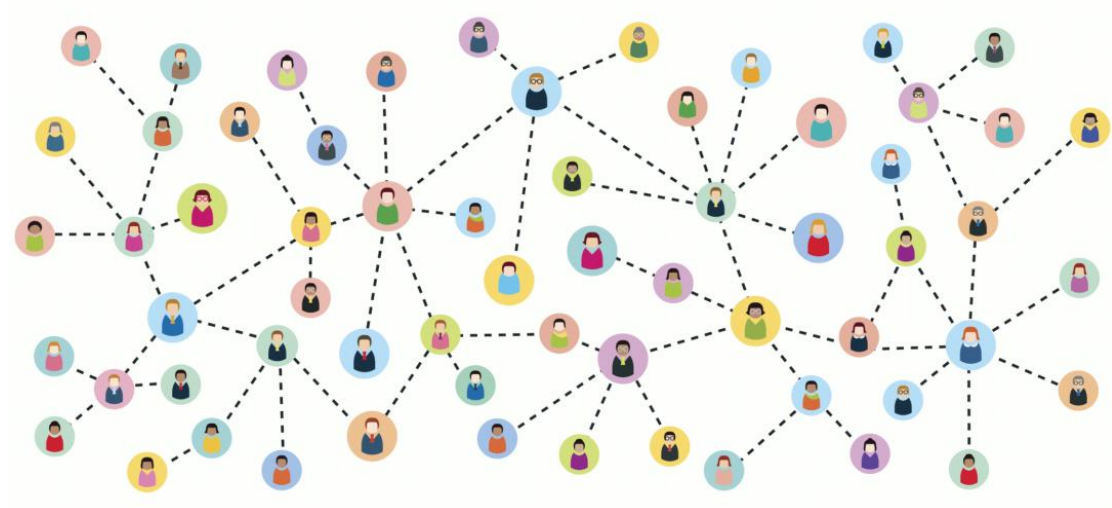
06

运算效率提升

Fraudar 的一个缺点是它的串行运算特性导致在大规模二部图上运算缓慢，其每次迭代只动态的删除一个结点并更新剩余结点状态，对于由 1 千万个消费者和 10 万个店铺构成的交易网络，一共需要迭代 1 万亿次。但在迭代初期，模型移除的结点都是交易量很少的低可疑度结点，由于单位时间内的客户交易量服从长尾分布，在总体客户中有大量这样低交易量低可疑度结点的存在。为此我们设定了一个对网络状态自适应的可疑度阈值： $2 \cdot G(S_0)$ ，即初始全局可疑度的 2 倍。在初



始时将结点可疑度低于此值的结点提前移除，可在不影响网络欺诈模式的前提下，减少约 40% 的计算量。在开始迭代后，也以 $x/1000$ 的数量做结点的批量移除来减少串行运行量，其中 x 是当前的某类结点的留存总量，之后随着留存结点数量的减少，每次批量移除的结点数也在不断减少，精度不断提升，以这种先粗后细的手段使模型在运算效率和子图挖掘有效性上做出了平衡。当然，批量的结点删除也需要在优先树的最值查找策略和网络状态更新上也做出调整，并付出相应可接受的计算代价。



07

结语

搭配上有监督模型对结点的刻画，Fraudar 成为了一把对抗二部图欺诈模式的精准利器。"如果你拍的不够好，是因为你离的不够近"，这是 20 世纪最伟大的战地



摄影师罗伯特·卡帕的名言。在电商、支付等行业的蓬勃发展带来的各类欺诈与反欺诈攻防大战中，如果你做的不够好，那是因为你分析的不够细致，没有挖掘出所有的欺诈模式。如果你做的不够好，也可能是你站的不够高。抛开繁杂的细节，不妨站在更高的维度把交易抽象成一张网，也不妨试试今天的 Fraudar。

今天的分享就到这里，谢谢大家。

参考文献：

[1] "FRAUDAR: Bounding Graph Fraud in the Face of Camouflage." Hooi, Bryan , et al. the 22nd ACM SIGKDD International Conference ACM, 2016.



携程金融自动化迭代反欺诈模型体系



文章作者：携程技术团队

编辑整理：Hoh

内容来源：《携程人工智能实践》

导读：支付欺诈风险是携程金融风控团队的主要防控对象，它一般是指用户卡片信息或账号信息泄露后，欺诈分子利用这些信息在携程平台进行销赃，侵害用户资金安全，给用户和携程平台带来损失。携程金融风控团队需要在不影响正常用户自由出行的前提下，对这样的风险交易进行精准识别并实时拦截，从而保护用户资金安全。支付欺诈风险具备以下 3 点特性。

1. 高对抗性



欺诈分子的作案手段绝非一成不变，他们也会根据我们的策略拦截结果对作案方式不断调整，不断形成风险转移，如果我们的策略模型不及时追踪这种变化，则无法做到“见招拆招”。

2. 复杂性

为了躲避风险控制的策略规则，欺诈分子也会尽力模仿正常用户的消费行为，而且从目前的数据来看，支付欺诈场景的批量操作行为比业务作弊场景少很多，原因是在携程平台上，欺诈分子相当于一个承接正常用户需求的代订中介，在没有需求的情况下，他们没有必要通过走量的形式进行销赃。这样的业务特性导致支付欺诈多为无规律性的单点攻击，普通规则在模拟这种复杂逻辑场景的时候会存在一定劣势，通常要么精度不够，造成误拦截过多，要么无法捕捉作案手法。与普通规则不同的是，模型可以在多变量前提下进行数据拟合，从而捕捉复杂作案手法。

3. 坏样本稀少

在支付反欺诈场景中，坏样本的主要来源有两部分：在风控运营过程中，人工拦截或规则、模型自动拦截的风险订单；风控策略无法识别并最终产生损失的风险订单。在风险订单中，风控策略能够拦截的占绝大部分，而我们需要注意的是无法识别的那部分，但这部分案件样本相当稀少，这给规则和模型的学习带来不小的挑战。

携程金融风控团队为了应对支付欺诈风险的特性，使用了一套自动化迭代反欺诈模型体系，以加快模型迭代速度，学习最新作案手法，在整个过程



中尽量减少工程师的人力投入成本，并且用生成式对抗网络辅助，增加稀少案件样本的数量，使模型的学习更加充分，最终达到"见招拆招"的目的。

01

风控变量体系

数据决定模型的上限。图 1 是风控变量体系。该体系主要基于携程平台的账户、支付、出行、金融及 IP Location 等数据构建。我们的风控变量体系主要从衍生方法的多样性、业务含义的完备性两个角度去考虑。

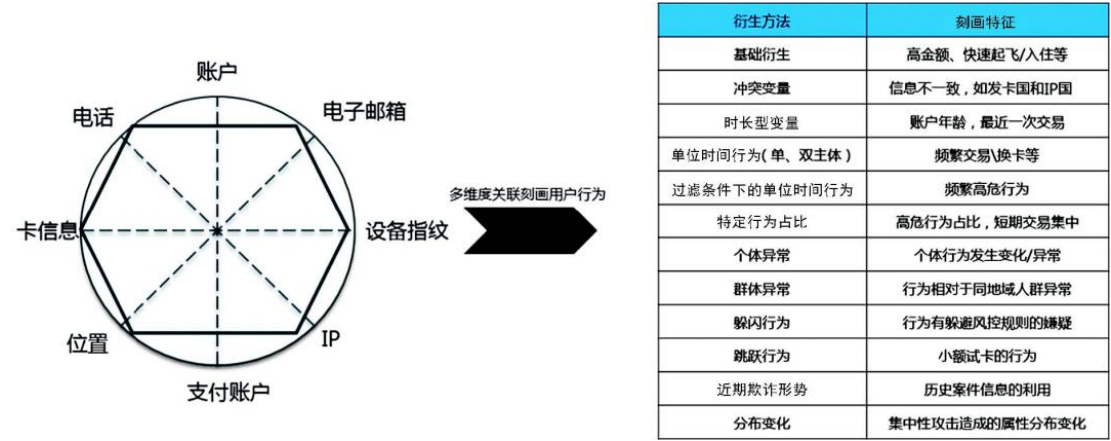


图 1 风控变量体系

02

自动化迭代模型框架

上文介绍了我们的风控变量体系，下面介绍自动化迭代反欺诈模型框架，如图 2 所示。



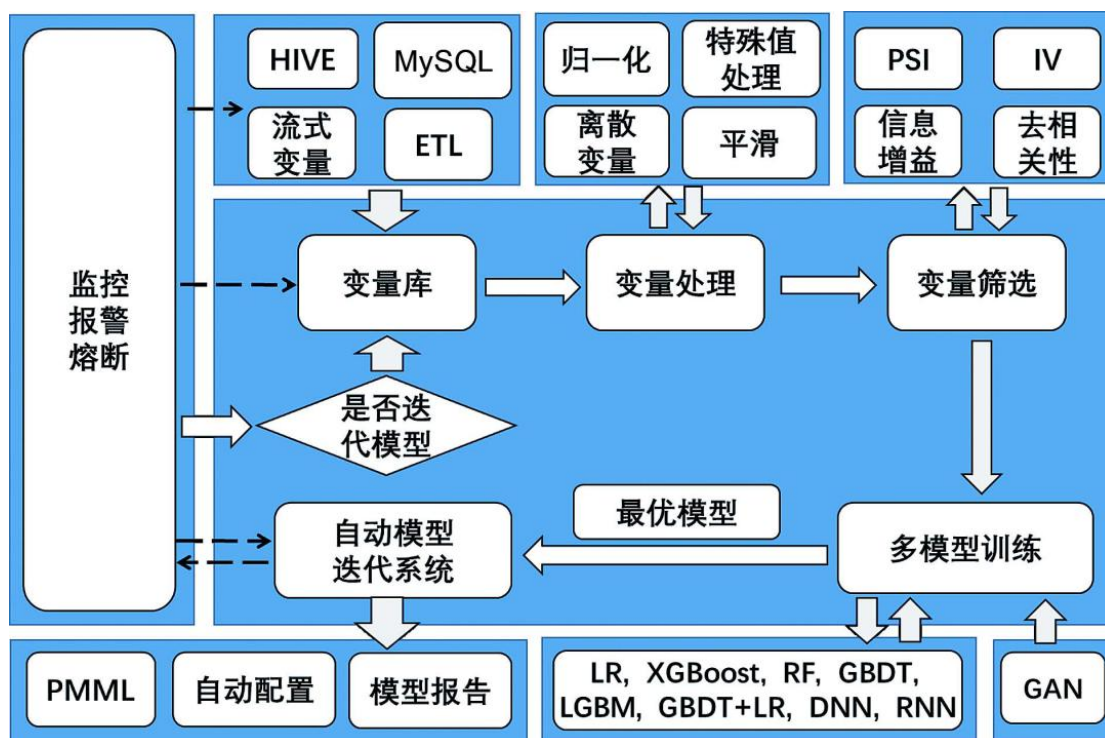


图 2 自动化迭代反欺诈模型框架

1. 迭代触发

迭代触发的条件可以自行设定，该模型可以按照时间周期迭代，也可以在实际业务效果下降后触发迭代。

2. 变量库

我们的变量库由线上实时计算的变量和离线 T+1 清洗的变量组成，当新的模型触发迭代时，近期的样本会构成候选变量池。

3. 变量处理和变量筛选

我们对候选变量先进行一轮特征工程。这主要包含两个步骤。

- 变量检查。我们将包括变量自身分布的稳定性指标（PSI）在内的有问题的变量剔除。



- 缺失值/特殊值填充。因为我们的流程中会用到 DNN 深度学习的方法，而此类方法对缺失值较为敏感，所以我们需要将这部分特殊情况用正常的取值情况去替代。我们目前使用的替代方法是找到与特殊值黑样本率最接近的正常值区间，用区间中的值进行特殊值替代。如果变量是离散型变量，那么我们需要将其转成 one-hot 形式。

4. 算法衍生变量

利用上一步的结果，通过深度学习的方法进行表征学习，目的是在基于业务衍生的变量之外，通过算法自动生成一些变量。由于目前对深度学习的弱解释性没有较好的解决方案，所以，为了保证主模型的可控性，我们建议限制深度学习模型的网络结构复杂性，同时控制最终入选模型的变量个数。我们以业务衍生变量为主，以算法衍生变量为辅。

5. GANs

我们利用 GANs 模型，通过对真实欺诈样本集数据分布的学习，自动合成符合该分布的欺诈样本，将人造欺诈样本纳入样本库，改善欺诈案件稀少的情况，增强模型对欺诈样本的学习。

6. 主模型

原则上主模型的类型没有限制，只要能够实现分类器的功能都可以。为了提高自动化迭代效率，我们推荐使用随机森林、GBM、GBDT、XGBoost 或 LGBM 等基于决策树的模型作为主模型，原因如下。



- 此类模型能实现在每次拆分时自行从候选变量池中挑选变量的功能，可以节省根据变量重要性手动挑选入模变量的时间。
- 此类模型可以捕捉变量与目标之间的非线性关系，并且自动交互组合变量，减少特征工程时间。
- 此类模型在每次拆分时，能够有效利用变量的碎片化信息。

7. 部署

框架输出：输出 PMML 格式模型打分文件、特征工程过程代码、算法衍生变量生成方法。其中，特征工程过程代码可以认为是原始变量到入模变量之间的映射方法。

变量配置：如果模型有新变量（原本生产环境未配置，只能离线清洗计算）入选，则需要将这部分变量配置至生产环境。

8. 确定阈值

在传统情况下，我们一般通过跨期测试集的效果来确定模型的阈值，由于支付欺诈场景的高对抗性，我们建议将近期样本放入模型中进行训练，这样可以压缩跨期测试集的空间，甚至使模型没有跨期测试集。针对这种情况，我们采用的方案是，先将模型部署上线但不使其生效，短期观察一段时间后，根据生产环境的真实表现确定阈值，参考的依据主要有阈值以上的交易占比及相应的准确率。

与传统建模流程相比，自动化迭代模型框架在追求快捷高效的同时有两个问题。



- 自动化迭代模型框架缺少了对所有入模变量人工审核的过程，减少了对入模变量的了解，降低了模型的可控性。
- 由于支付欺诈场景的高对抗性，所以我们建议使近期样本进入训练集，让模型学习近期的风险特征，这样会造成每次迭代几乎没有用来检验模型是否会出现过拟合的跨期测试集。

针对这些问题，为了最终实现自动化迭代模型体系对支付欺诈风险的高对抗性、复杂性的克服，达到模型效果与模型可控性的平衡，我们需要一套完备的监控体系为模型保驾护航。

9. 监控体系

变量监控：对生产环境中的变量监控 PSI、特殊值（包括缺失率）占比变化。

模型监控：通过模型打分进行 PSI 监控。

业务效果监控：针对支付反欺诈场景，我们主要监控模型线上交易的真实拦截率与预估拦截率是否有差异，以及通过事后的人工定性监控模型拦截的准确率是否符合预期。

在运营维护自动化迭代模型框架的过程中，我们认为除了新变量配置流程目前需要人工介入，对业务的深入了解和对支付欺诈案件的人工分析也是不能节省人工投入的，毕竟任何模型算法只是工具，一个好的业务沉淀转化的变量给模型的效果带来的提升，要远远大于算法优化带来的提升。



03

RNN 表征学习

在支付反欺诈场景中，传统方法衍生的特征一般刻画用户在某个时间窗口内进行了几笔交易、换了几张信用卡、累计消费了多少金额等，也就是量化用户在这段时间内干了什么，但是这种衍生方法在用户时序行为信息表达上有不足之处，它没有刻画用户先干了什么后干了什么。我们可以利用 RNN 捕捉时序行为的特性，在主模型进行训练前，先基于用户 UBT 行为数据训练一个 RNN 子模型，将这个模型的隐藏层输出作为主模型的输入变量，如图 3 所示。

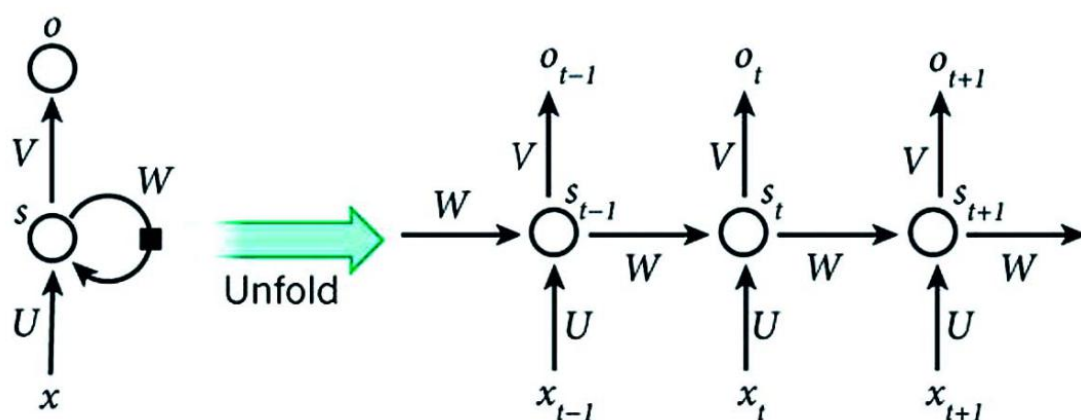


图 3 RNN 结构示例图

UBT 数据主要分为 action、pageview 数据，action 是指用户在浏览时进行的细节操作，如筛选、排序、填写等，pageview 数据是用户在浏览的页面的 ID，每个页面都有对应的 ID，同样是酒店详情页，如果酒店



不相同，则页面的 ID 也不一样。我们以 pageview 数据的应用为例，介绍利用 UBT 数据建模的主要步骤。

1. 业务转化

如果直接将 pageview 数据作为模型的输入，那么会出现两个问题。

- 用户的相同的浏览动作，如浏览页面详情，会因页面来自不同的酒店或航司而产生不同输入，即输入不稳定。
- 页面的 ID 在作为输入时，从业务角度理解起来过于枯燥，不够生动。

针对以上问题，我们花了大量时间对 UBT 数据进行了人工汇总，以业务理解为核心，有选择地将原本枯燥无味的页面转化为用户的行为动作，并将用户在页面的停留按照停留时长的分组转化为几种输入动作。业务转化示例如图 4 所示。



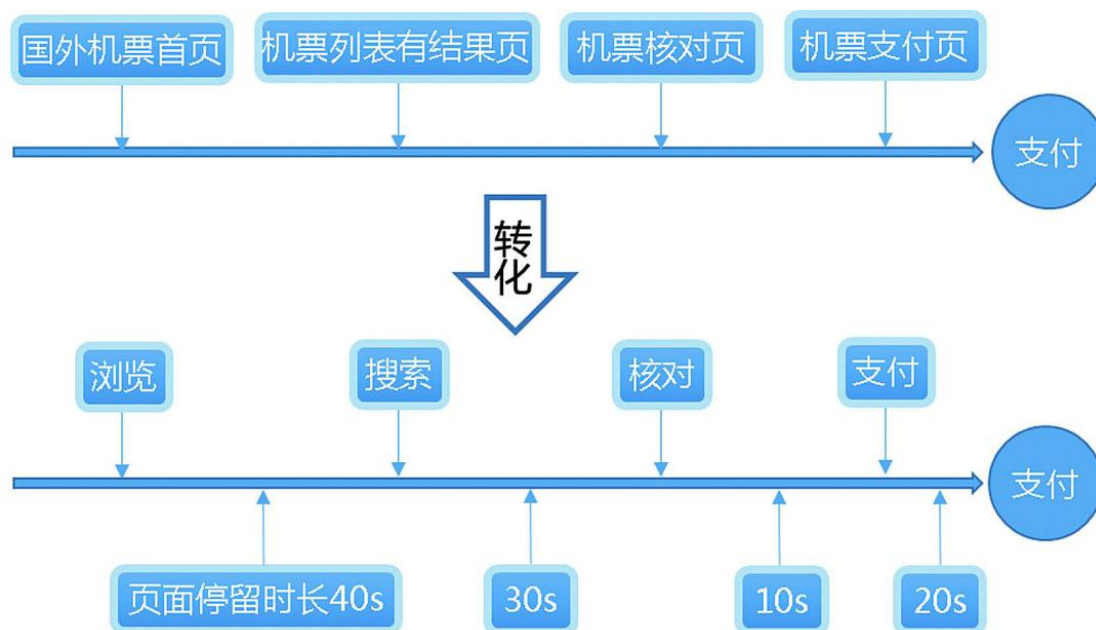


图 4 业务转化示例

2. 动作字典

我们将希望能采集到的用户浏览动作汇编成一个输入字典，如表 1 所示。

ID	输入元素
0	浏览
1	搜索
2	关注详情
3	价格排序
4	页面停留 10s 以内
5	页面停留 10 ~ 20s
6	页面停留 20 ~ 30s
7	页面停留 30s 以上
⋮	⋮
N	支付

表 1 输入字典



3. one-hot

用传统的 one-hot 的方式将这些动作逐一量化为向量，使其生成矩阵，并将矩阵输入模型。

最后，在训练完模型后，提取模型隐藏层的输出，将它们作为特征输入主模型。由于这部分特征表达的是用户的时序行为信息，其数据来源和表达信息的维度都与以往传统的衍生变量有所不同，所以我们在实践中发现，这部分特征与传统衍生特征之间的冗余性非常低，对主模型来说是一个非常好的信息补充。

04

自动化与传统方法的效果对比

1. 传统方法

在传统情况下，我们构建一个模型的时间成本要高于构建规则的时间成本，因此以往当我们上线一个模型时，稳定性是我们参考的重要指标。我们希望上线一个模型后，模型能正常表现几个月甚至半年。然而，以风控支付反欺诈模型——天眼模型为例，在天眼一期的时候，我们构建了一个模型，在 OOT1（第一个跨期测试数据集）中，当准确率为 80% 时，模型的召回率为 12% 以上；仅仅过了一个月，在 OOT2（第二个跨期测试数据集）中，同样当准确率为 80% 时，模型的召回率衰减到 7.9% 左右，如图 5 所示。



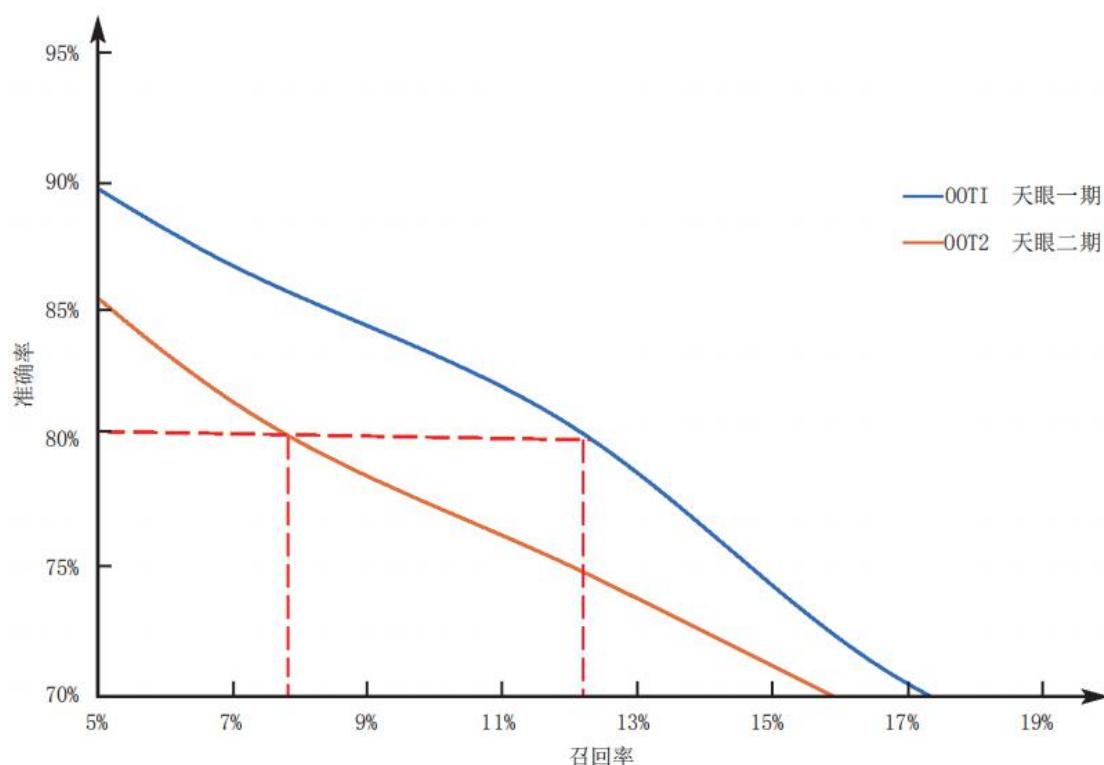


图 5 天眼一期效果

由于模型的衰减发生在两个跨期测试集之间，而不是发生在训练集与测试集之间的，所以我们排除了模型本身的过拟合造成的效果衰退。分析后发现，造成模型覆盖率降低的原因是支付欺诈场景的高对抗性，欺诈分子在不断变换他们的欺诈手法，因此历史数据训练的模型无法覆盖最新的手法。

2. 自动化迭代 vs 传统方法

在天眼二期的时候，我们采用了自动化迭代的流程框架，在降低建模成本的同时使模型与欺诈形势保持同步。如图 6 所示：



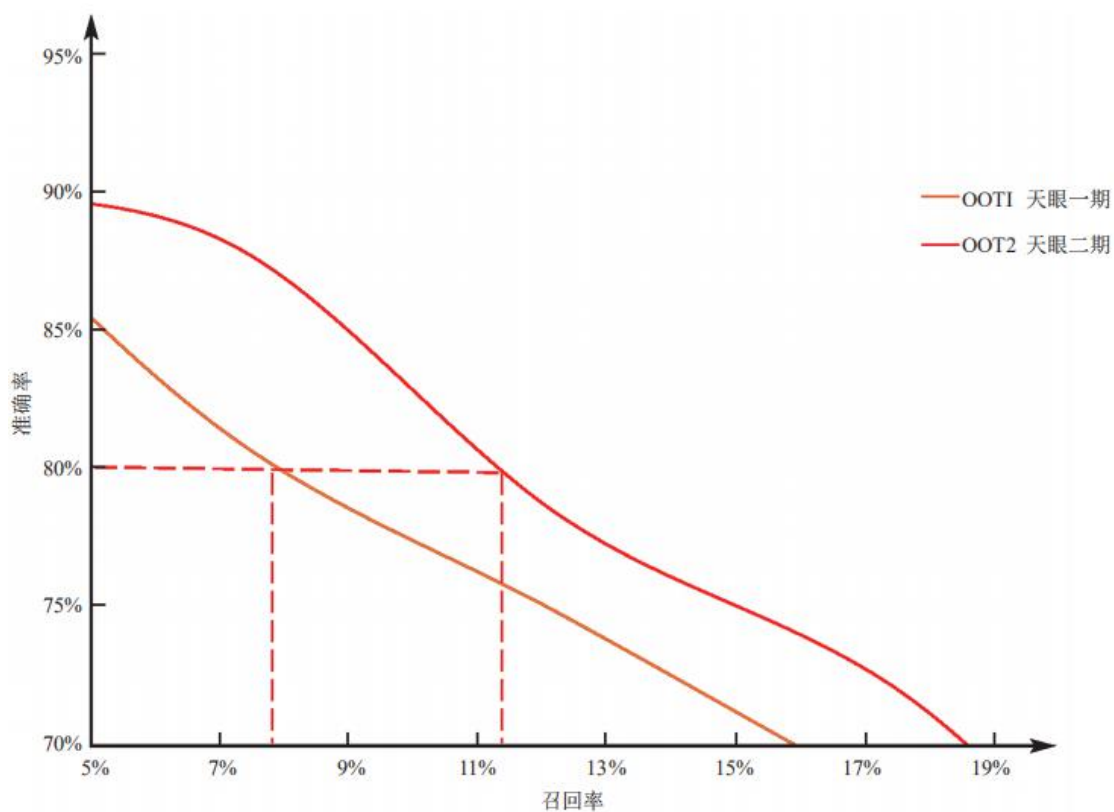


图 6 天眼二期效果 vs 天眼一期效果

在天眼一期效果已经下降的 OOT2 中，天眼二期的召回率能在准确率为 80%的情况下重新回到 11.5%左右。



微众银行在联邦推荐算法上的探索及应用



分享嘉宾：谭奔 微众银行 高级研究员
编辑整理：王少华
内容来源：微众·AI
出品平台：DataFunTalk
注：欢迎转载，转载请留言。

导读：现如今推荐系统已经得到了广泛的应用，其性能通常是随着数据量的增加而提高。但现实中，数据往往分散在各个提供方手上，且随着数据监管政策趋严，以及数据隐私保护问题越来越受到重视，导致目前集中式的推荐系统难以进一步提高。因此，联邦推荐算法得以兴起。联邦推荐可以在数据不出本地的情况下，在加密的环境中实现和传统推荐算法一致的性能。其中，微众银行的开源平台 FATE 在这方面落地了多种算法，得到了越来越多的应用。今天将为大家带来微众银行在联邦推荐算法上的探索及应用。

主要包括：



- 推荐系统简介
- 联邦推荐系统
- 联邦推荐算法原理及实现
- 联邦推荐云服务和应用

01

推荐系统简介

1. 推荐系统应用广泛



推荐系统在互联网中的应用非常广泛，包括：电商、社交、信息流、在线视频、在线广告等场景。

2. 推荐系统极大提高效率





个性化推荐服务



精准营销

YouTube: 60%+ more clicks [Davidson et al. 2010]

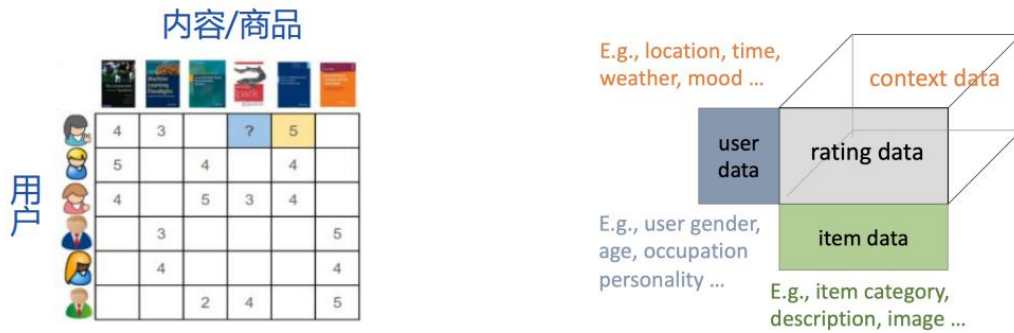
Netflix: 80%+ more movie watches [Gomez-Urbe et al 2016]

Amazon: 30%+ more page views [Smith and Linden, 2017]

同时，推荐系统也是 AI 在商业领域成功应用的典型代表。推荐系统大大提高了用户获取信息的效率，帮助用户在海量信息中快速找到匹配的信息；另外，推荐系统还可以帮助企业找到合适的用户群，从而降低获客成本，提高投资回报率。比如，在应用推荐系统后，Youtube 视频点播率提高了 60%；Netflix 电影的播放量提高了 80%，亚马逊的用户浏览量提高了 30%，这都是很大，很可观的提升。

3. 推荐系统的原理





输入: 用户在内容/商品上的历史行为, 用户、内容/商品的信息, 例如用户基础信息、兴趣、内容/商品属性等

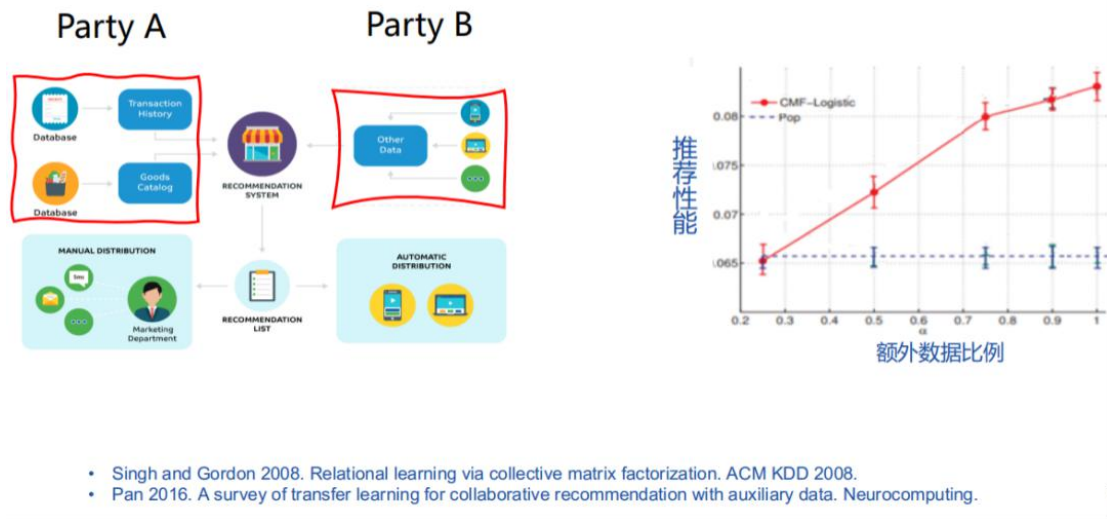


输出: 用户与内容/商品发生交互的概率, 交互行为根据场景有点击、购买、评分等

在数学上, 推荐系统是一个矩阵填充的问题。其中, 矩阵的行表示用户, 列表示内容/商品, 矩阵中的值表示用户的行为。根据场景的不同, 值可以是点击, 购买, 评分。以评分为例, 1-5 表示对应的评分, 空白表示用户在这个商品上还没有评分。我们的目的是根据已有的行为记录来预测空白的值, 然后根据预测的值进行推荐。为了更精准的进行推荐, 还会用到很多其它的信息: 包括用户信息 (如用户的性别、年龄、职业等维度) 和商品信息 (如商品的类别、描述文字以及推荐的环境数据 (如推荐发生的时间点))。



推荐系统是数据驱动的应用 – 数据越多越精准



推荐系统通常会与多个数据方进行合作，从而收集更多的信息，以此提升推荐的效果。因为数据越多，对用户了解越深入，覆盖面更广，效果的提升就越大。图中相关研究也表明：随着额外数据的增加，性能随之提高。

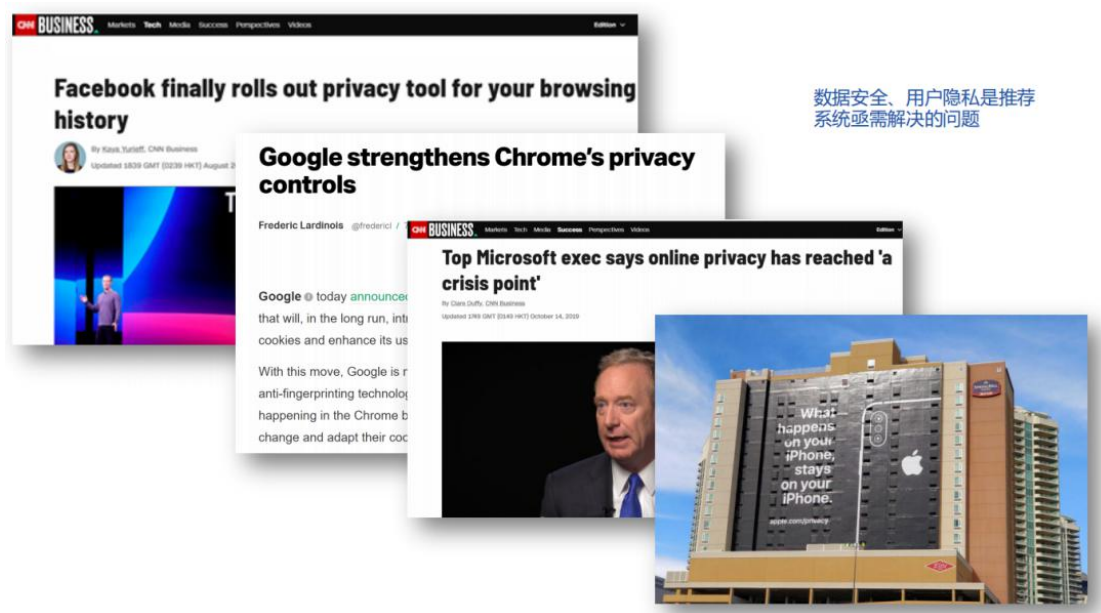
4. 推荐系统面临的问题

① 数据孤岛问题



推荐系统是大数据驱动的应用。在理想情况下，大家可能认为数据质量很好，标签数据很充足，数据也是可以集中收集，并且触手可得的。但现实是，数据通常是分散隔离的，存储在不同的企业，即所谓的数据孤岛问题。

② 数据监管问题

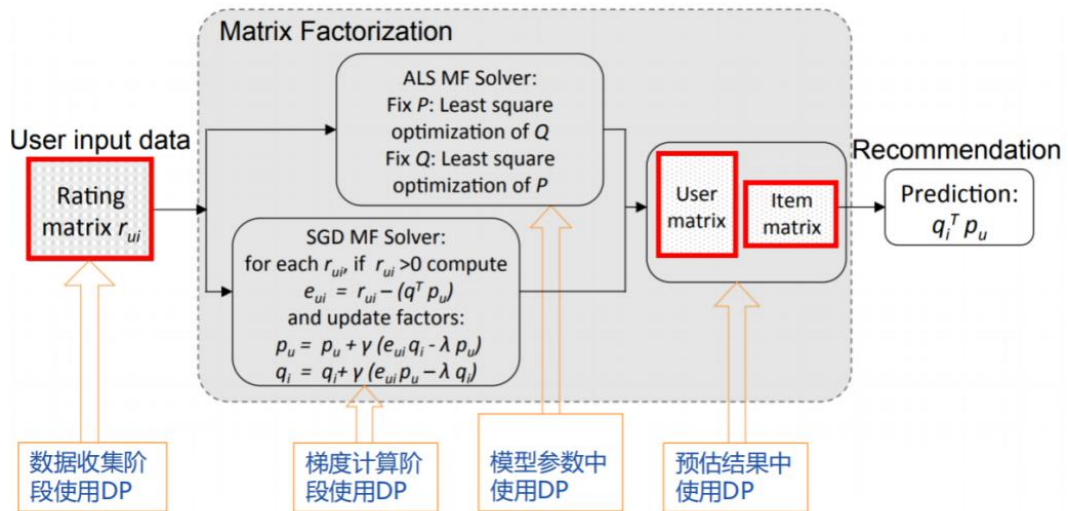


例如欧盟的 GDPR 和国内监管政策的严格，隐私保护是推荐系统必不可少的一个环节。随着数据监管体系不断完善，各个企业都在不断的加强对用户隐私的保护。

5. 差分隐私解决方法



差分隐私可用于建模的各个环节 [Knijnenburg and Berkovsky, 2017]



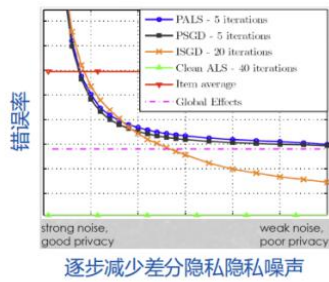
Bart P. Knijnenburg and Shlomo Berkovsky, 2017. Privacy for Recommender Systems. RecSys 2017 Tutorial.

10

在这种情况下，比如差分隐私是一种比较常用的隐私保护方法。它是在数据中加入一些差分隐私来保护用户的隐私。它可以用于建模的各个环节，比如在使用户数据收集的过程中加入，也可以在建模的过程中，比如梯度计算，模型参数中加入差分隐私噪声，还可以在预估结果中加入噪声。



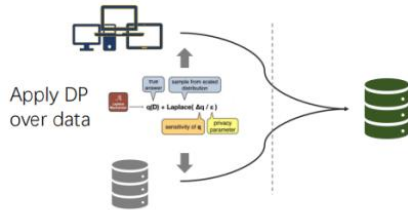
差分隐私噪声带来性能下降



增加DP噪声
保护更多隐私
带来性能下降

我们需要新的技术:

与集中收集数据相比，性能无损



加入DP噪声后的数据，仍有可能被集中管理。

数据不出本地，保障数据安全和用户隐私

11

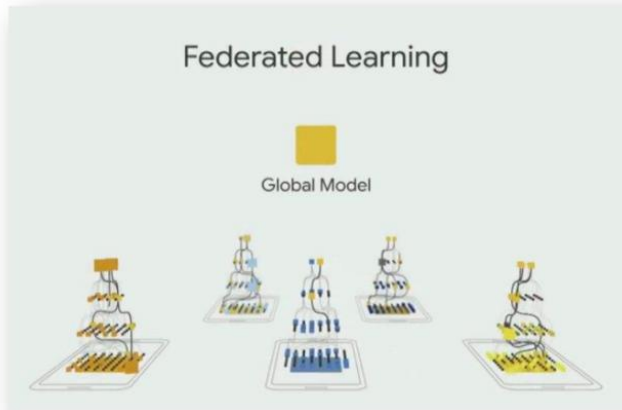
但是，现在的方法，包括差分隐私并不能完全满足我们的需求。随着差分隐私噪声的增加，确实可以保护隐私，但性能是下降的。如上图，在增加差分隐私噪声的情况下，性能下降非常快。我们希望，有一种新的技术可以在保护隐私的情况下做到性能无损。另外一个问题是在，加入噪声的数据还是有可能被集中收集，所以其并没有从根本上解决数据安全的问题。我们希望新的技术可以做到数据不出本地，从而保障数据安全和用户隐私。

02

联邦推荐系统

1. 联邦学习的优点





性能无损

- 'A fed B' 几乎等于 'A+B'

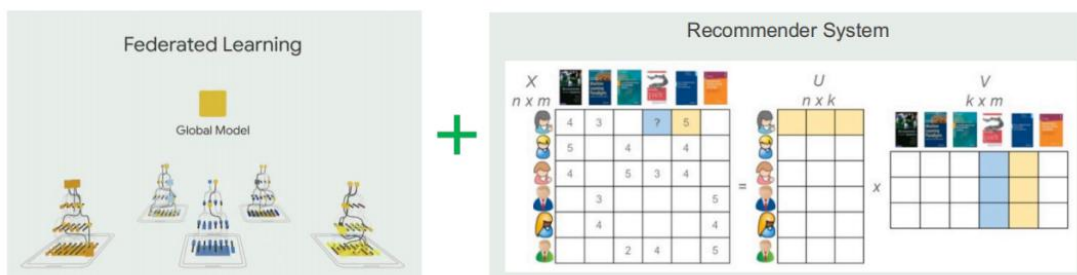
保障数据安全和用户隐私

- 原始数据不出本地
- 仅模型参数被加密传输

Yang et al. 2018, Federated machine learning: concepts and applications. ACM TIST.

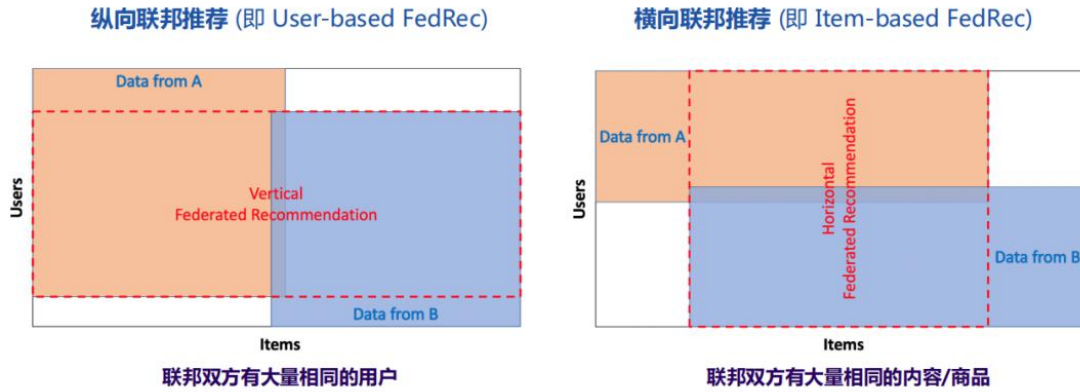
- 性能无损，A fed B 在保护隐私的前提下，性能和直接收集 A 和 B 数据的效果是一样的
- 原始数据不出本地
- 在加密条件下交换模型的参数来构建模型

2. 联邦推荐系统的分类



推荐系统中的核心数据为用户和物品的交互行为数据。根据物品和用户的共享情况，可以将推荐系统分为：





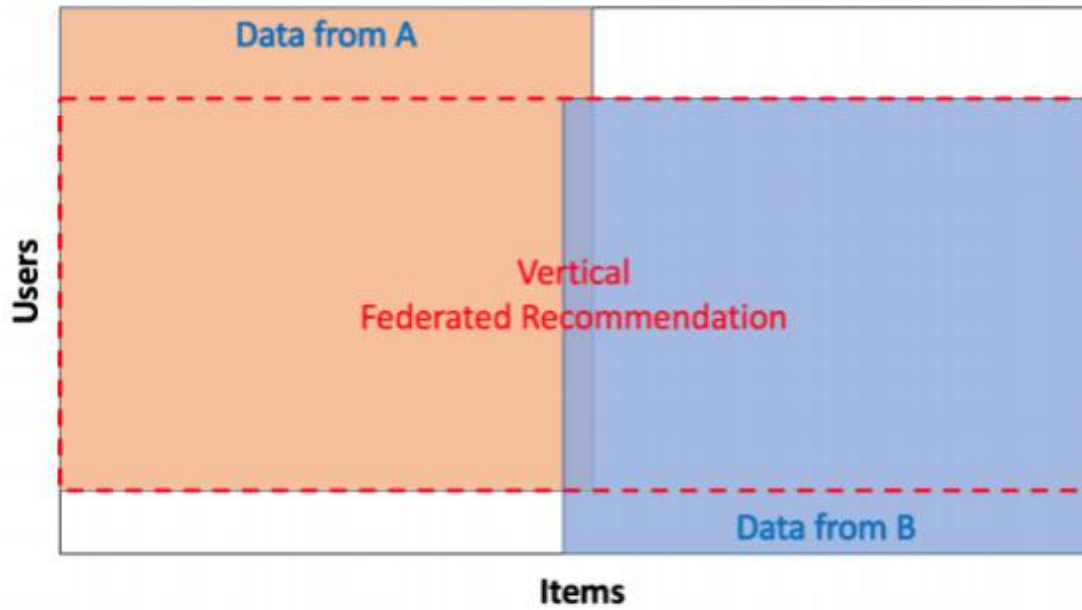
- 纵向联邦推荐：有大量相同的用户，但是 item 不同，因此也叫 user-based 联邦推荐系统。
- 横向联邦推荐：有大量相同的物品，但是用户不同，也叫 item-based 联邦推荐系统。

03

联邦推荐算法原理及实现

1. 纵向联邦推荐

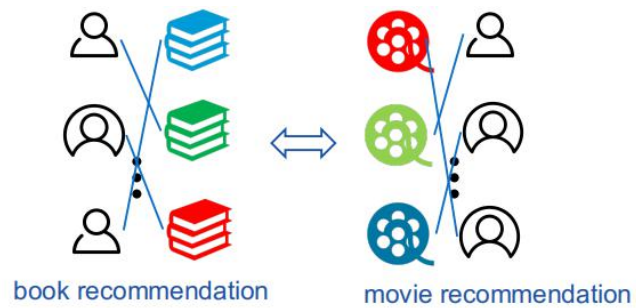




联邦双方有大量相同的用户

纵向联邦推荐介绍：

Example:
相同用户,
不同内容/商品



	4	3		?	5	
	5		4		4	
	4		5	3	4	
		3				5
		4				4
			2	4		5

Party A



No data exchange

	4			4	3	
	5		3		4	
	4		5	3	4	4
		3	4			5
		4		5		4
	3		2	4		5

Party B



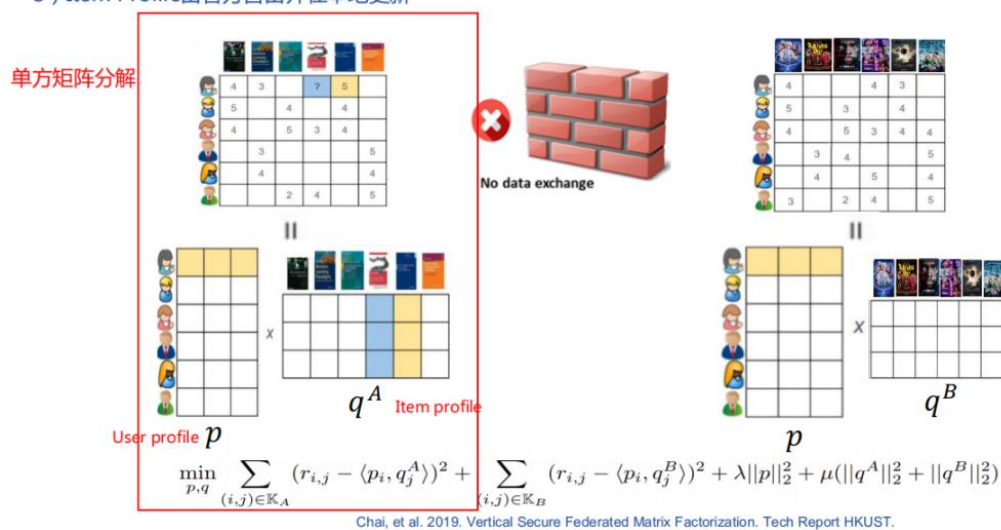
在这个场景中，书籍和电影这两个推荐系统有共同的 user，但是 item 不同。

根据一些研究指出：将这两个推荐系统进行联合，可以提高推荐效果，因为这两个推荐系统的用户偏好是一样的。

纵向联邦矩阵分解：

纵向联邦矩阵分解 [Chai et al. 2019]

解决方案: 1) 矩阵分解中的 User Profile 由参与方共享；2) 各方 User Profile 梯度加密传输给第三方，汇总并更新；3) Item Profile 由各方自由并在本地更新



以矩阵分解为例，在数据不出本地的情况下，如何构建推荐系统？

在单方的情况下，用户的评分矩阵可以分解为两个低维矩阵的乘积，即 user profile p 和 item profile q 。这两个矩阵乘积可以很好的拟合历史数据，同时可以填充矩阵中的空白值，并利用这些值进行推荐。

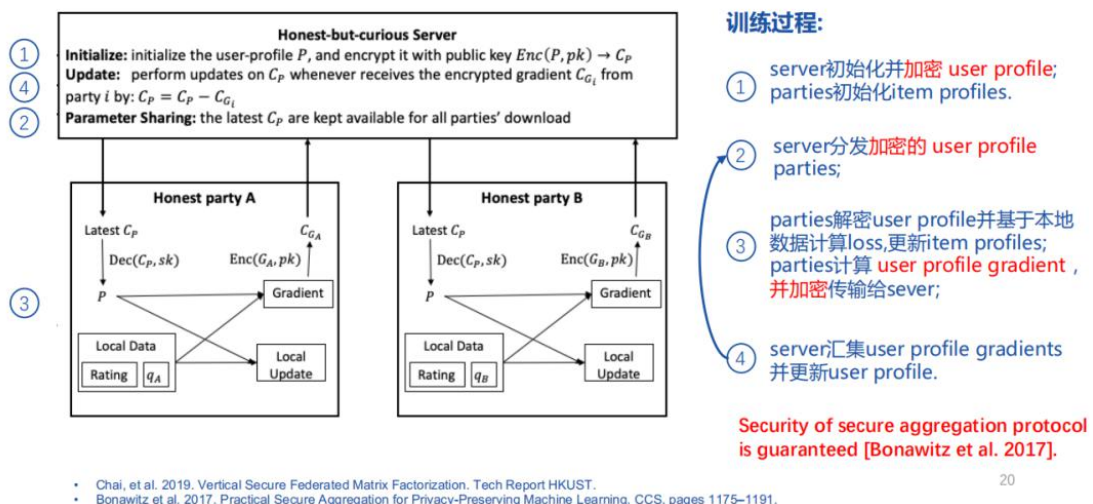
在多方的情况下，可以把两者的矩阵分解合并在一起，即图中的公式：第一项为 A 方的矩阵分解，第二项为 B 方的矩阵分解，第三项是正则项。其中，两方的 user profile 是共享的，item profile 是各自独有的。为了让两方可以及



时的获得最新的 user profile，通过引入第三方服务器，由它来维护 p ，并实时分发给 A 方和 B 方。

纵向联邦矩阵分解的训练过程：

纵向联邦矩阵分解 [Chai et al. 2019]



- ① 服务器初始化并加密 user profile，参与两方各自初始化 item profile
- ② 服务器分发加密的 user profile 给 A 和 B 方
- ③ AB 两方解密 user profile 并基于本地数据计算 loss，更新各自的 item profile，然后两方分别计算 user profile 的梯度，并加密传输给服务器
- ④ 服务器汇集 user profile 梯度并更新 user profile 矩阵
- ⑤ 如此迭代，直到模型收敛

纵向联邦矩阵分解在 FATE 中的实现：

计算更新的公式如下：



矩阵分解目标函数：

$$\min_{p,q} \sum_{(i,j) \in \mathbb{K}_A} (r_{i,j} - \langle p_i, q_j^A \rangle)^2 + \sum_{(i,j) \in \mathbb{K}_B} (r_{i,j} - \langle p_i, q_j^B \rangle)^2 + \lambda \|p\|_2^2 + \mu (\|q^A\|_2^2 + \|q^B\|_2^2)$$

各参与方计算use profile梯度、更新item profile公式：

$$p_i^t = (1 - 2\gamma\lambda)p_i^{t-1} + G_{A,i}^{t-1} + G_{B,i}^{t-1}$$

$$q_j^{A,t} = q_j^{A,t-1} - \gamma \nabla_{q_j} F(p^{t-1}, q^{A,t-1}), \text{ where } j : (i, j, r) \in \mathbb{D}_{n_b}^A$$

$$q_j^{B,t} = q_j^{B,t-1} - \gamma \nabla_{q_j} F(p^{t-1}, q^{B,t-1}), \text{ where } j : (i, j, r) \in \mathbb{D}_{n_b}^B$$

核心逻辑

Clients send shared profile
to 3rd party coordinator



3rd party coordinator
aggregate and broadcast
back to Clients

```
while self.aggregator_iter < self.max_iter:
    LOGGER.info(f"start {self.aggregator_iter}_th aggregation")

    # train
    self._model.train(
        data, aggregate_every_n_epoch=self.aggregate_every_n_epoch)

    # send model for aggregate, then set aggregated model to local
    modify_func = typing.Callable = functools.partial(
        self.aggregator.aggregate_then_get,
        degree=epoch_degree * self.aggregate_every_n_epoch,
        suffix=self._iter_suffix())
    self._model.modify(modify_func)
```

```
def aggregate_model(self, ciphers_dict=None, suffix=tuple()) -> Weights:
    models = self.get_models_for_aggregate(ciphers_dict, suffix=suffix)
    total_model, total_degree = reduce(lambda x, y: (x[0] + y[0], x[1] + y[1]), models)
    total_model /= total_degree
    LOGGER.debug("In aggregate model, total_model: {}, total_degree: {}".format(total_m
    return total_model

def aggregate_and_broadcast(self, ciphers_dict=None, suffix=tuple()):
    """
    aggregate models from guest and hosts, then broadcast the aggregated model.

    Args:
        ciphers_dict: a dict of host id to host cipher
        suffix: tag suffix
    """
    self.model = self.aggregate_model(ciphers_dict=ciphers_dict, suffix=suffix)
    self.send_aggregated_model(self.model, ciphers_dict=ciphers_dict, suffix=suffix)
    return self.model
```

纵向联邦矩阵分解在 FATE 中的使用：

首先是准备好数据，数据是以三元组形式保存的，包括有 user ID，item ID 和用户的评分。再配置好 FATE job 的文件，最后 submit job，就可以看到模型训练的结果。



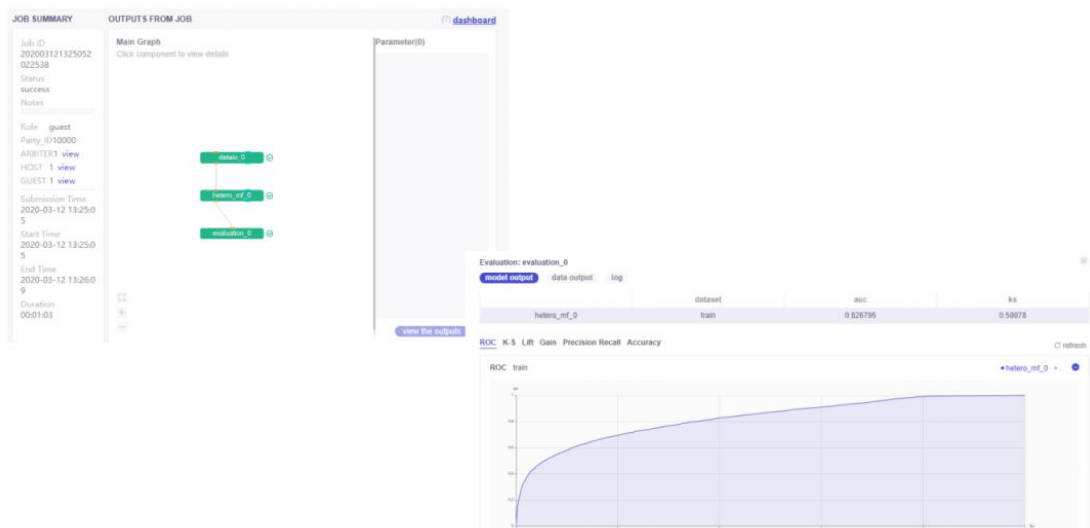
训练时的模块流程：Data IO – Intersection – Hetero MF - Evaluation

Hetero MF job DSL:

Sample Data:

```
# id,userId,itemId,rating
0,2031,5286,0
1,4368,5958,0
2,2174,4059,0
3,3815,5824,0
4,3831,3182,0
5,3359,3612,1
6,1094,3288,0
7,3573,3229,0
8,5391,4596,0
```

```
{
  "components": {
    "dataio_0": {
      "module": "DataIO",
      "input": { ... },
      "output": { ... },
    },
    "hetero_mf_0": {
      "module": "HeteroMF",
      "input": { ... },
      "output": { ... },
    },
    "evaluation_0": {
      "module": "Evaluation",
      "input": { ... },
      "output": { ... },
    },
  }
}
```

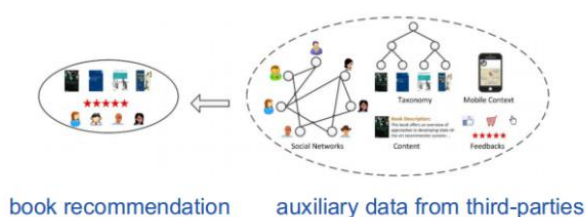


2. 纵向联邦推荐案例

在这个案例中，参与各方是有相同用户，但用户特征不一样。这个场景也是很常见的，一个推荐系统和另外一个数据提供方进行合作来提高推荐效果。



Example:
相同用户
不同用户特征



4	3		?	5	
5		4		4	
4		5	3	4	
	3				5
	4				4
		2	4		5

Location

Time

Georgia	2018.5
Florida	2019.1
Hawaii	2017.3
Kansas	2018.5
Georgia	2018.10
Florida	2019.9

Sports

Photography

Movie

Food

Y	N	N	N
N	N	N	N
Y	N	N	Y
Y	Y	N	N
N	Y	Y	N
N	Y	Y	Y

Party A

Party B

联邦因子分解机[Zheng et al. 2019]

解决方案: 因子分解机是处理交叉特征的常用算法。联邦因子分解机分别计算各参与方内部的交叉特征，以及跨参与方的部分交叉结果，并引入第三方在加密环境下汇总结果、协助传递梯度信息和更新模型。



交叉特征的好处：例如，“性别（男、女）x 电影类型（战争、爱情）”能够挖掘男性在战争电影、男性在爱情电影、女性在战争电影、女性在爱情电影上的不同偏好程度

Prediction function
$$f([\mathbf{x}_p^{(A)}; \mathbf{x}_q^{(B)}]) = f(\mathbf{x}_p^{(A)}) + f(\mathbf{x}_q^{(B)}) + \sum_{i,j} \langle \mathbf{v}_i^{(A)}, \mathbf{v}_j^{(B)} \rangle x_{p,i}^{(A)} x_{q,j}^{(B)}$$

Cross features in A

Cross features in B

Cross features between A and B

- Rendle 2012: Factorization Machines with libFM, in ACM Trans. Intell. Syst. Technol., 3(3), May.
- Zheng. 2019. Federated factorization machine. Tech Report WeBank.

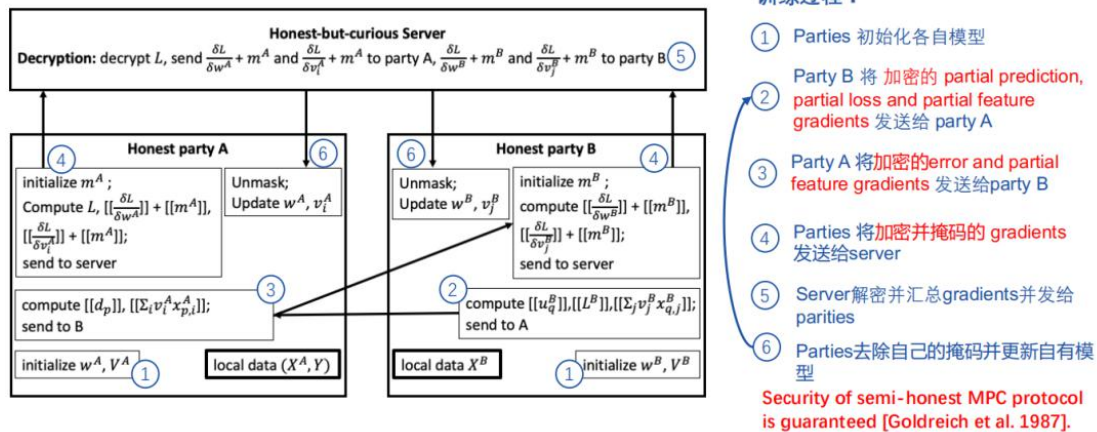
25

在数据可以自由传输的情况下，是很容易处理的。但是联邦场景下，如何进行交叉呢。联邦因子分解机的目标函数经优化后，由三部分组成：在 party A 和 party B 各自内部进行特征交叉，以及在 A 和 B 之间的特征交叉。我们分别在 A 方和 B 方做一部分计算，然后在合并起来，数据不出本地。同样的，通过引入第三方服务器，在加密的状态下在 A 方和 B 方之间传递模型参数和特征交叉求和的中间结果。

联邦因子分解机训练过程：



联邦因子分解机[Zheng et al. 2019]



Goldreich et al. 1987. How to Play ANY Mental Game. STOC, pages 218–229.

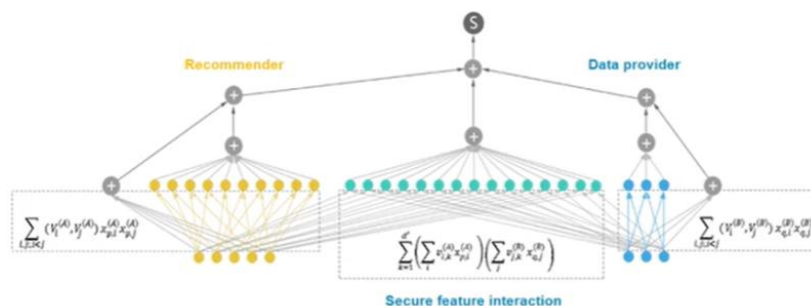
26

- ① Party A 和 Party B 分别初始化各自的模型
 - ② Party B 基于自己的特征计算部分预估值和部分 loss，并加密发送给 A
 - ③ A 方接收到以后，基于自己的特征计算部分预估值，并结合 B 方的预估值，计算最终的损失函数和梯度，然后将 B 需要的梯度和损失函数发送回 B
 - ④ A 和 B 在完成梯度计算以后，分别将梯度进行加密并掩码，发送给第三方服务器
 - ⑤ 第三方服务器解密并汇总梯度并发给 A 和 B
 - ⑥ A 和 B 去除自己的掩码并更新自有模型
- 不断循环，直至收敛，最终得到自己那部分的模型。

联邦因子分解机推理过程：



推理过程： 基于party A特征的加密结果 + 基于交叉特征的加密结果 + 基于party B特征的加密结果



1. Party A and B 计算中间结果并加密
2. Server 汇集加密的中间结果
3. Server将结果发送 party A

在推理过程中，对于新样本，需要 A 方和 B 方共同参与，才能完成模型的预估。

- ① A 和 B 首先基于自己的模型和自己的特征计算自己的特征交叉结果以及跨合作方的特征交叉结果，并加密发送给第三方服务器
- ② 第三方服务器汇集的中间结果
- ③ 第三方服务器将汇总结果发送给 A 方，解密后即可使用

联邦因子分解机在 FATE 中的实现：

联邦因子分解机的预测函数和损失函数如下图所示。特别的，将 AB 两方共同完成的交叉项分解为各自需要完成的计算过程。



联邦因子分解机的预测函数：

$$\begin{aligned}
 f([X_p^{(A)}; X_q^{(B)}]) &= f(X_p^{(A)}) + f(X_q^{(B)}) + \sum_{i,j} x_{p,i}^{(A)} x_{q,j}^{(B)} \\
 &= f(X_p^{(A)}) + f(X_q^{(B)}) + \sum_i \sum_j \sum_{k=1}^{d'} v_{i,k}^{(A)} v_{j,k}^{(B)} x_{p,i}^{(A)} x_{q,j}^{(B)} \\
 &= f(X_p^{(A)}) + f(X_q^{(B)}) + \sum_{k=1}^{d'} (\sum_i v_{i,k}^{(A)} x_{p,i}^{(A)}) (\sum_j v_{j,k}^{(B)} x_{q,j}^{(B)})
 \end{aligned}$$

联邦因子分解机的损失函数：

$$\begin{aligned}
 \ell([W^{(A)}; W^{(B)}], [V^{(A)}; V^{(B)}]) \\
 = \frac{1}{2nA} \sum_{p=1}^{nA} (y_p - f([X_p^{(A)}; X_q^{(B)}]))^2 + \frac{\alpha}{2} \Omega([W^{(A)}; W^{(B)}], [V^{(A)}; V^{(B)}])
 \end{aligned}$$

其中 $\alpha > 0$ 是超参数。

其核心逻辑有四个：

- 首先是实现预测函数；
- 然后计算 loss 和梯度；
- 发送给第三方服务器以及汇总，并更新；
- 分别发送给 ab 方，不断进行迭代。



联邦因子分解机在FATE中的实现

核心逻辑



联邦因子分解机在 FATE 中的使用：



训练时的模块流程：Data IO – Intersection – Hetero FM - Evaluation

Hetero FM job DSL:

Sample data:

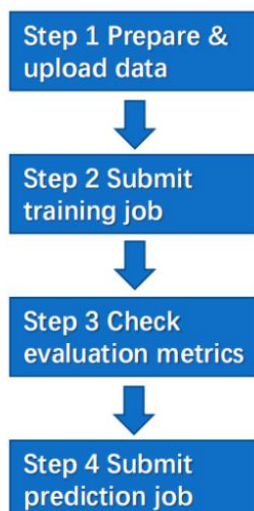
```
id,x0,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x
133,0.449512,-1.247226,0.413178,0.3037
0,-0.728193,-0.442587,-0.272757,-0.608
273,-1.245485,-0.842317,-1.255026,-1.0
3639,0.348620,-0.552483,-0.526877,2.25
175,-1.549664,-1.126219,-1.546652,-1.2
66881,-0.779358,-0.708418,-0.637545,0.
551,-0.851273,0.733108,-0.843535,-0.78
772,0.453852,-0.431696,-0.494754,-1.18
199,0.091654,0.216499,0.103839,-0.0346
```

```
{
  "components": {
    "dataio_0": {
      "module": "DataIO",
      "input": { ... },
      "output": { ... },
    },
    "intersection_0": {
      "module": "Intersection",
      "input": { ... },
      "output": { ... },
    },
    "hetero_fm_0": {
      "module": "HeteroFM",
      "input": { ... },
      "output": { ... },
    },
    "evaluation_0": {
      "module": "Evaluation",
      "input": { ... },
      "output": { ... },
    },
  }
}
```

① 参与方准备数据，数据是以矩阵的形式进行存储的，包括用户 ID，样本 ID，样本标签，每个特征的特征值。

② 定义好配置文件

Submit steps:



Host



Guest



③ Guest 提交训练任务，获得 jobid

```

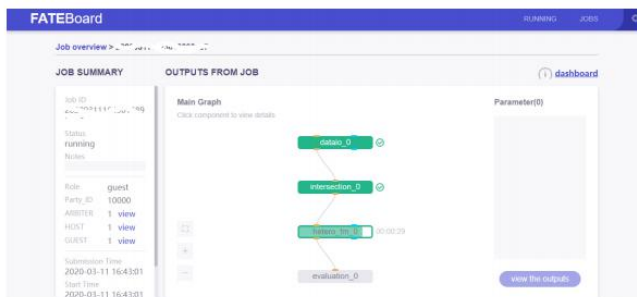
[app@PV_119_108_centos hetu:cm]$ python /home/app/standalone-fate-master-1.3.0/fate_flow/fate_flow_client.py -f submit_job -d test_hetu:cm_train_tenor
exit:0;job_id= test_hetu:cm_train_tenor_train_conf.json

{
  "status": {
    "submit_url": "http://10.107.150.10000:10000/hetui/dps/hetui/job_id=_____job_id=_____&quest=query&party=100000",
    "job_id_path": "/home/app/standalone-fate-master-1.3.0/job/_____.job_id.json",
    "job_runtime_conf_path": "/home/app/standalone-fate-master-1.3.0/job/_____.job_runtime_conf.json",
    "log_directory": "/home/app/standalone-fate-master-1.3.0/log/_____.log",
    "model_info": {
      "model_id": "writter-100000quest-100000model",
      "model_version": "_____"
    }
  },
  "job_id": "_____",
  "retcode": 0,
  "retmsg": "success"
}

[app@PV_119_108_centos hetu:cm]$

```

提交任务获取jobid

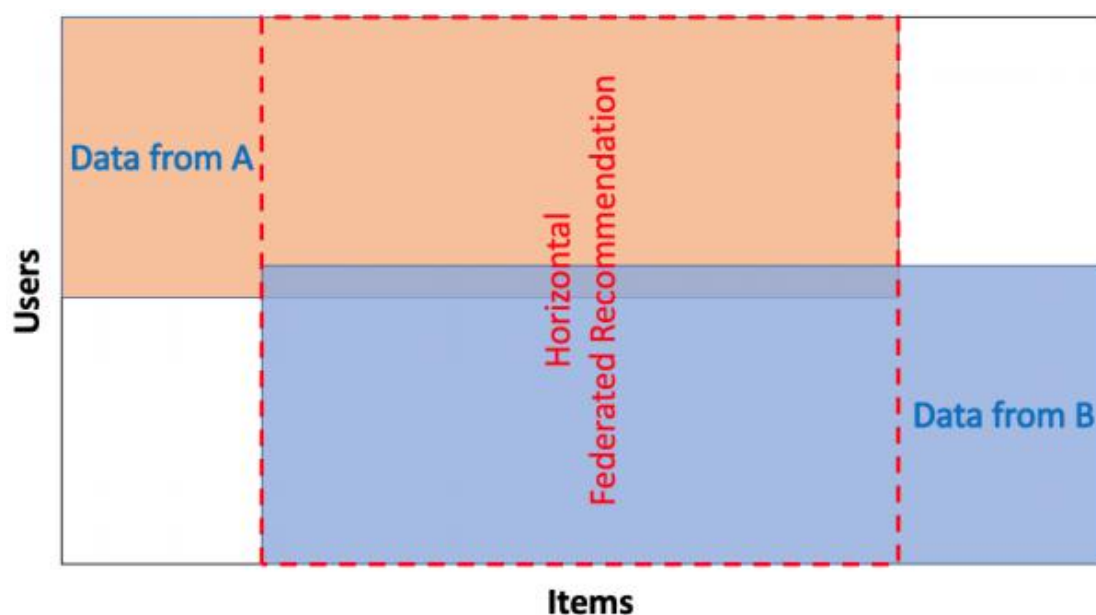


FATEBOARD中不同模块中的结果

④ 在 fate board 中查看中间结果，收敛情况，预估结果，预估准确率，并根据结果调整参数

3. 横向联邦推荐



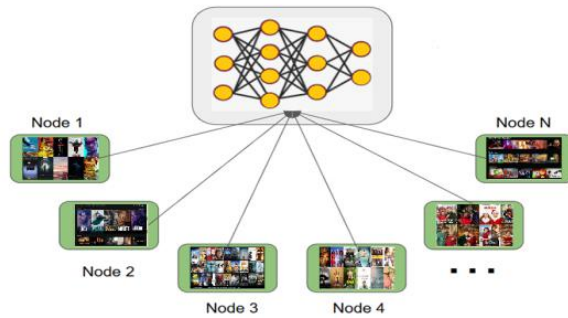


联邦双方有大量相同的内容/商品

横向联邦推荐是有大量相同的物品，但各方有不同的用户。比如不同地区的电影推荐系统中，内容是一样的，但用户群体不一样。每个参与方都是一个单独的用户，但推荐的电影是一样的。



Example:
相同内容/商品
不同用户或用户群

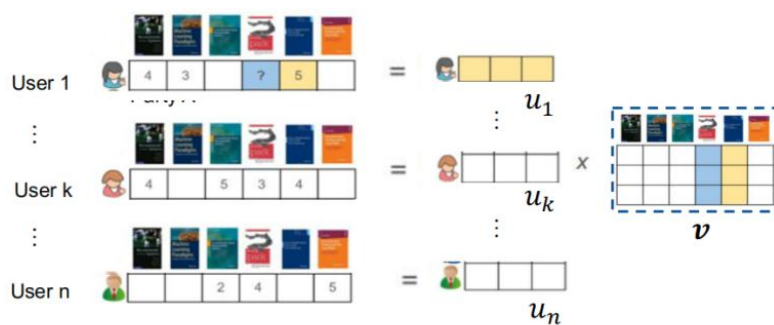


那么用户数据不出本地，如何构建推荐系统呢。

横向联邦矩阵分解：

横向联邦矩阵分解[Ammad et al. 2019]

解决方案: 1) 矩阵分解中的Item Profile由参与方共享；2) Item Profile梯度加密传输给第三方，并汇总各方更新；3) User Profile由各方自有并在本地更新



Loss function
$$\min_{U,V} \frac{1}{M} (r_{i,j} - \langle u_i, v_j \rangle)^2 + \lambda \|U\|_F^2 + \mu \|V\|_F^2$$

Update function

$$u_i^t = u_i^{t-1} - \gamma \nabla_{u_i} F(U^{t-1}, V^{t-1})$$

$\rightarrow$

User local updates

$$v_j^t = v_j^{t-1} - \gamma \nabla_{v_j} F(U^{t-1}, V^{t-1})$$

$\leftarrow$

Gradients from users

Server updates

Ammad-ud-din, et al. 2019. Federated Collaborative Filtering for Privacy-Preserving Personalized Recommendation System. arXiv:1901.09888.

34

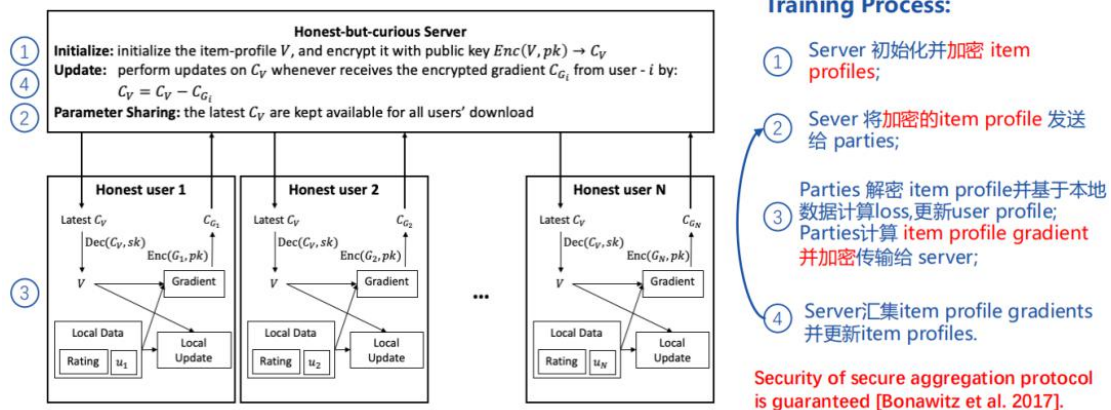


类似于纵向联邦矩阵分解，横向联邦矩阵分解是 item profile 共享。

横向联邦矩阵分解的训练过程：

横向联邦矩阵分解[Chai et al. 2019]

解决方案: 1) 矩阵分解中的User Profile由参与方共享；2) User Profile更新梯度加密传输给第三方，并汇总各方更新；3) Item Profile由各方自由并在本地更新



• Chai, et al. 2019. Secure Federated Matrix Factorization. arXiv:1906.05108.
 • Bonawitz et al. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. CCS, pages 1175–1191.

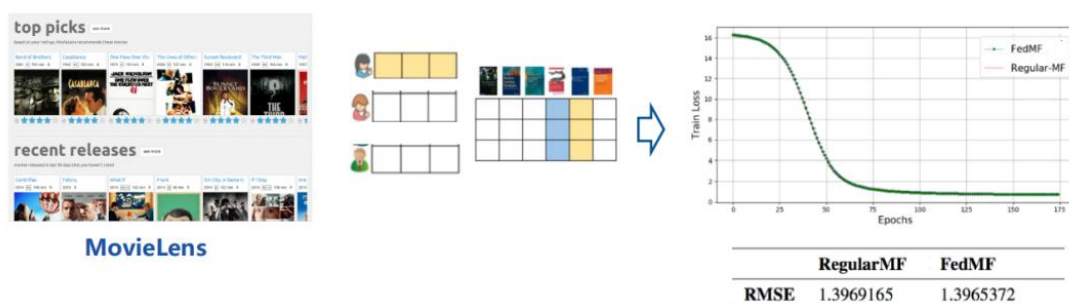
35

- ① 服务器初始化并加密 item profile
- ② 服务器将加密的 item profile 发送给各参与方
- ③ 参与方解密 item profile 并基于本地数据计算 loss，更新各自的 user profile，并将 item profile 的梯度加密传输给服务器
- ④ 服务器汇总 item profile 的梯度并更新 item profiles
- ⑤ 完成以后，每个参与方都会有各自的 user profile 和 item profile，然后利用矩阵分解进行推荐

4. 联邦矩阵分解与传统矩阵分解的效果对比



测试中，我们将联邦矩阵分解和明文中心化的传统矩阵分解算法进行比较，两者的收敛曲线一致



Chai, et al. 2019. Secure federated matrix factorization. arXiv:1906.05108.

36

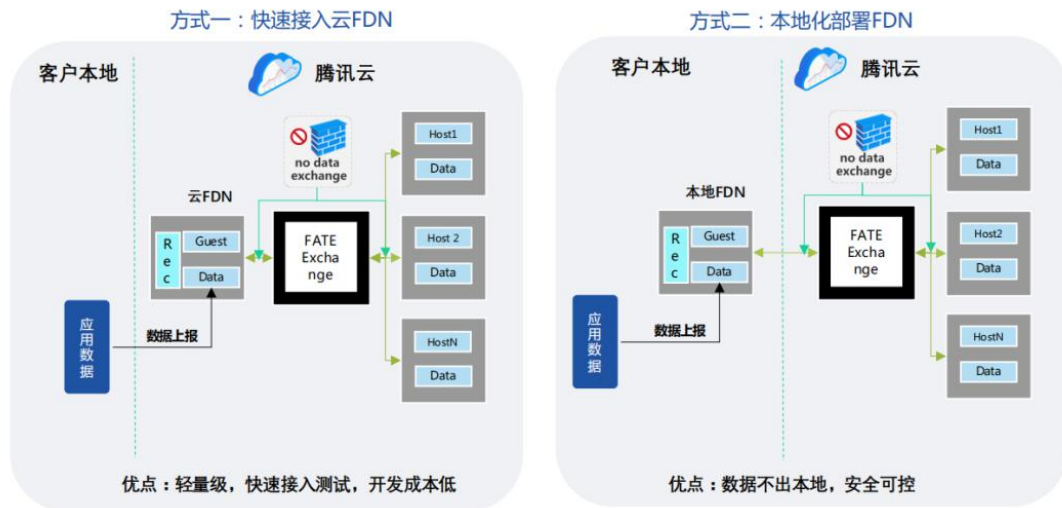
以 movielens 数据集为例，将联邦矩阵分解和明文中心化的传统矩阵分解算法进行比较。从下图中可以看出 training loss 的变化趋势基本是一致的，因此可以说联邦矩阵分解是无损的，是可以保护用户隐私的。

04

联邦推荐云服务和应用

1. 目前腾讯云上提供两种接入方式





<https://market.cloud.tencent.com/products/19083>

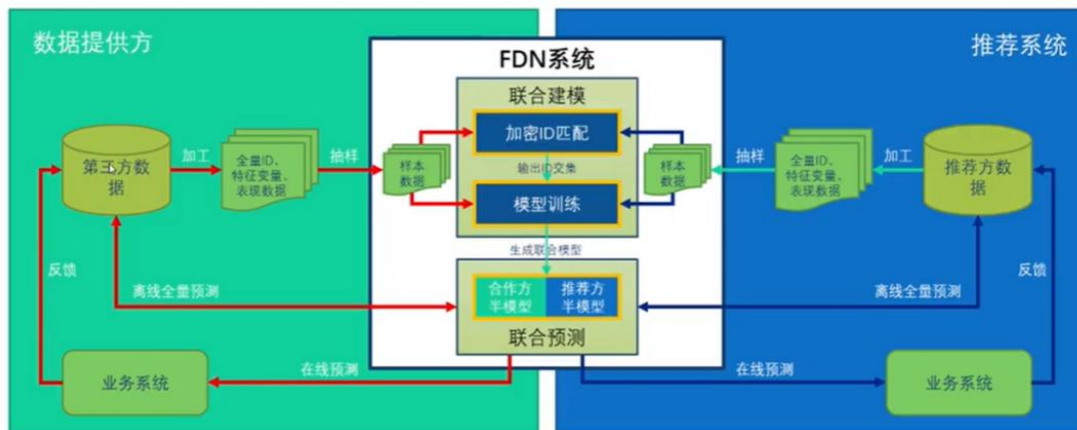
客户可以将自己的数据上传到腾讯云 FDN，在加密的环境下，和数据提供方进行联邦推荐模型训练，并最终保存到云服务上。然后就可以调用训练好的模型。优点是轻量级，快速接入测试，

另外一种是在本地部署 FDN，优点是数据不出本地，安全可控。将应用数据上传到本地 FDN 中，FDN 以加密的形式，与云服务和数据提供方进行联邦建模，最后将训练好的模型发送回本地 FDN 上，进行调用。

2. 联合建模预测示意图



联合建模、预测示意图 —— 安全合规的数据合作过程



注：客户ID包括但不限于客户身份证号码、手机号、设备ID (imei) 等；联合建模过程由拥有Y (表现数据) 的一方发起；

数据提供方和推荐系统分别处理自己的数据，然后进行特征加工，抽样到 FDN 系统中，FDN 会加密 id 匹配，并输出 id 交集或并集，进行模型训练。参与方和数据提供方会得到自己部分的模型，然后进行离线预测，评估效果或在线预测，不断的收集数据，不断的迭代和重复这个过程，优化模型。

3. 联邦推荐的应用

应用: 基于纵向联邦的推荐的内容推荐系统

联合第三方数据，在保证各方数据安全和用户隐私的情况下，构建推荐系统，提升各推荐指标



User's Internet browsing behaviors from 3rd-party



内容推荐

News Feeds Recommendation

PV	21%
UV	22%
CTR	11%



通过新闻推荐引擎和第三方数据提供方进行联邦学习，可以看到各种推荐指标都大幅提升。

应用：基于纵向联邦的企业金融服务

联合第三方数据，在保证各方数据安全和用户隐私的情况下，优化用户画像、制定个性化的推送策略



联合第三方数据，在保证各方数据安全和用户隐私的情况下，可以做到更精确的用户分层。根据用户的数据，优化用户画像，判断是否对利润有不同的偏好，从而制定个性化的推送策略。最终提高了 30% 的性能。

05

总结

推荐系统是数据驱动的应用，数据对效果的提升显著。在现实中，数据孤岛、隐私保护和数据安全是亟需解决的问题。微众银行首创联邦推荐，并将其应用于多个实际场景中取得了良好的效果。



联邦推荐算法 Github 地址：

<https://github.com/FederatedAI/FATE/tree/master/federatedrec>

今天的分享就到这里，谢谢大家。



机器学习中的隐私保护



分享嘉宾：郭夏玮 第四范式 资深研究员
编辑整理：蒋瑞尧
内容来源：第四范式 | 先荐
出品平台：DataFun
注：转载请在后台留言“转载”。

导读：近年来，随着 GDPR 通用数据保护条例出台以及一些互联网公司数据隐私泄漏等事件的发生，数据隐私的保护问题在行业应用中备受关注。与数据密切相关的机器学习算法的安全性成为一个巨大挑战。本文将介绍在机器学习领域中数据隐私安全的相关工作，并介绍第四范式在差分隐私算法效果提升上所做的工作。

主要和大家分享数据隐私的 3 方面：

- 隐私保护的问题与案例



- 基于数据的隐私保护技术：数据匿名化
- 机器学习模型训练中的隐私保护技术：差分隐私

■ 机器学习中的隐私保护问题

1. 信息隐私

信息隐私（Privacy）：指的是当一个组织内敏感数据被拥有权限的人员所使用于某些技术、过程（如数据分析、训练模型）时，对数据敏感信息进行保护的过程与规则。

数据的隐私（Privacy）与安全（Security）并不等价：有的时候很多人提到数据隐私时，会与数据安全混为一谈，但其实两者并不等价。**数据安全**通常指防止**数据被非法访问**；而**数据隐私**则一般指在**数据被合法访问**时，防止其中的敏感信息被访问者以某些方式"逆向"获取，避免因数据被"逆向"推导出而造成的敏感信息泄露和滥用。当然，对于企业来说，数据隐私和数据安全都非常重要。

2. 信息隐私问题

如今，在很多需要用到私人敏感数据的领域中，都存在数据隐私的问题。特别地，当机器学习技术应用在一些个人隐私数据上时，可能会暴露敏感数据，对个人造成很多负面影响。

虽然现实生活中有很多数据隐私的问题，但**严格意义上，个人隐私不可能被绝对保护**。



1977 年，统计学家 Tore Dalenius 给出关于数据隐私的严格定义：**攻击者不能从隐私数据里获取任何在没有拿到数据之前他们所不知道的个人信息。**

2006 年，计算机学者 Cynthia Dwork 证明上述定义的隐私保护是不存在的。有一个直观例子可以帮助理解：假设攻击者知道 Alice 的身高比 Lithuanian 女性平均身高高 2 英寸，这样攻击者只需要从一个数据集里获得 Lithuanian 女性身高平均值（在接触数据前攻击者并不知道），就能准确获得 Alice 的精确身高，甚至 Alice 都不需要在这个数据集里。因此，对于一份有信息量的数据，不可能完全不暴露隐私信息。

3. 隐私泄露的危害

隐私泄露的危害，主要有：

- 隐私信息被应用于欺诈与骚扰：如盗刷信用卡、电话诈骗、冒用身份等
- 用户安全受到威胁：用户信息受到泄露，引起更多有目的犯罪
- 非法机构利用用户隐私信息操控用户
- 用户信任危机
- 违背相关法律

不过一般来说，对于不特别极端的情况，我们仍可以从很大程度上来保证数据在机器学习过程中不被泄露。

这里再分享一个隐私泄露的案例：



1997 年，马萨诸塞州 GIC 对外公布了一组医疗数据，其中包括病人的 5 位邮编、性别、生日。州长 Weld 对公众保证这份数据是匿名的，因为标识特征（如名字地址）已经被移除。结果一位 MIT 毕业生通过结合这份数据与她花了 20 美元购买的选民登记数据，找到了 Weld 的医疗记录。同时，她还宣布 87% 的美国公民可以通过 5 位邮编、性别、生日进行唯一标识。

这个真实案例印证了：**任何有一定信息量的数据都有可能产生隐私泄露；单纯的数据匿名方法是很难保护隐私的。**接下来，我们来看一下，针对潜在的隐私泄露问题我们有哪些技术可以在很大程度上杜绝隐私的泄露。

■ 数据匿名化技术（Data Anonymization）

数据匿名化是一个**从数据层面上**进行隐私保护的技术。匿名化很多时候是通过 hash 掉"姓名"等关键标识符来实现的。而在只简单 hash 掉姓名的情况下，数据访问者有很多方法通过利用其它特征的信息来反推出某人在数据表中对应的那一条数据。这时候，就需要一种技术来防止访问者能够通过结合多个特征的数据来确定人与数据间的对应关系。

1. 数据表中数据列的类别（按所含隐私量分类）



Key Attribute	Quasi-identifier			Sensitive attribute
Name	DOB	Gender	Zipcode	Disease
Andre	1/21/76	Male	53715	Heart Disease
Beth	4/13/86	Female	53715	Hepatitis
Carol	2/28/76	Male	53703	Brochitis
Dan	1/21/76	Male	53703	Broken Arm
Ellen	4/13/86	Female	53706	Flu
Eric	2/28/76	Female	53706	Hang Nail

图片来自: <https://elf11.github.io/2017/04/22/kanonymity.html>

- **标识列 (Key Attribute , 简称 KA)** : 单凭其一系列便能直接锁定某人的数据列 , 如 : 身份证号、姓名 (大多数情况下) ;
- **半标识列 (Quasi-identifier , 简称 QID)** : 无法单凭其一系列确定对象 , 但通过结合多列数据或查其它信息 (缩小范围) 可以对应到个人的数据列 , 如 : 生日、邮编等 ;
- **敏感信息列 (Sensitive Attribute , 简称 SA)** : 如疾病、收入等。

虽然只有标识列的数据能够直接缩小范围到单个样本 , 但如前文所述 , 仅匿名化标识列数据对于隐私保护是完全不够的。

2. 攻击方法与防护方法

① 链接攻击



数据匿名化技术

链接攻击(linkage attack)

- 攻击者通过其他表获取敏感信息

ID	QID			SA
Name	Zipcode	Age	Sex	Disease
Alice	47677	29	F	Ovarian Cancer
Betty	47602	22	F	Ovarian Cancer
Charles	47678	27	M	Prostate Cancer
David	47905	43	M	Flu
Emily	47909	52	F	Heart Disease
Fred	47906	47	M	Heart Disease

Name	Zipcode	Age	Sex
Alice	47677	29	F
Bob	47983	65	M
Carol	47677	22	F
Dan	47532	23	M
Ellen	46789	43	F

Paradigm | 先研 Copyright © 2019 4Paradigm All Rights Reserved.

链接攻击 (Linkage Attack) : 通过获取其它渠道的信息 (如右表) 来锁定 (见左表) 当前数据表中数据所对应的个人。

② K-Anonymity

数据匿名化技术

K-Anonymity

- 对于一份数据, 要求其发布的版本内, 对于每一条记录, 至少有 $k-1$ 条其他记录的半标识属性与其一致

QID			SA
Zipcode	Age	Sex	Disease
47677	29	F	Ovarian Cancer
47602	22	F	Ovarian Cancer
47678	27	M	Prostate Cancer
47905	43	M	Flu
47909	52	F	Heart Disease
47906	47	M	Heart Disease

QID			SA
Zipcode	Age	Sex	Disease
476**	2*	*	Ovarian Cancer
476**	2*	*	Ovarian Cancer
476**	2*	*	Prostate Cancer
4790*	[43,52]	*	Flu
4790*	[43,52]	*	Heart Disease
4790*	[43,52]	*	Heart Disease

Paradigm | 先研 Copyright © 2019 4Paradigm All Rights Reserved.



K-匿名 (K-Anonymity)：针对链接攻击的防护方法。对于每一份数据的各版本内的**每一条记录**，规定至少有 $K-1$ 条**其它记录**的**半标识属性**与**其一致**。

上方右表就是对上方左表做"3-匿名"之后的结果：我们可以发现，在统一对"zipcode"、"age"、"sex"三列的数据的末尾作模糊（相当于匿名化）处理后，含有"zipcode : 47677** , age : 2* , sex : *"的信息的数据共有 3 条（这 3 条记录中的任一条，均满足至少有 $3-1=2$ 条其它记录的半标识属性与其一致）；左表原数据中的另外 3 条经变换后也满足 3-匿名的条件。

左表在 K-Anonymity 之后，虽然数据会有所损失，但保证了每条数据中的敏感信息无法与个人一一对应，降低了隐私泄露的风险。如何在尽可能小的数据损失的情况下做 K-Anonymity 也是一个数据隐私的研究方向。

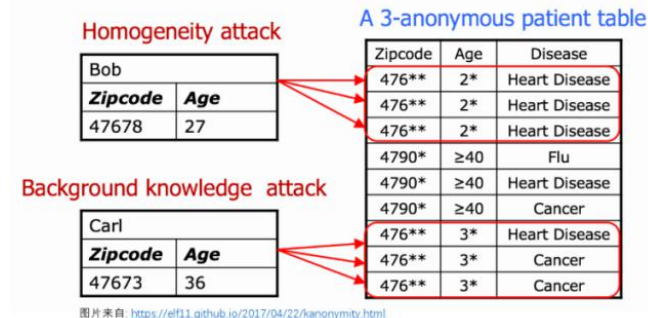
③ Homogeneity attack



数据匿名化技术

针对K-anonymity攻击 (Homogeneity attack)

- 通过找到一组模糊化数据的共同点来获得敏感信息



图片来自: <https://elf11.github.io/2017/04/22/k-anonymity.html>

4Paradigm | 先荐 Copyright © 2019 4Paradigm All Rights Reserved.

同质化攻击 (Homogeneity Attack)：可克制 K-Anonymity 防护的攻击。某些情况下，可通过**直接对比其它列信息**或**查找其它外部资料**发现，K-Anonymity 中半标识列均相同的（至多）K 条数据的**其他列的信息其实是同质的**，因而确定下来某人的隐私数据。

上方右表说明：经对比可发现 Bob 必然患有 Heart Disease；而假如能够另通过查资料发现 Carl 所在地人们普遍患有 Heart Disease，就还可以推断出 Carl 大概率是患 Heart Disease 的那例样本。

④ L-Diversity



数据匿名化技术

• L -Diversity

- 对满足 K -Anonymity 的发布数据，每一个等价类中敏感信息至少取 L 个不同的值

Caucas	787XX	Flu
Caucas	787XX	Shingles
Caucas	787XX	Acne
Caucas	787XX	Flu
Caucas	787XX	Acne
Caucas	787XX	Flu
Asian/AfrAm	78XXX	Flu
Asian/AfrAm	78XXX	Flu
Asian/AfrAm	78XXX	Acne
Asian/AfrAm	78XXX	Shingles
Asian/AfrAm	78XXX	Acne
Asian/AfrAm	78XXX	Flu

图片来自: <https://elfs1.github.io/2017/04/22/k-anonymity.html>

4Paradigm | 先荐 Copyright © 2019 4Paradigm All Rights Reserved.

L-散度 (L -Diversity) :对 K -Anonymity 的改进。在满足 K -Anonymity 的基础上，要求所有**等价类**（即各半标识列信息均相同的一些样本）中的**敏感信息**（如：病史中的 Heart Disease、Flu）的**种类/取值**至少有 L 种。上表就是一个满足 3-散度的数据。

⑤ 相似性攻击



数据匿名化技术

- 相似性攻击
 - Bob收入在20k-40k
 - Bob有胃部疾病

Similarity attack

Zip	Age
47678	27

A 3-diverse patient table

Zipcode	Age	Salary	Disease
476**	2*	20K	Gastric Ulcer
476**	2*	30K	Gastritis
476**	2*	40K	Stomach Cancer
4790*	≥40	50K	Gastritis
4790*	≥40	100K	Flu
4790*	≥40	70K	Bronchitis
476**	3*	60K	Bronchitis
476**	3*	80K	Pneumonia
476**	3*	90K	Stomach Cancer

图片来自 <https://edf11.github.io/2017/04/22/kanonymity.html>

Paradigm | 先荐 Copyright © 2019 4Paradigm All Rights Reserved.

相似性攻击 (Similarity Attack): 可对满足 L-Diversity 的数据进行攻击。从外界获取个人多方面的背景信息，缩小范围以锁定目标。

⑥ T-Closeness

数据匿名化技术

- T-Closeness
 - 为L-Diversity的拓展，对于满足K-anonymity的发布数据，每一个等价类中敏感信息的分布与数据集总体的敏感信息分布距离（如KL距离）小于T

Caucas	787XX	Flu
Caucas	787XX	Shingles
Caucas	787XX	Acne
Caucas	787XX	Flu
Caucas	787XX	Acne
Caucas	787XX	Flu
Asian/Am	78XXX	Flu
Asian/Am	78XXX	Flu
Asian/Am	78XXX	Acne
Asian/Am	78XXX	Shingles
Asian/Am	78XXX	Acne
Asian/Am	78XXX	Flu

Paradigm | 先荐 Copyright © 2019 4Paradigm All Rights Reserved.



T- 保密 (T-Closeness)：对 L-Diversity 的拓展。对于满足 K-Anonymity 的数据，规定每一个等价类中的**敏感信息的分布与数据集整体的敏感信息分布的距离**（可自行定义，常见的有：K-L 散度）小于 T。

■ 差分隐私 (Differential Privacy) 技术

除了数据匿名化不当以外，由数据和特定建模方法得到的模型同样有隐私泄露的风险（被攻击获取到原数据中的隐私信息甚至原数据本身）。差分隐私在机器学习模型的**建模过程**中应用较广。

1. 模型的隐私保护风险

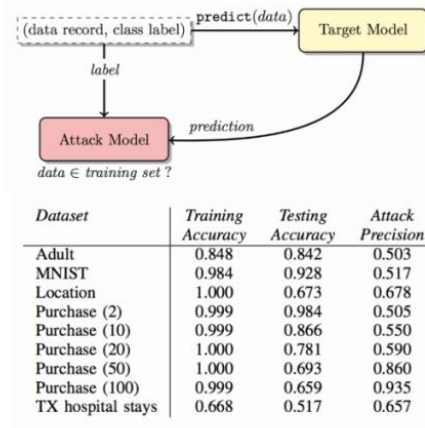
未经过隐私保护处理的机器学习模型理论上可能暴露训练数据里的敏感信息。

Membership inference attack :



模型的隐私保护风险

- Membership inference attack
 - 探测一个样本是否属于训练集：训练集泄露
 - (Shokri, Reza, et al. "Membership inference attacks against machine learning models." *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017.)



Paradigm | 先荐 Copyright © 2019 4Paradigm All Rights Reserved.

成员推断攻击 (Membership Inference Attack)：探测一个给定样本是否属于训练某"待破解"模型 (target model) 的**原数据集**。

步骤 (参考论文见上图)：

① **模拟生成和训练 target model 的原数据集分布类似的一系列数据集 (即 shadow datasets)**，用与训练 target model 相同的训练算法 (训练算法是什么很容易得知) 去训练出对应的一系列的模型，这些模型称为 "shadow models"，这些 shadow models 和 target model 的特征分布是很像的；

② **Attack Model 的训练**。数据集构成：对于每一个 shadow datasets 中**每一条数据**，可以构建**此条数据在某个 shadow model 上的预测结果**、**该数据真实 label 作为训练样本**，将此条数据是否属于之



该 **shadow model** 作为标签。通过构建的数据集再即可训练 **Attack Model**，能够判断任意一条给定的数据是否属于原数据集；

③ 对于一条数据，对于该数据 **target model** 的预测结果和真实 **label** 作为输入，通过判断模型就可以判断出该数据是否属于 **target dataset** 的。

右图为上述思路的一例实际应用，可以看出如果对训练模型的方法不加处理的话，原数据集被成员推断攻击法破解的程度还是很高的。

Model Inversion Attack :

模型的隐私保护风险

- 论文Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures中，作者通过一个人脸识别模型与一个名字，反向恢复了一张人脸（左图），作为对比，右图人脸为数据集里的真实样本



模型逆向攻击 (model inversion attack)：是指攻击者通过模型与某个样本的其他特征，直接推断某个敏感特征值。

2. 差分隐私技术 (Differential Privacy)



如果对于任两个只相差 1 个样本的数据集 D_1 、 D_2 ，在通过过程 M 建模后，产生的模型与给定模型 t 完全相同的概率之比不大于 e^ϵ （其中 ϵ 为某非负数，为**隐私预算**），那么我们说建模过程 M 是 ϵ -差分隐私的。

推荐系统学院

差分隐私技术(Differential Privacy)

- 差分隐私定义了一个处理过程（如查询、建模）中，输入数据的变动对结果的影响越小，隐私保护越好

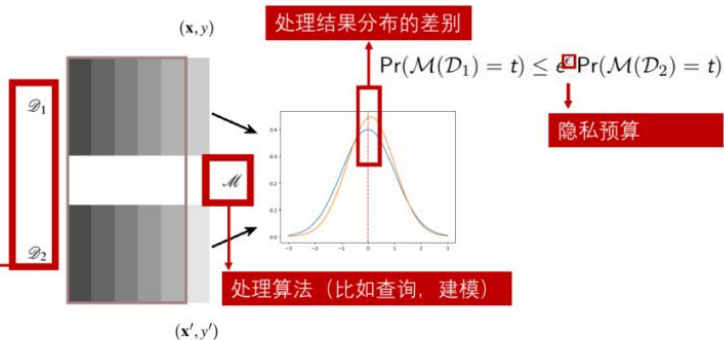
- 可用于对外数据查询接口

Definition (Dwork et al., 2006)

A randomized mechanism $\mathcal{M}$ is ϵ -differentially private if for all output t of $\mathcal{M}$ and for all input data D_1, D_2 differing by one element,

$$\Pr(\mathcal{M}(D_1) = t) \leq e^\epsilon \Pr(\mathcal{M}(D_2) = t).$$

两个数据集只差一个样本



Paradigm 先荐 Copyright © 2019 4Paradigm All Rights Reserved

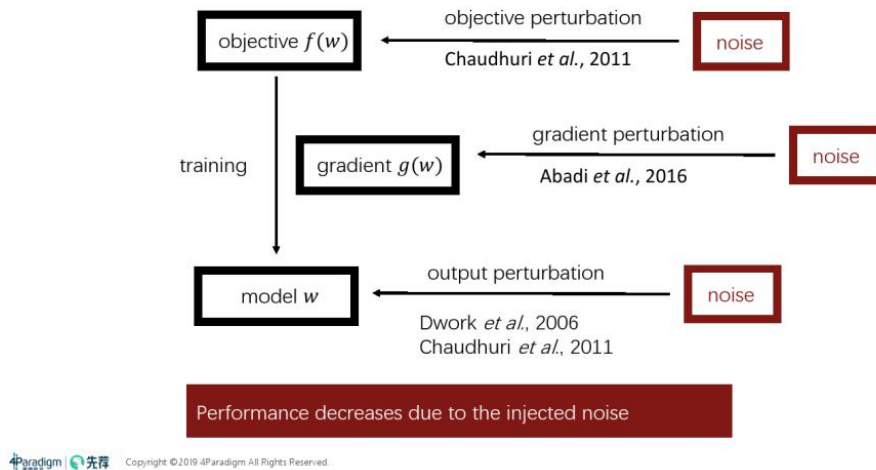
理解：如果建模过程 M 是 ϵ -差分隐私的，且 ϵ 很小，那么数据集和其训练出来的模型关系很小，即两个相差有固定上限的数据集训练出来的模型是几乎一样的，所以拿到模型的访问者**即便知道大致数据分布也并不能推断出原数据集**；反之，如果 ϵ 很大，那么两个相差有固定上限的数据集训练出来的模型可能天差地别的，那么这种情况下，原数据集与得出的模型间一一对应的比例很高，**知道训练出来的模型就很有可能能够通过模型的分布倒推出原数据集**。也就是说，如果改变一条的数据对最终得到的模型影响很小，那么数据集隐私暴露的风险就很小，反之则很大。



① 差分隐私下的模型训练

差分隐私下的模型训练

推荐系统学院



- **对目标函数添加一定的噪声**：训练出来的模型会带有一定的随机性，可证明这样的随机性可满足差分隐私。
- **在梯度上加噪声**：深度学习比较常用，因为对于深度学习模型，如果把噪声加在目标函数或输出上，差分隐私的分析会很复杂。
- **在模型输出上加噪声**：使得不一样的原数据集输出的模型差异变小。



差分隐私下的模型训练

- 目标扰动算法
- 差分隐私算法会降低算法效果
- ϵ -差分隐私保障
- 在达到同样的泛化误差情况下, sample complexity 为: $O(\frac{d \log(d)}{\epsilon})$, d 是特征维度

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^n \ell(\mathbf{w}^\top \mathbf{x}_i, y_i) + \lambda g(\mathbf{w})$$

↓ objective perturbation

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^n \ell(\mathbf{w}^\top \mathbf{x}_i, y_i) + \underbrace{\mathbf{b}^\top \mathbf{w} / n}_{\text{noise term}} + \underbrace{\Delta \|\mathbf{w}\|^2 / 2}_{\text{correction term}} + \lambda g(\mathbf{w})$$

$h(\mathbf{b}) \propto \exp(\epsilon' / \|\mathbf{b}\|)$
 ϵ' modified privacy budget

Δ is a scalar depending on λ, n, ϵ

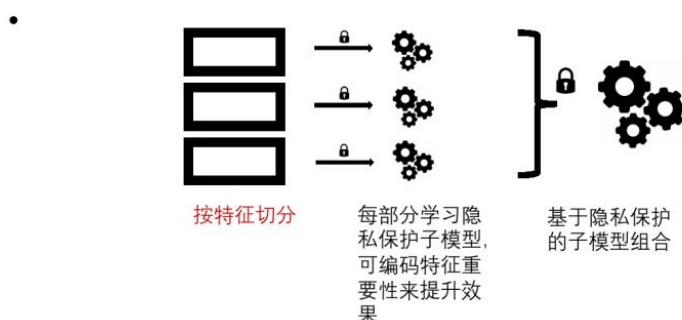
4Paradigm | 先荐 Copyright © 2019 4Paradigm All Rights Reserved.

不过，作为代价，数据隐私保护得越好，算法的性能下降得越多。在训练样本复杂度相同的情况下，隐私保护越好（ ϵ 越小），泛化误差越大。

② 基于特征切分的差分隐私算法

基于特征切分的差分隐私算法

- X. Guo, Q. Yao, J.T. Kwok, W. Tu, Y. Chen, D. Wen, Q. Yang. Differential private stack generalization with an application to diabetes prediction. **International Joint Conference on Artificial Intelligence (IJCAI)**, Macao, China, Aug 2019.



24

4Paradigm | 先荐 Copyright © 2019 4Paradigm All Rights Reserved.



按特征来切分做差分隐私，而非做基于样本切分的差分隐私：

数据先按样本切分为上下 2 份，其中一份按特征切成 K 份去学 K 个子模型，这些子模型再在剩下的一半样本上用预测值去融合出新的（第二层的）K 个子模型，其中切分和融合都是用差分隐私去做的。可证明最后总的输出是满足差分隐私的。

按特征切分的原因：

推荐系统学院

- 达到损失 ϵ 所需要的样本数量复杂度（越小效果越好）

$$O\left(\frac{d \log\left(\frac{d}{K}\right)}{\epsilon} \frac{1}{K q_k}\right)$$

- D 是特征维度，K 是切分份数，q 为对应的特征重要性
- 我们算法考虑了特征重要性，在理论隐私保障下提升了算法效果

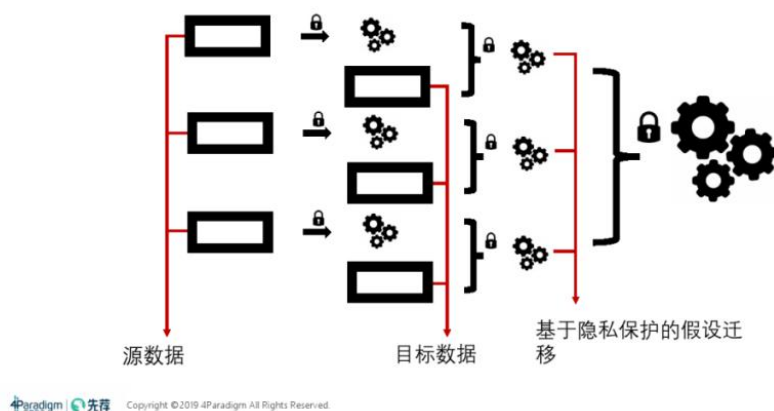
Paradigm | 先知 Copyright © 2019 Paradigm All Rights Reserved

仅保证差分隐私是不够的，还需保证算法的性能。由上图公式可发现，给重要的特征分配更多隐私预算，训练样本复杂度会减小，训练出的子模型效果会更好。

③ 在迁移学习中的应用



- 基于特征切分的差分隐私算法（迁移学习）



按②中所述方法做特征切分，在假设迁移部分对子模型做差分隐私保护，最后做含差分隐私的融合。

今天的分享就到这里，谢谢大家。



关系图谱在贝壳找房风控体系的应用与实践



分享嘉宾：王学志 贝壳 资深工程师

文章整理：李光明

内容来源：贝壳找房知识图谱技术大会

出品平台：DataFun

温馨提示：疫情期间，勤洗手，戴口罩。

导读：



贝壳业务模式

- ACN (Agent Cooperation Network) 经纪人合作网络

打破跨品牌间房源、用户、经纪人之间的联动难题，建立合作共赢 的良性生态，实现更加高效的资源配置。



风控是与具体业务强相关的技术领域，贝壳的 ACN(Agent Cooperation Network，经纪人合作网络) 业务模式旨在打破跨品牌间房源、用户、经纪人之间的联动难题，建立合作共赢的良性生态，实现更加高效的资源配置，并在此业务模式下，逐步建立了经纪人，房源，客源间的利益联盟和信任联盟，以及为经纪人服务的尊严联盟，品质联盟和网络联盟。围绕贝壳的 ACN 业务模式，从以下三方面介绍贝壳的风控体系：

- 贝壳风控的业务概况
- 关系图谱在贝壳风控的应用与实践
- 关系图谱的规划

■ 贝壳风控的业务概况

首先分别介绍贝壳的风控业务风险，风控特点以及对应的风控体系。



1. 贝壳的业务风险

贝壳业务风险



6 2019 KE.COM ALL COPYRIGHTS RESERVED

- 虚假房源/客源/带看带来的“虚假繁荣”，其短期影响有限，但长此以往，会损害平台的信誉度，严重影响用户体验，从而导致客源及房源的流失，降低平台吸引力；
- 爬虫导致的资源泄露，以及经纪人违规的体外行为（绕过平台促进成单），都存在损害平台收益的风险；
- 藏房，藏客以及低费率等经纪人间的恶性竞争有违贝壳的平台规则，给平台健康生态的构建带来了风险。



2. 贝壳的风控特点

风控—不同场景的特点

贝壳



7 2019 KE.COM ALL COPYRIGHTS RESERVED

业界的风控大多体现在线上场景，而贝壳多元化的业务场景中：

贝壳风控特点

贝壳



8 2019 KE.COM ALL COPYRIGHTS RESERVED

涉及了线上+线下的多个环节，给风控带来了额外的高难度，而在整个环节的长链条中混杂的各种小 B (business) /大 B 的违规行为，更加剧了平台的加盟商风险和经纪人风险，这就要求贝壳的风控体系，不仅能够识别风险，



同时还需要提供可解释的，完备的证据链。尤其是贝壳特有的**低频、大额、长周期**的交易属性，有别于业界普遍存在的**高频、小额、短周期**的场景，是风控体系的重点与难点。

3. 贝壳的风控体系

结合业务场景和风控特点，贝壳构造了如下图所示的层级风控体系：



最底层是数据能力层，提供包括经纪人数据，房产/地产数据，门店数据以及经纪人行为数据在内的全量的业务相关数据；其上层为核心技术层，利用关系图谱挖掘风险关系，同时提供经纪人风险标签，门店风险标签以及城市的风险罗盘；最后的能力输出层可分为事前、事中和事后三个环节：事前环节，主要在门店/经纪人准入阶段发挥作用，确保“我是我”的经纪人/门店真实性和有效性；事中环节主要通过管控经纪人的线上实时行为进行风险刻画，同时会构建经纪人的风险罗盘，便于了解风险态势的变



化；而事后环节可以通过举报，系统主动识别等方式找到问题门店/经纪人，并通过关联关系提供完备的证据链。

关系图谱在贝壳风控的应用与实践

1. 为什么用关联关系

WHY | 大家为什么用



11 2019 KE.COM ALL COPYRIGHTS RESERVED

人力介入程度从高到低的风控手段依次包括黑名单机制，专家规则，有监督的模型，关联图谱以及行为序列分析五个层次，其中黑名单机制，专家规则都需要较高的人力成本，而且从行业现状来看，很难构建一份完整的业界黑名单，从而导致黑名单机制的可实施性较差，而专家规则过度依赖专家经验，存在较高的经验误判风险，主观犯错概率较大，因此通过黑名单或专家规则的方式，很难构建切实可行的风控体系，而监督模型方法的难点在于样本的定义与构造，因此关联图谱和行为序列分析这种自动化识别风险的方法吸引了越来越多的关注。



WHY | 贝壳风控为什么用

➤ 从危害程度和处理上，**大B风险为优先处理的重点**

➤ 大B的违规和复杂的关联关系强相关



12 2019 KE.COM ALL COPYRIGHTS RESERVED

贝壳之所以使用关联图谱的出发点还是出于贝壳业务中常见的小 B 风险和大 B 风险的权衡考虑。大 B 风险具有隐蔽性差，团伙规模大，单量高以及查处成本低等特点，在危害程度上高于小 B 风险，因此需要重点控制大 B 风险。另外，从针对大 B 的违规行为中分析发现，与关联关系强相关的违规行为占比达到 29.3%，因此关联关系在贝壳风控体系中的作用，总体上可以表述为先通过关联关系或举报发现高风险的违规行为，而后可利用关联关系识别中低风险的违规行为，进而实现对“黑”，“灰”，“白”三个不同层次违规行为的监控和打击。



2. 贝壳的关系图谱

贝壳关系图谱演进

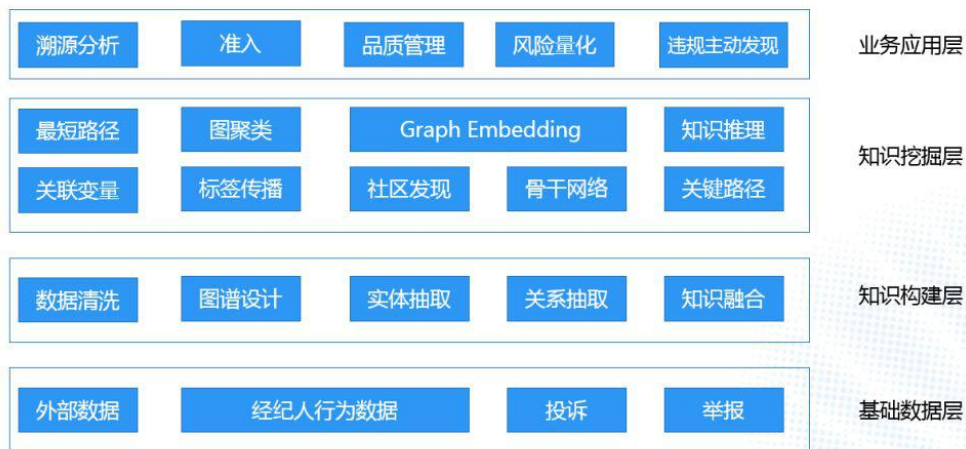
贝壳



贝壳风控的关系图谱经历了事实图谱 -> 推理图谱 -> 图谱融合三个阶段的演进。事实图谱涵盖了贝壳找房所有线上的动作和行为，形成了 10 亿级节点，100 亿级边的巨大图谱；推理图谱构建在事实图谱基础上，利用不同节点间的关系，分别构建行为图谱，社交图谱，作业图谱和工商图谱，这也是贝壳风控目前所处的阶段；而未来的图谱融合阶段，会利用 ID 打通，完成多个推理图谱的关系融合，进而完成人与人之间亲密度的定性或定量的表达，从而实现深层次的风险控制。



贝壳关系图谱整体架构

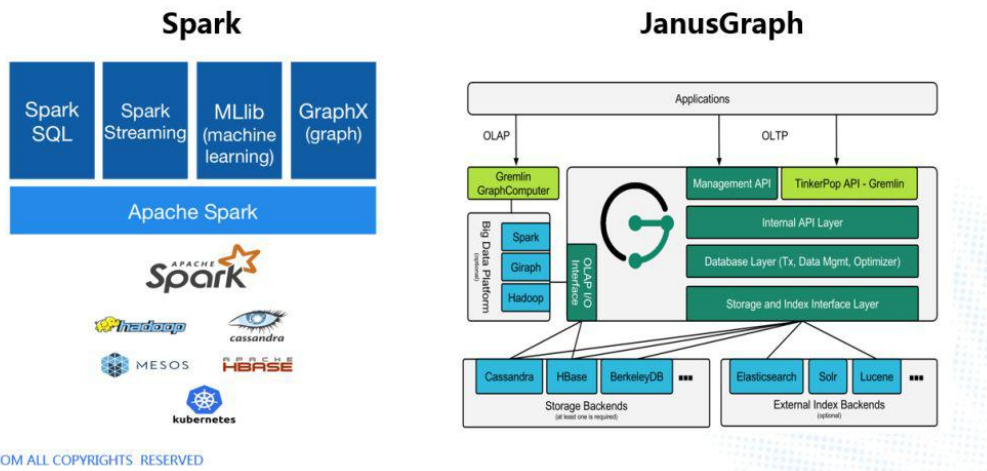


14 2019 KE.COM ALL COPYRIGHTS RESERVED

贝壳关系图谱的整体架构主要包括四层：基础数据层负责收集各种来源的行为和属性数据；知识构建层通过多种手段抽取实体和关系，完成构图；知识挖掘层会结合传统的最短路径/关键路径方法，以及社区发现，标签传播和 Graph Embedding 等机器学习方法，进一步挖掘节点之间的关系；业务应用层基于关联图谱提供了溯源分析，风险量化，违规行为主动发现等业务能力。



贝壳关系图谱技术选型



贝壳的关联图谱技术选择了 Spark GraphX 作为图分析的工具，Janus Graph 作为图查询的工具，之所以选择 GraphX 主要是考虑 Spark 拥有较完备的社区支持，并且 GraphX 实现了 Graph 在 Spark 上的重写，具有较高的效率；而选用 Janus Graph 主要是出于业务场景中较多的图的可视化以及溯源分析的需求考虑。



3. 贝壳关系图谱的应用

贝壳

关系图谱在贝壳风控的应用



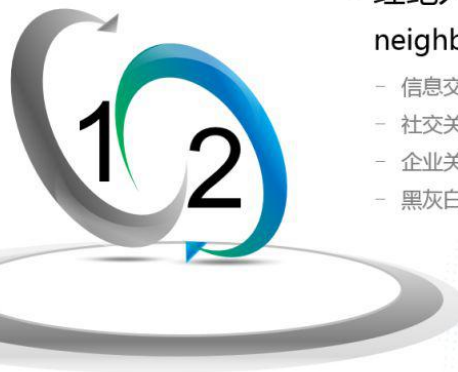
16 2019 KE.COM ALL COPYRIGHTS RESERVED

从前文的贝壳关系图谱的整体架构可以看出，在应用层，关系图谱提供了准入防控，风险量化，品质管理，风险发现，查案溯源等能力。

贝壳

应用case：准入防控

- 门店/加盟商
 - path searching
 - risk path ranking
 - 体外公司搜索
 - 负面信息评估
 - 人员历史风险



- 经纪人
 - neighbor analysis
 - 信息交叉核验
 - 社交关联扫描
 - 企业关联扫描
 - 黑灰白分级

17 2019 KE.COM ALL COPYRIGHTS RESERVED

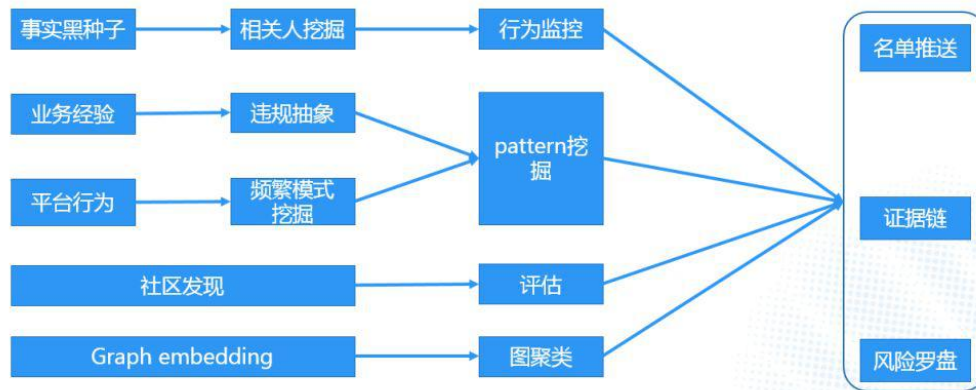
准入防控，主要利用 path searching，risk path ranking 技术，解决风控关系传播中长路径下的风险识别能力，可以进行针对门店的体外公司搜索（包括多种变



种），负面信息评估（路径上是否存在失信人员，被投诉人员等），人员历史风险，以及针对经纪人的信息交叉验证，关联关系扫描以及黑灰白的分级。

贝壳

应用case：风险主动发现

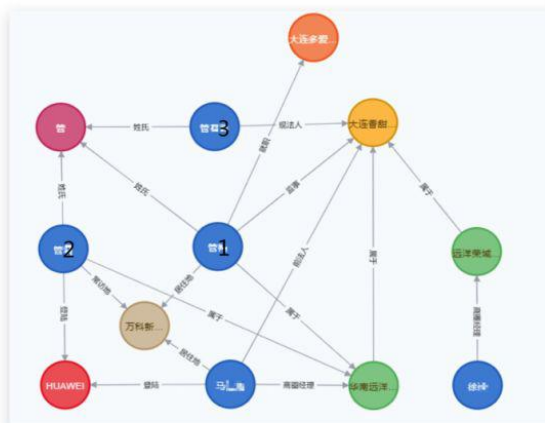


18 2019 KE.COM ALL COPYRIGHTS RESERVED

风险主动发现，主要有三种手段：① 基于举报获得风险种子人群，利用关系图谱监控相关人的行为，发现“白”到“灰”、“黑”的变化；② 基于用户行为以及行业经验的模式提取与搜索；③ 社区发现的 Louvain 算法以及 Graph Embedding 的自动化机器学习挖掘风险方法（下面有进一步介绍）。



应用case：查案溯源



20 2019 KE.COM ALL COPYRIGHTS RESERVED

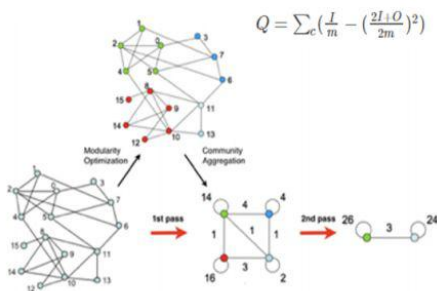
查案溯源，根据种子风险节点的多维属性，探索多条关系路径，找到相关人，进而判断亲密度，发现潜在的风险团伙。

4. 社区发现和自动化机器学习

应用case：风险主动发现

➤ Louvain算法

基于模块度modularity $Q = \sum_c (\frac{\sum_{i \in c} m_i}{2m} - (\frac{\sum_{i \in c} d_i}{2m})^2)$



19 2019 KE.COM ALL COPYRIGHTS RESERVED

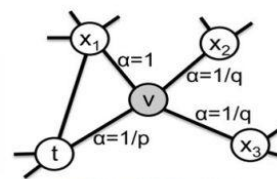
➤ node2vec

优化目标：给定每个顶点条件下，令其近邻顶点出现的概率最大

$$\max_f \sum_{u \in V} \log \Pr(N_S(U) | f(u))$$

Step1 Random Walk => Step2 word2vec

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$



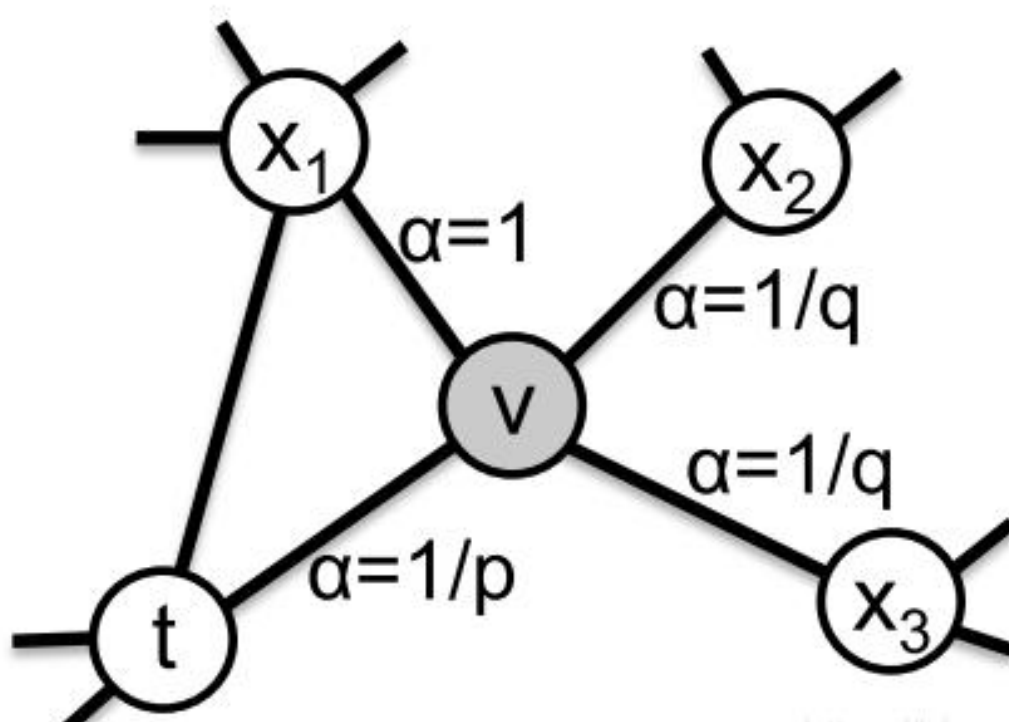
Louvain 是一种经典的基于图的社区发现算法，其优化目标为尽可能提升图的模块度（衡量社区紧密度的标准），模块度提升的定义如下：



$$\Delta Q = \left[\frac{k_{i,in}}{2m} - \frac{\sum_{tot} k_i}{2m^2} \right]$$

其中前面一项为节点加入邻居社区后，社区内的所有边，后一项代表节点加入邻居社区后，社区所有边（包括内部边，以及连接外部的边），目的是希望变化后的子社区内边多，外边少，即社区更聚集。Louvain 是一种迭代式算法，每一轮迭代可以分成两个步骤：① 算法扫描图中的所有节点，针对每个节点遍历该节点的所有邻居节点，衡量把该节点加入其邻居节点所在的社区所带来的模块度的提升，并选择对应最大收益的邻居节点，加入其所在的社区，这一过程化重复进行直到每一个节点的社区归属都不再发生变化；② 对 ① 中形成的社区进行折叠，把每个社区折叠成一个单点，此时新生成的节点之间的边的权重为两个结点内所有原始节点的边权重之和。重复以上两步，多轮迭代直至算法收敛，则可以发现图上的多个社区。Graph Embedding 是图表达的一种方法，其大概思想是将图数据转化为序列数据，进而利用 word2vec 等处理序列数据的方法得到图中结点的向量化表示，而这样的转化通常可以通过图上的随机游走来完成，Node2Vec 就是结合了深度优先和广度优先的图游走方法。





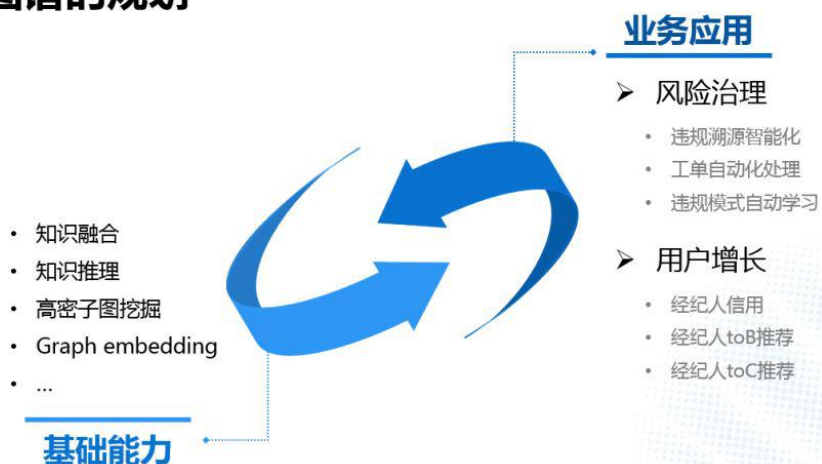
如上图所示，当从结点 t 走到节点 v 时，面临的选择可以分为三种：① 返回节点 t ；② 走到与节点 t 具有一度关系的节点 x_1 ；③ 与节点 t 有二度关系的节点 x_2 或 x_3 。利用 p 和 q 控制着路径返回和继续探索的关系，如果倾向于深度探索，可以将 q 设置小一点， p 设置大一点。



关系图谱的规划



关系图谱的规划



22 2019 KE.COM ALL COPYRIGHTS RESERVED

高密子图的挖掘是社区发现的升级，可以更好地发现团伙，结合关系融合，以及 Graph Embedding 算法，共同提升关系图谱的基础能力。在此基础上的业务应用，可以推广到风险治理和用户增长等场景。本次分享就到这里，谢谢大家。



新一代联邦学习技术及应用实战

WeBank

FATE：新一代联邦学习技术及应用实战

范涛
微众银行高级研究员
dylanfan@webank.com



<https://www.fedai.org/>
<https://github.com/WeBankFinTech/FATE>

分享嘉宾：范涛 微众银行 高级研究员

编辑整理：Hoh Xil

内容来源：微众银行 & DataFun AI Talk

出品社区：DataFun

注：欢迎转载，转载请注明出处

导读：随着大数据的进一步发展，重视数据隐私和安全已经成为了世界性的趋势，同时，大多数行业数据呈现数据孤岛现象，如何在满足用户隐私保护、数据安全和政府法规的前提下，进行跨组织的数据合作是困扰人工智能从业者的一大难题。而“联邦学习”将成为解决这一行业性难题的关键技术。今天会和大家分享下微众银行主导的新一



代联邦学习技术及应用，而 FATE 则是联邦学习落地的一个工业级开源平台。

今天的介绍会围绕下面五点展开：

- 联邦学习背景介绍
- 纵向联邦学习
- 横向联邦学习
- 应用案例
- FATE：联邦学习开源平台

■ 联邦学习背景介绍

首先和大家分享下联邦学习的背景。

1. AI 落地的理想与现实



我们知道 AI 落地的时候，会遇到很多现实的问题：



- 。现实中，我们的数据质量是非常差的，比如聊天数据中有很多噪音；
- 。数据标签，收集是比较困难的，很多场景中的数据是没有标签的；
- 。数据是分散的，（这也是最重要的一点）每家应用的数据不一样，比如腾讯用的是社交属性数据，阿里用的是电商交易数据，微众用的是信用数据，都是分散来应用的。现实中，如何进行跨组织间的数据合作，会有很大的挑战。

2. 国内数据监管法律体系研究

国内数据监管法律体系研究



严格化：数据控制方责任明确，刑罚到自然人

全面化：各领域数据管理细则密集出台，用户授权+监管部门审批

WeBank

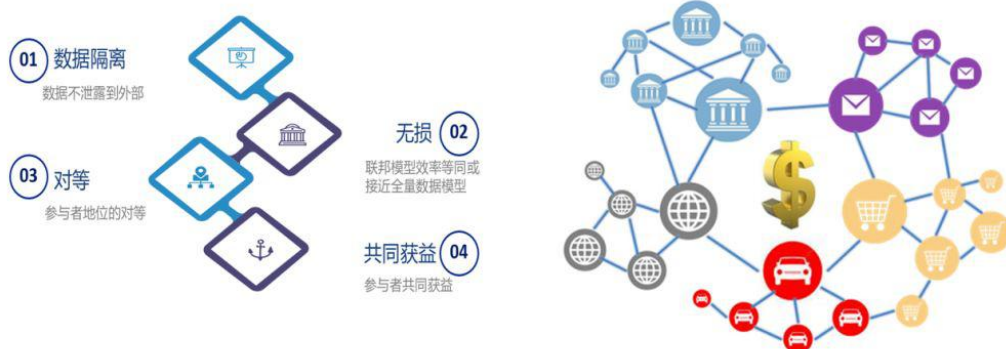
12

从09年到现在的10年时间内，国家关于数据的法律条例是趋向于严格化的，同时趋向于全面化，每个细分领域都纷纷出台了相应的条例和条款。相对来讲，让之前可行的一些数据合作方案变得不太可行。



3. 基于联邦学习的技术生态

基于联邦学习的技术生态



WeBank

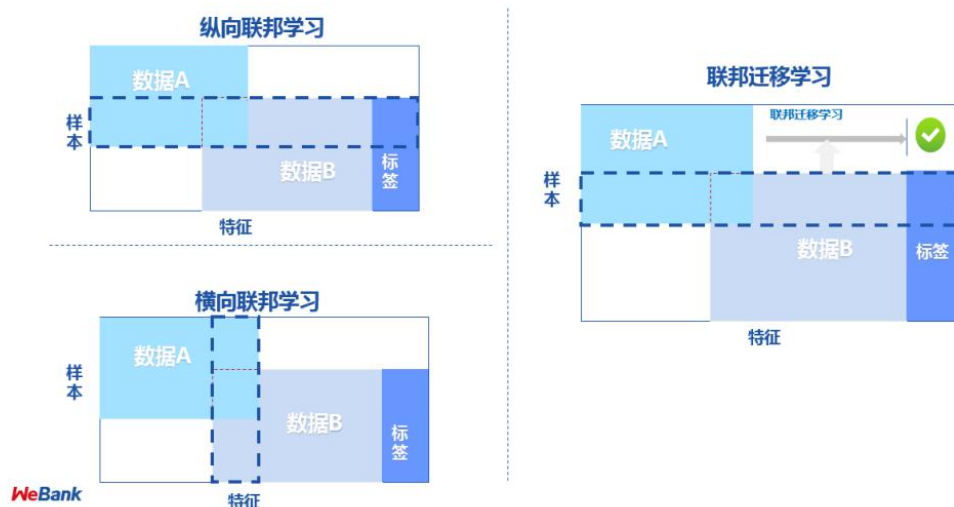
针对上述问题，微众银行提出了基于联邦学习的技术生态，特点如下：

- 数据隔离：联邦学习的整套机制在合作过程中，数据不会传递到外部。
- 无损：通过联邦学习分散建模的效果和把数据合在一起建模的效果对比，几乎是无损的。
- 对等：合作过程中，合作双方是对等的，不存在一方主导另外一方。
- 共同获益：无论数据源方，还是数据应用方，都能获取相应的价值。

4. 联邦学习的分类体系



联邦学习的分类体系



联邦学习的分类体系，包括：

- 纵向联邦学习，两个数据集的用户（ $U_1, U_2, \dots$ ）重叠部分较大，而用户特征（ $X_1, X_2, \dots$ ）重叠部分较小；
- 横向联邦学习，两个数据集的用户特征（ $X_1, X_2, \dots$ ）重叠部分较大，而用户（ $U_1, U_2, \dots$ ）重叠部分较小；
- 联邦迁移学习，通过联邦学习和迁移学习，解决两个数据集的用户（ $U_1, U_2, \dots$ ）与用户特征重叠（ $X_1, X_2, \dots$ ）部分都比较小的问题。

下面，重点分享下纵向联邦学习和横向联邦学习。

纵向联邦学习

1. 联合建模需求场景



纵向联邦学习-联合建模需求场景

举例：微众与合作企业联合建模，微众有Y（业务表现），期望优化本方的Y预测模型

◆ 设定：

- ✓ 只有微众拥有 Y=“逾期表现”
- ✓ 合作企业无法暴露含有隐私的 X

◆ 传统建模方法问题：

- ✓ 合作企业缺乏Y无法独立建模
- ✓ X数据全量传输到微众不可行

◆ 期望结果：

- ✓ 保护隐私条件下，建立联合模型
- ✓ 联合模型效果超过单边数据建模

合作企业				微众银行			
ID 证件号 电话号	X1 帐龄	X2 月薪	X3 等级	ID 证件号 电话号	X4 央行征信分	X5 微众 内部分	Y 表现数据
U1	9	8000	A	U1	600	600	无
U2	4	5000	C	U2	550	500	有
U3	2	3500	C	U3	520	500	有
U4	10	10000	A	U4	600	600	无
U5	5	7500	B	U8	600	600	无
U6	5	7500	A	U9	520	500	有
U7	8	8000	B	U10	600	600	无

业务系统A 数据

业务系统B 数据

WeBank

8

举个例子：微众与合作企业进行联合建模，比如做信贷逾期模型，微众有 Y 数据，包括标签数据，逾期记录，用这样的数据可能会建一个很好的模型，但我们希望用更多的数据，比如合作方的标签数据和画像数据来更大的提升风控模型的效果和稳定性。

传统模式的问题是：

合作企业缺乏 Y 无法独立建立模型，需要微众把 Y 数据，带入到合作方的生产环境建模，但是由于国家的数据保护条款和各企业自身对数据的严格规定，得到的 X 数据不能全量的传输到微众。

针对这个问题，通过纵向联邦学习来解决。如右图，两边的数据都有共同的 ID，特征是完全不一样的，可以通过一方特征来弥补另一方特征的不足。

2. 同态加密技术保护隐私





纵向联邦学习的技术实现，首先应做好两点，来保护数据隐私：

- 建模样本 ID 差集不向对方泄露，在合作之初需要进行用户匹配，需要找出用户的交集，但是不能泄露差集，因为这是企业最核心的资产。
- 任何底层（X, Y）数据不向对方泄露，建模过程中如何保证数据不被泄露。

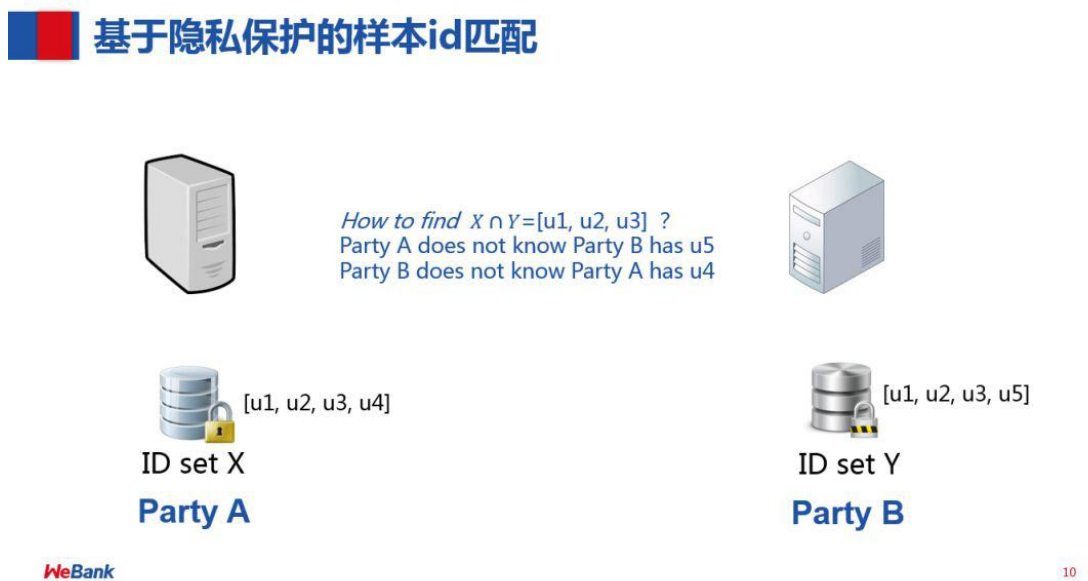
解决方案：

- 通过 RSA 和 Hash 的机制，保证双方最终只用到交集部分，且差集部分不向对方泄露。
- 采用同态加密技术，这个过程中，各方的原始数据，以及数据加密态都没有被传输。交互部分，双方通过损失中间结果，用同态加密的机制进行交互，模型训练完之后，会各自得到一个



模型，各自的模型会部署在各自的一方，就是如果我只提供了 3 个特征，那么我只有 3 个特征的模型，只提供 2 个特征，就只有 2 个特征的模型，任何一方的模型都没法单独去应用，只有共同应用的时候，才能进行决策。

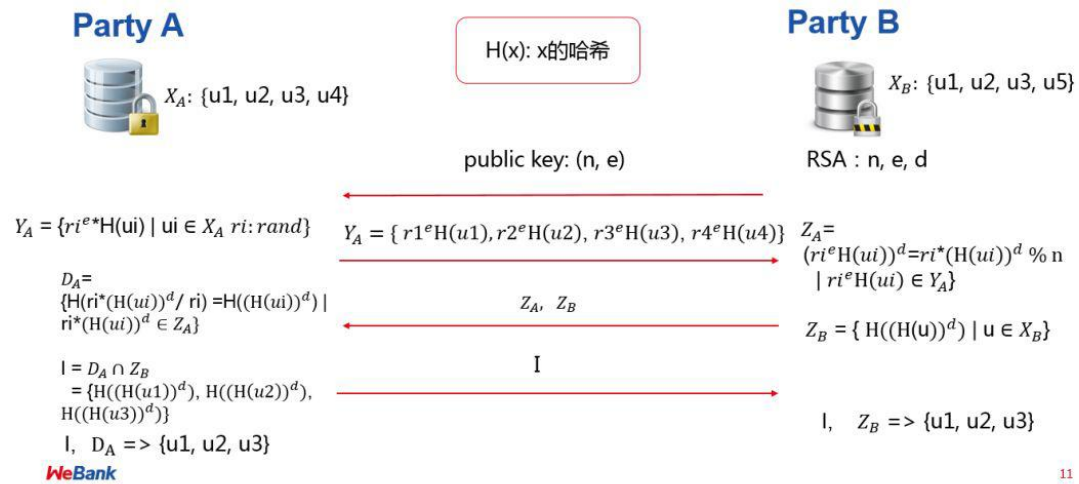
3. 基于隐私保护的样本 id 匹配



刚才提到基于隐私保护的样本 id 匹配，和大家分享下具体的技术方案。比如，A 方有 [u1, u2, u3, u4] 四个用户，B 方有 [u1, u2, u3, u5]，那么整个过程中，如何保证双方知道 [u1, u2, u3]，而 A 方不知道 B 方有 [u5]，B 方不知道 A 有 [u4]？



基于隐私保护的样本id匹配



这里是通过 RSA 和 Hash 的机制做到的，B 方会作为公钥的生成方，会把公钥给到 A 方，A 方基于 Hash 引用一个随机数，再交互传给 B 方，B 方同时做 Hash 然后传给 A 方，A 方会最后做一个结果的交集。整个过程中，你可以看到没有任何一个明文数据传递过来，即使采用暴力或者碰撞的方式，依然解析不出原始的 id。通过这套机制，我们很好的保护了双方的差集部分。

4. 同态加密



同态加密

- 全同态或者半同态 Full Homomorphic Encryption and Partial Homomorphic Encryption
- 数据层面的信息保护 Data-level information protection

Paillier 半同态加密 Partially homomorphic encryption

$$\begin{array}{ll} \text{Addition :} & [[u]] + [[v]] = [[u+v]] \\ \text{Scalar multiplication:} & n[[u]] = [[nu]] \end{array}$$

Rivest, R. L.; Adleman, L.; and Dertouzos, M. L. 1978. On data banks and privacy homomorphisms. Foundations of Secure Computation, Academia Press, 169 - 179.

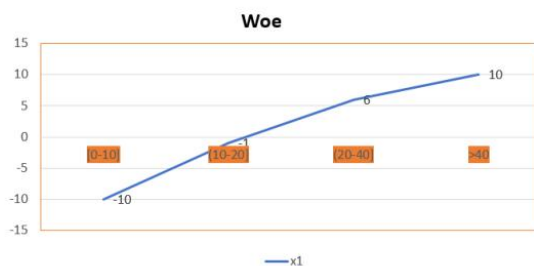
WeBank

12

分享了匹配过程中隐私保护的问题，接下来分享一个通用的技术，同态加密。

刚才提到建模过程中，引用了同态加密技术，比如对两个数字进行加密，加密后两个数字的密文可以进行数学运算，比如加法，其结果依然是密文，对密文解密后得到的结果和它们明文的加法结果是一样的。

联邦特征工程



特征IV值

特征	IV
x1	0.3
x2	0.2
x3	0.4
x4	0.01
x5	0.05

问题：在保护双方隐私下，A侧(含x)和B侧(含x, y)特征如何计算Woe和IV？

难点：

- A侧只有特征x，没有y；计算Woe和IV得同时依赖x,y (B侧特征Woe & IV可以本地计算)
- A侧不能对B侧暴露x，B侧不能对A侧暴露y
- 最终只能让B侧获得所有特征Woe & IV

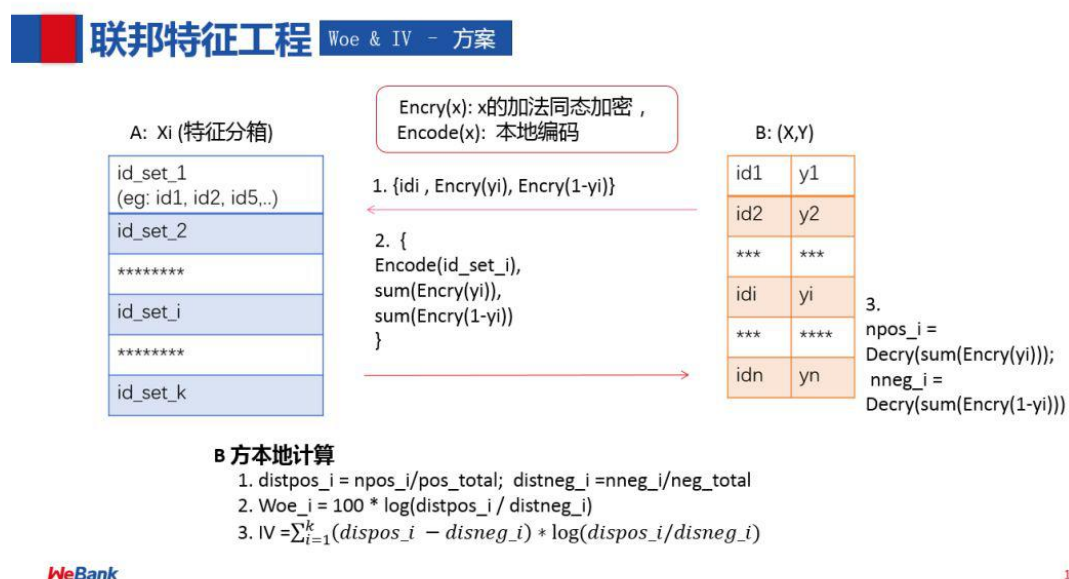
WeBank

13



通过这样的同态加密技术，我们把它应用到机器学习，包括特征工程中。接下来会重点介绍，联邦机制下特征工程和机器学习建模的一些细节。

我们知道特征工程是机器学习建模中非常重要的一环，在联邦机制下，如何完成联邦特征工程？尤其 A 方只有 X 没有 Y，如果想做一个 WOE 或者 IV 值的计算是非常困难的。那么如何在联邦学习的机制下，A 方利用 B 方有 Y 的数据计算 WOE 和 IV 值，且在这个过程中 B 方没有泄漏任何数据？



首先，B 方对 y 以及 1-y 进行同态加密，然后给到 A 方，A 方会对自己的特征进行分箱处理，进而 A 方在分箱中进行密文求和的操作，再把结果给到 B 方进行解密，然后算出 A 方每个特征分箱的 WOE 值和 IV 值。在这个过程中，没有明文数据传输，A 方不知道 B 方的 y 值，同时 B 方



也不知道 A 方每个特征的值是什么，从而在安全隐私保护的情况下，完成了特征工程的计算。

同态加密在机器学习上应用

多项式近似 Polynomial approximation for logarithm function

$$l(w) = \log(1 + \exp(-y w^T x))$$

$$\approx \log 2 - \frac{1}{2} y w^T x + \frac{1}{8} (w^T x)^2$$

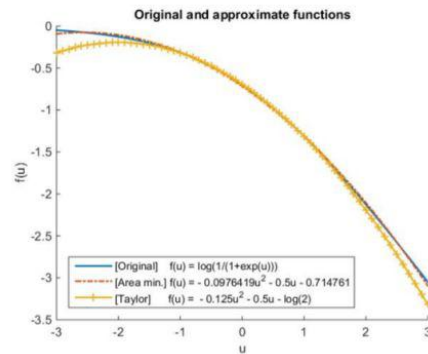
$$\nabla l(w) = \left(\frac{1}{1 + \exp(-y w^T x)} - 1 \right) y x$$

$$\approx \left(\frac{1}{2} y w^T x - 1 \right) \frac{1}{2} y x$$

加密计算 Encrypted computation for each term in the polynomial function

$$loss = \log 2 - \frac{1}{2} y w^T x + \frac{1}{8} (w^T x)^2$$

$$[[loss]] = [[\log 2]] + \left(-\frac{1}{2}\right) * [[y w^T x]] + \frac{1}{8} [[(w^T x)^2]]$$



- Kim, M.; Song, Y.; Wang, S.; Xia, Y.; and Jiang, X. 2018. Secure logistic regression based on homomorphic encryption: Design and evaluation. JMIR Med Inform 6(2)

WeBank • Y. Aono, T. Hayashi, T. P. Le, L. Wang, Scalable and secure logistic regression via homomorphic encryption, CODASPY16 15

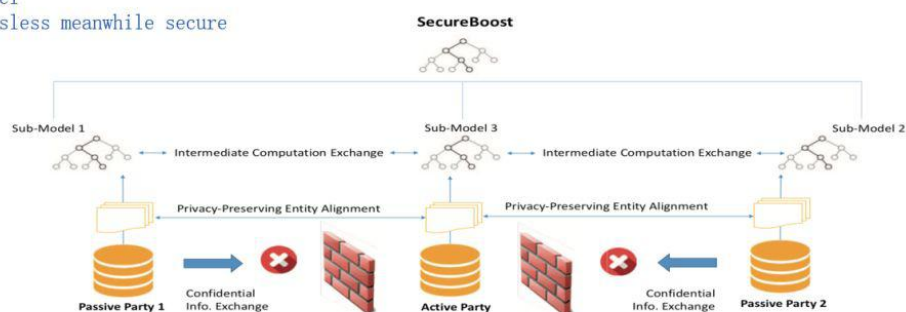
说完特征工程，再讲下最核心的机器学习，比如常见的逻辑回归，这是经典的 loss function 和梯度，刚才说的同态加密的特性，目前用到的是半同态的技术。所以，需要对 loss function 和梯度进行多项式展开，来满足加法操作。这样就可以把同态加密的技术应用在 loss function 和梯度中。

5. SecureBoost



SecureBoost

- Collaboratively learn a shared gradient-tree boosting model
- Lossless meanwhile secure



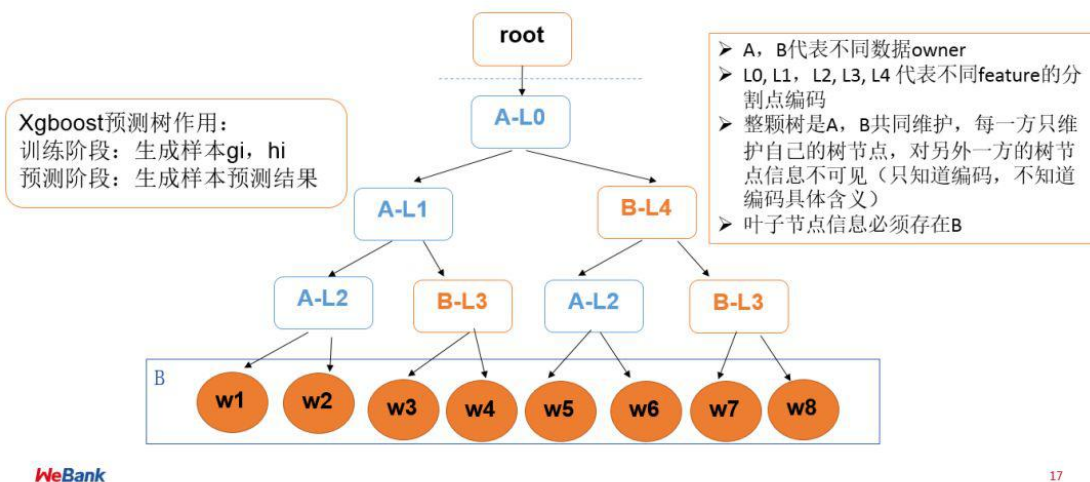
WeBank

Cheng, K., Fan, T., Jin, Y., Liu, Y., Chen, T., & Yang, Q. (2019). SecureBoost: A Lossless Federated Learning Framework. *arXiv preprint arXiv:1901.08753*.

在很多现实的业务应用中，树模型是非常重要的，尤其是 XGBoost，对很多应用来说，提升非常明显，因而被业界广泛使用。在联邦机制下，如何构建这样的树？这里我们提出了 SecureBoost 技术方案，双方协同共建一个 boosting 树，我们证明了整个过程中是无损的。图中为一些 paper 和链接，感兴趣的小伙伴可以查找下。



SecureBoost Xgboost单棵树预测实例



17

SecureBoost 的核心技术点。上图为基于 SecureBoost 构建的树, A 和 B 代表不同数据的 owner, L0, L1, L2, L3, L4 代表不同 feature 的分割点编码, 整颗树由 A 和 B 共同维护, 每一方只维护自己的树节点, 对另外一方的树节点信息不可见（只知道编码, 不知道编码具体含义）, 保证整个训练和预测过程都是安全的。

构建 SecureBoost 核心的关注点是如何构建分裂节点, 尤其一方只有 X , 另一方有 (X, Y) 的情况。





A: Xi (特征bin)

id_set_1
id_set_2

id_set_i

id_set_k

Step1: {idi, Encry(gi), Encry(hi)}

Step2: {
index(id_set_i),
sum(Encry(gi)), sum(Encry(hi))
}Step3:
{max(gain), argmax(gain)}

B

id1	g1	h1
id2	g2	h2
***	***	***
idi	gi	hi
***	***	***
idn	gn	hn

Decry: sum(Encry(gi)), sum(Encry(hi))
计算max(gain), argmax(gain)

隐私保护:

- 信息增益是在B侧本地计算，B侧没有样本信息泄露
- A本地计算加密后的梯度直方图，B解密梯度直方图，但是不知道具体对应的id集合，保护了A侧id集合隐私信息

WeBank

18

基于同态加密的机制，B 方会把 1 阶梯度和 2 阶梯度传递给 A 方，A 方基于分箱之后的结果算出每个分箱中的 1 阶梯度和 2 阶梯度的求和值，然后传递给 B 方，B 方会解密这个求和值，算出信息增益，然后给到 A 方。求分裂节点的核心就是如何算信息增益，通过这样的机制，就可以算出每个分裂节点，同时没有泄露任何隐私信息。

横向联邦学习



横向联邦学习-联合建模需求场景

举例：微众和合作行共建反洗钱模型，期望优化反洗钱模型

◆ 设定：

- ✓ Y 表示“是否存在洗钱行为”
- ✓ 合作行和微众都有 (X,Y)
- ✓ 双方不暴露自己的 (X,Y)

◆ 传统建模方法问题：

- ✓ 微众和合作行各自样本不够多

◆ 期望结果：

- ✓ 保护隐私条件下，建立联合模型
- ✓ 联合模型效果超过单边数据建模

WeBank

微众银行

ID 证件号 电话号	X1 资金来源和 经营范围不符笔数	X2 大额交易 笔数	Y 表现数据
U1	5	15	有
U2	8	20	有
U3	0	5	无
U4	0	0	无
U5	2	1	无
U6	50	50	有
U7	60	6	有

业务系统A 数据

合作行

ID 证件号 电话号	X1 资金来源和 经营范围不符笔数	X2 大额交易 笔数	Y 表现数据
U8	5	10	有
U9	10	2	有
U10	2	30	有
U11	0	10	有
U12	8	7	有

业务系统B 数据

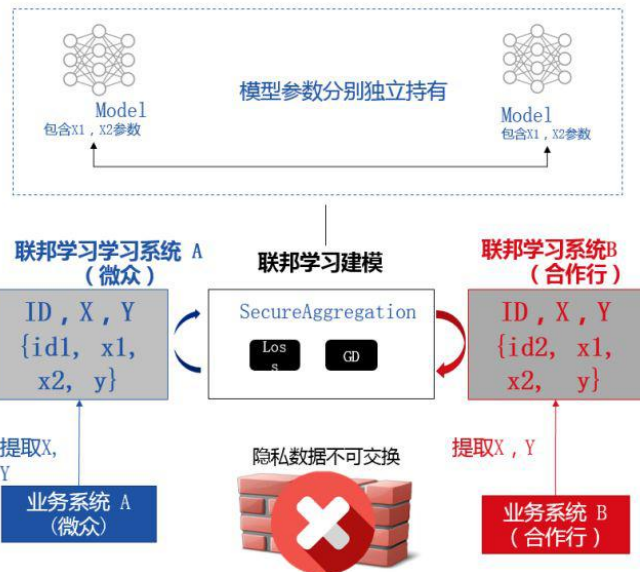
20

举个例子：微众和合作行共建反洗钱模型，期望优化反洗钱模型。因为，各自利用自家样本建立的反洗钱模型的效果和稳定性都不能满足现实需求。我们可以利用联邦学习的机制，充分利用多家的反洗钱样本，同时在不泄露样本的条件下，构建一个非常大的模型，可以看到横向联邦学习中，微众银行和合作行，都是有 (X, Y) 的。



同态加密技术保护隐私

- ◆ 数据隐私保护：
 - ✓ 任何底层X, Y数据不向对方泄露
- ◆ 结果：
 - ✓ 联合模型比单独建模好

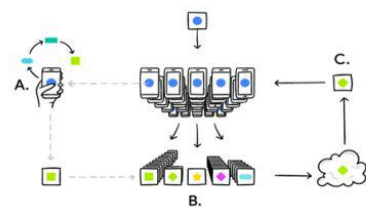


WeBank

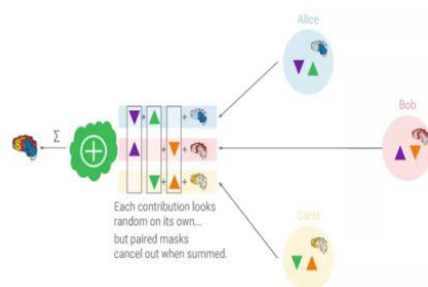
21

技术层面上，采用了同态加密、Secret-Sharing 技术，整个过程中，双方交互的是模型和梯度，同时引入了 SecureAggregation 机制，让交互过程中的梯度也是很难被反解的。最终，大家都会得到一个相同的模型。横向联邦学习，综合多家样本，可以让模型更加稳健，效果更好。

横向联邦核心技术点



H. Brendan McMahan et al, *Communication-Efficient Learning of Deep Networks from Decentralized Data*, Google, 2017



Bonawitz K, Ivanov V, Kreuter B, et al. *Practical secure aggregation for privacy-preserving machine learning*, Google, 2017

WeBank

22



下面分享下横向联邦学习的核心技术点，这是 Google 的两篇 paper，第一个是每个端自己训练模型，然后发给云端进行综合各个模型的效果，但是这里面会有个问题，就是模型本身也可能会泄露隐私信息，所以引入了 SecureAggregation，各方在传输模型的时候会加一些噪音，通过云端来消除这些噪音，使数据和模型得到保护。

应用案例

赋能众多关键领域



24

联邦学习目前已经赋能众多关键领域，取得了不错的效果：

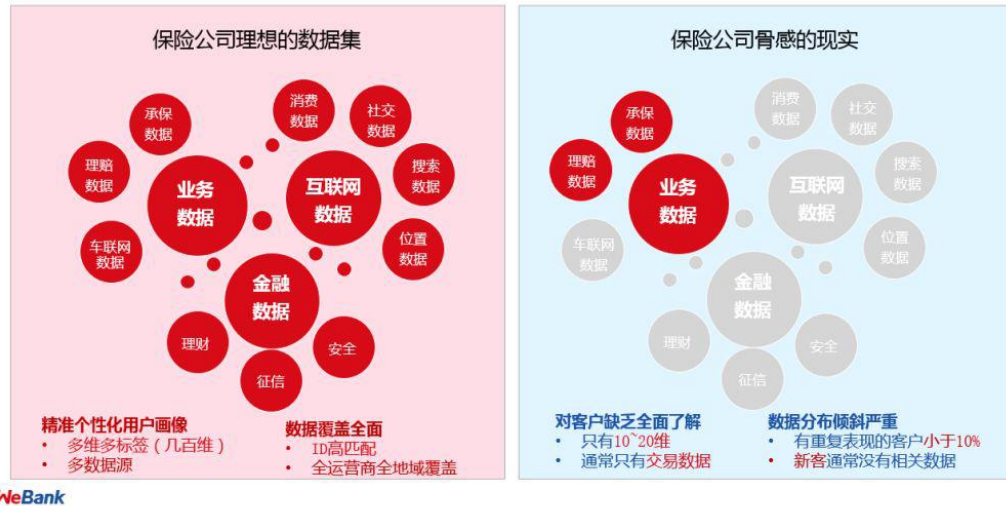
- 银行+监管，联合反洗钱建模
- 互联网+银行，联合信贷风控建模
- 互联网+保险，联合权益定价建模
- 互联网+零售，联合客户价值建模

以其中的两个场景以及在视觉领域的应用来说明：



1. 保险业个性化定价

保险业的个性化定价难题



对于保险公司如果想做个性化的定价，是一件非常困难的事情，保险公司只有一些业务数据、承保数据和理赔数据，通过这样的数据，对用户做千人千面的定价是非常难的。通过联邦学习机制，可以融合多个数据源，来构建这样一个千人千面的定价。



基于联邦学习的保险定价

通过对年龄、职业、年租车次数等标签属性进行联邦学习建模，预测出险概率，决策是否出险

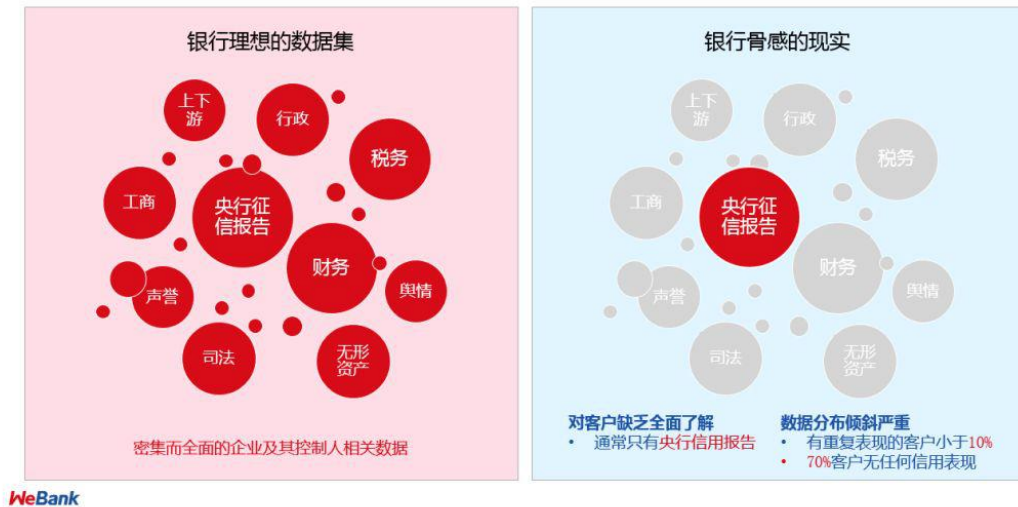


这是我们目前正在做的案例，当你购买权益产品之后，如果违章了，可以帮你免赔，在这里每个人看到的价格是不一样的，这就是我们通过联邦学习帮它们做的。我们综合了出险数据和互联网数据，如标签、用户画像，构建了一个基于保险定价的联邦学习产品。

2. 小微企业信贷风险管理



小微企业信贷的风险管理难题



很多时候对小微企业我们只有央行的征信报告，想要更精准的刻画企业的信用状况，需要更多的数据，比如发票、工商、税务的数据。

基于联邦学习的企业风控模型

通过对近3/6个月开票金额、央行征信分等标签属性进行联合建模，预测是否预期



这是我们的一个案例，我们利用了开票金额和央行的征信数据共建了一个联邦学习模型，来预估每个企业的风险。

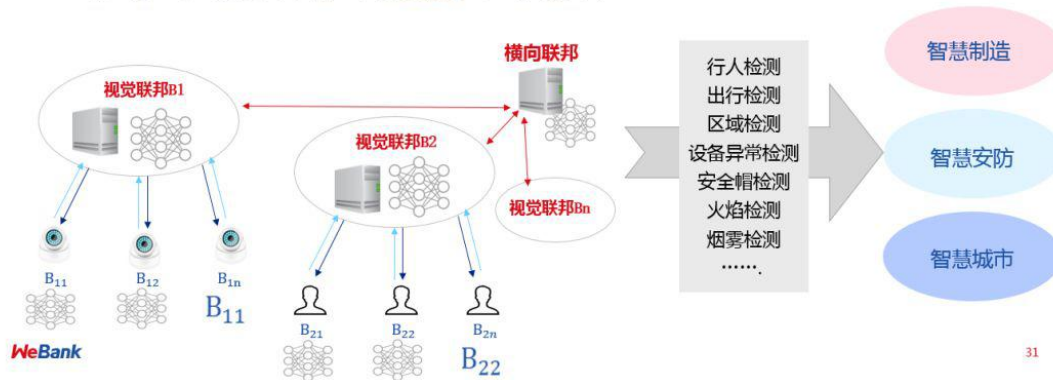


3. 联邦学习在视觉领域的应用

与AI公司探索重塑机器视觉市场

装备制造、物联网AIOT、智慧安防等行业，依托联邦学习，进行视觉市场的场景拓宽

- 优势：
 - 相对于本地建模进一步提升算法准确率；
 - 形成网络效应，降低长尾应用成本，提升视觉业务总体利润率；



与 AI 公司探索重塑机器视觉市场，利用联邦学习的机制，相对于本地建模，进一步提升算法准确率，并且形成网络效应，降低长尾应用成本，提升视觉业务总体利润率。

联邦学习视觉应用 – 城市管理



挑战

- 标签数量少
- 数据分散，集中管理成本高
- 离线延迟的模型更新和反馈

联邦学习

- 在线模型更新和反馈
- 无需集中上传数据
- 数据保护，隐私性高



这是联邦视觉在城市管理上的应用。

■ 联邦学习开源平台：FATE

通过刚刚的介绍，相信大家对联邦学习能够做什么，有了大体的认识。而联邦学习想要落地，不可避免的就是开源，接下来为大家介绍下微众银行开源的联邦学习平台——FATE：



愿景

- 工业级别联邦学习系统
- 有效帮助多个机构在符合数据安全和政府法规前提下，进行数据使用和联合建模

设计原则

- 支持多种主流算法：为机器学习、深度学习、迁移学习提供高性能联邦学习机制
- 支持多种多方安全计算协议：同态加密、秘密共享、哈希散列等
- 友好的跨域交互信息管理方案，解决了联邦学习信息安全审计难的问题

首次发布

2019年1月份，FATE宣布对外开源

Github : <https://github.com/WeBankFinTech/FATE>

WeBank

34

FATE 定位于工业级联邦学习系统，能够有效帮助多个机构在符合数据安全和政府法规前提下，进行数据使用和联合建模。

设计原则：

- 支持多种主流算法：为机器学习、深度学习、迁移学习提供高性能联邦学习机制。
- 支持多种多方安全计算协议：同态加密、秘密共享、哈希散列等。

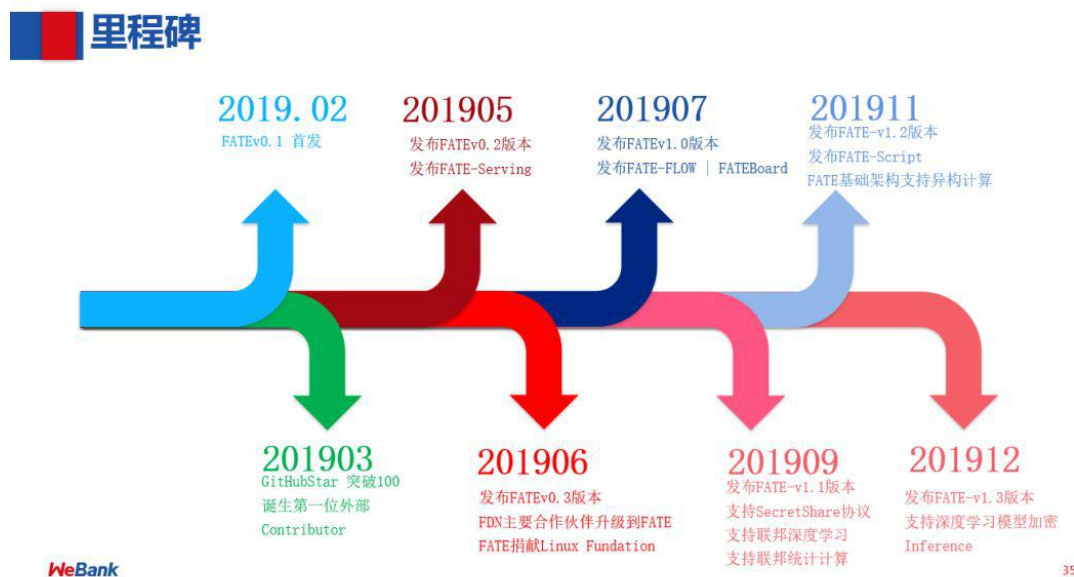


- 。友好的跨域交互信息管理方案，解决了联邦学习信息安全审计难的问题。

FATE 的 Github 地址：

<https://github.com/FederatedAI/FATE>

1. 里程碑



介绍下 FATE 的里程碑，FATE 在今年 2 月份首发了 0.1 版本，3 月份的时候有了第一位外部 Contributor，同时 GitHubStar 突破 100，5 月份发布了 0.2 版本，支持了联邦特征工程和在线推理，6 月份发布了 0.3 版本，把主要的合作伙伴迁移到了 FATE，并把 FATE 捐献给了 Linux Foundation，8 月份发布了 1.0 版本，支持 FATE-Flow 和 FATE-Board，还有些后续的计划，大家可以了解下。

2. 挑战



挑战 Challenges in developing a real-world Federated AI

- 一站式建模过程的联邦化
- MPC协议下分布式算法（on WAN）易理解和易维护
- 跨站点数据传输安全性和可管理性
- 异构基础架构自适应

WeBank

36

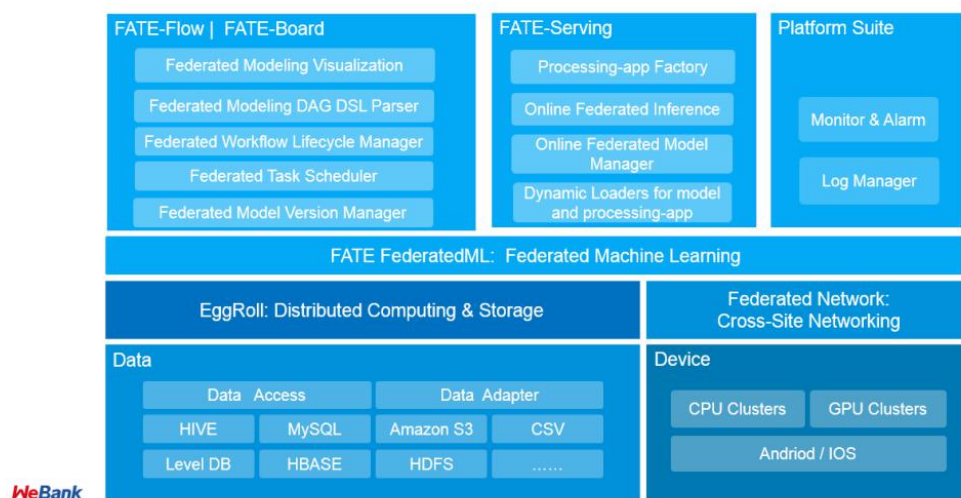
联邦学习从一项技术真正成为一个关键系统和产品方案时，我们遇到了下述挑战：

- 一站式建模过程的联邦化
- MPC 协议下分布式算法（on WAN）易理解和易维护
- 跨站点数据传输安全性和可管理性，如何让交互部分是可以被管理和被审计的。
- 异构基础架构自适应，联邦学习可能会运行在 CPU、GPU 和端上，如何让上层不受底层的变化而变化。

3. 技术架构



技术架构总览



37

接下讲下 FATE 的整体架构：

- EggRoll：分布式计算和存储的抽象；
- Federated Network：跨域跨站点通信的抽象；
- FATE FederatedML：联邦学习算法模块，包含了目前联邦学习所有的算法功能；
- FATE-Flow | FATE-Board：完成一站式联邦建模的管理和调度以及整个过程的可视化；
- FATE-Serving：联邦学习在线推理模块。



一站式联合建模Pipeline

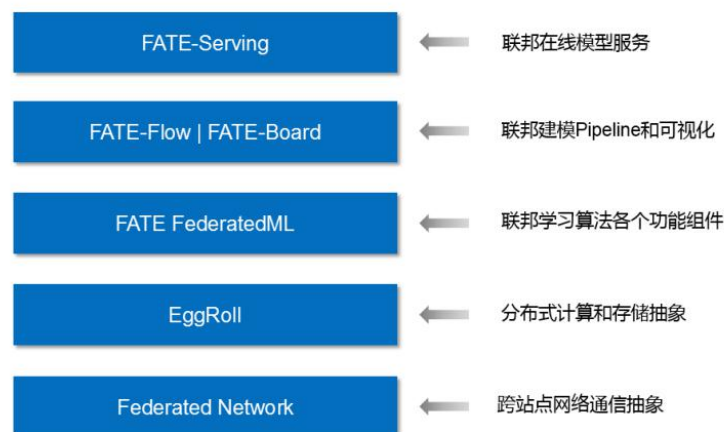


WeBank

38

一站式联合建模 Pipeline，其流程：在开发环境下，其流程是从联邦统计->联邦特征工程->联邦模型训练，当上线部署的时候会有联邦在线推理模块，底层则会采用多方安全计算协议去支持上层各种联邦算法。

核心功能



WeBank

39

这是 FATE 的五大核心功能，这里不再复述。



① FATE FederatedML

 FATE FederatedML

Algorithms	Secure Intersection	Secure Federated Feature Engineering	Secure LR	Secure Boost	Secure DNN/CNN	Secure FTL		
ML Operator	Federated Aggregator	Activation	Regulation	Loss	Optimizer	Gradient	Hessian	
Numeric Operator	Add	Sub	MUL	DIV	Comparison	AND	OR	Scalar Product
MPC Protocol	Homomorphic Encryption	Secret-Sharing	Oblivious Transfer	Garbled Circuit	RSA			
Eggroll & Federation API	Map	MapPartitions	MapValues	Reduce	Join	Remote	Get	

WeBank

42

FATE FederatedML :

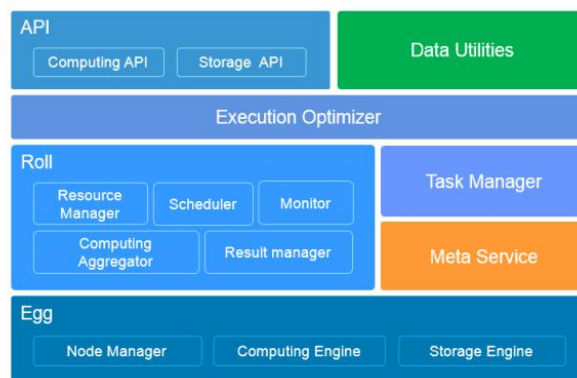
- EggRoll & Federation API : 底层是 EggRoll 的算子 , 比如 Map 和 MapValues , Remote 和 Get 可以完成整个分布式计算的抽象 ;
- MPC Protocol : 包括同态加密、秘密共享等多种多方安全协议 ,
- Numeric Operator : 会抽象出数学算子 , 比如加法或者乘法 ;
- ML Operator : 用建好的数学算子构建机器学习算子 , 而不用管底层的安全协议是什么 ;
- Algorithms : 有了 ML 算子之后就构建各种算法模型。

② EggRoll



EggRoll

- 编程框架
 - EggRoll API：面向算法开发者，通过API实现分布式计算。
- 计算/存储架构
 - 联邦学习一方分布式计算和存储
 - 模块
 - Meta-Service: 元信息管理
 - Roll: 数据 / 计算 调度 (to eggs), 聚合操作等
 - Egg: 计算、存储引擎



WeBank

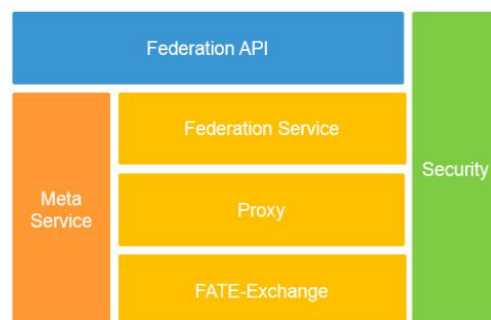
43

EggRoll，是整个分布式计算和存储的抽象。面向算法开发者，通过API 实现分布式计算和存储。上面为 EggRoll 的整体架构图。

③ Federated Network

Federated Network

- 编程框架
 - Federation API：面向算法开发者，通过API实现跨站点通信和交互。
- 通信架构
 - 联邦学习多个参与方跨站点通信
 - 模块
 - Meta-Service: 元信息管理
 - Proxy: 应用层联邦学习路由
 - Federation: Global Object (i.e. data to be 'federated' among parties) 抽象和实现
 - FATE-Exchange



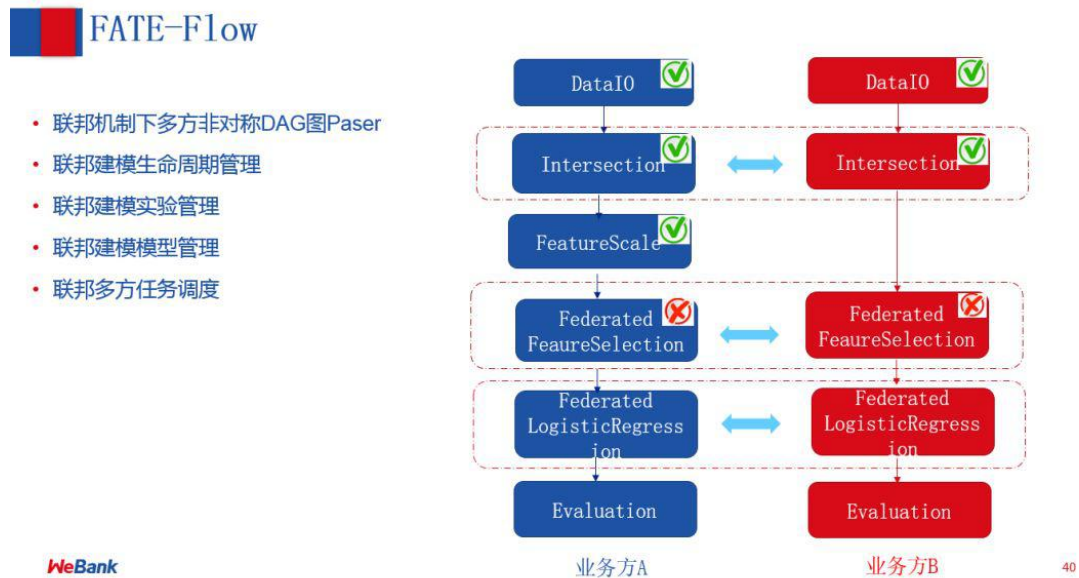
WeBank

44



Federated Network：联邦学习不仅需要分布式计算，还需要跨站点通信和交互，上层会提供一个 API 给到开发者，通过 Remote 和 Get 就可以完成数据点的收发。具体模块，如上图。

④ FATE-Flow



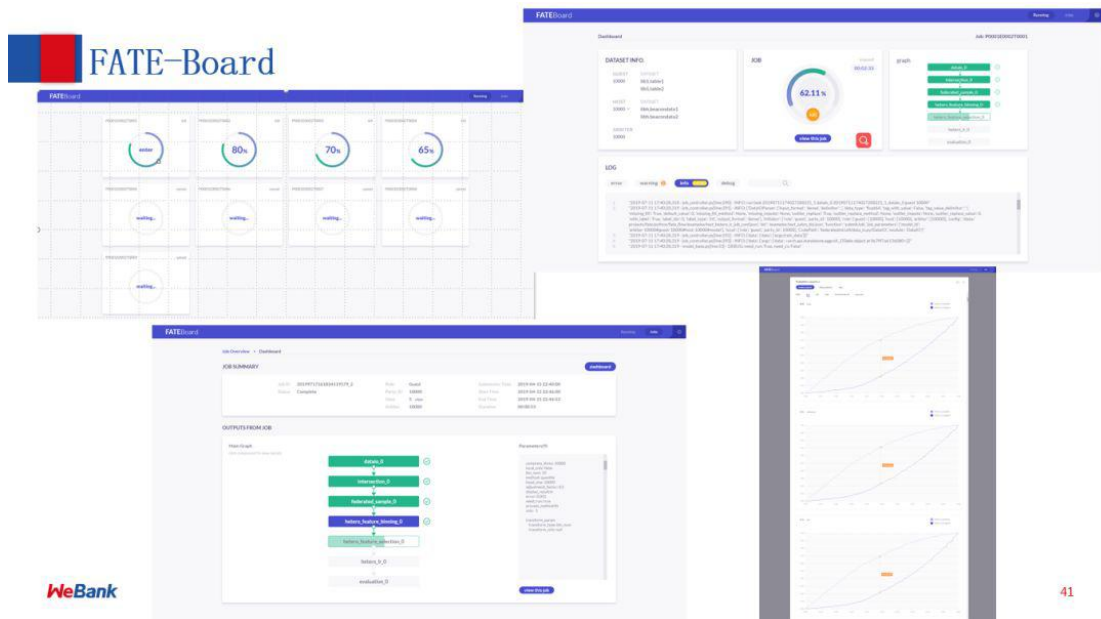
FATE-Flow：整个一站式联合建模 Pipeline 需要统一的调度管理。

右边为 A、B 双方的建模流程，某些步骤是 A、B 双方共有的，某些步骤可能只有一方有，所以 FATE-Flow 完成了下述管理：

- 联邦机制下多方非对称 DAG 图 Paser
- 联邦建模生命周期管理
- 联邦建模实验管理
- 联邦建模模型管理
- 联邦多方任务调度



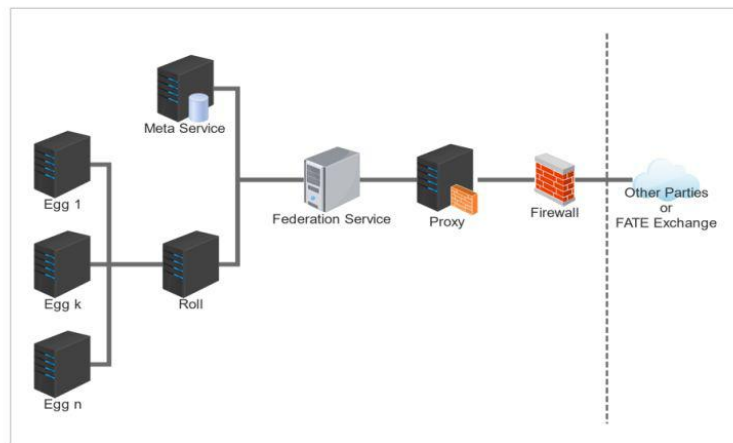
⑤ FATE-Board



FATE-Board 的目的是实现整个联合建模 Pipeline 可视化追踪，记录联邦学习的全过程，可以监控建模进行到哪个步骤，以及模型效果如何。

4. FATE 部署架构

一方部署网络拓扑-示例

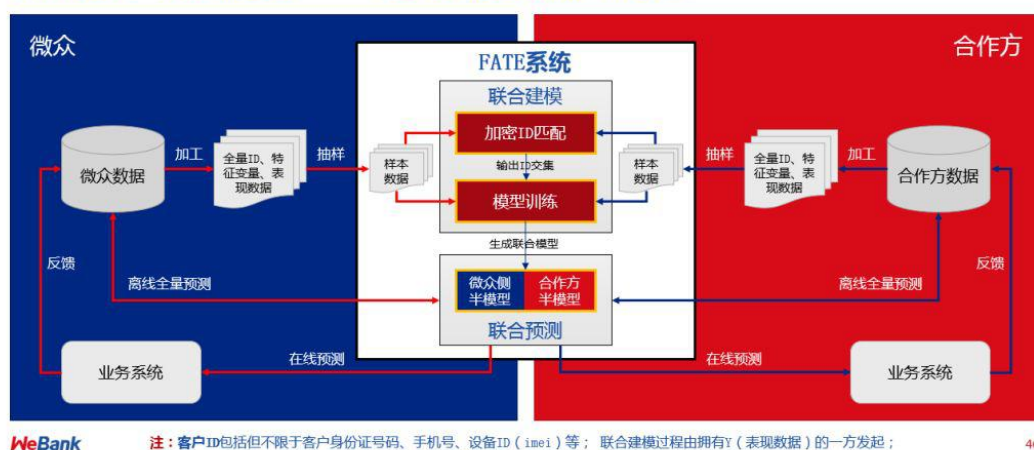


这是 FATE 的部署架构，每一方都是差不多的，是一个对称的结构，通过 EggRoll 实现分布式计算和存储，通过 Federation Service 和外部交互。

5. FATE 应用

基于FATE的联合建模

联合建模、预测示意图 —— 安全合规的数据合作过程



现实中，FATE 是如何应用的呢？这里和大家分享一个示意图，会在每一方部署一套 FATE 系统，双方都是在各自部署的 FATE 系统中进行交互。

6. 开发流程



开发流程



WeBank

47

如果大家对 FATE 感兴趣，作为开发者利用联邦学习框架实现算法，只需要
四步：

- ① 选择一个机器学习算法，设计多方安全计算协议
- ② 定义多方交互的数据变量
- ③ 构建算法执行 workflow
- ④ 基于 EggRoll & Federation Api 实现算法 workflow 中各个功能组件

7. 目前 FATE 项目中的算法和案例



目前 FATE 项目中算法&案例

- Secure Intersection for Sample Alignment
- Vertical-Split Feature Space Federated Feature Engineering
 - Secure Feature Binning
 - Secure Feature Selection
 - Secure Feature Correlation (Coming Soon)
- Vertical-Split Feature Space Federated Learning
 - Secure Logistic Regression
 - Secure Boosting Tree
 - Secure DNN/CNN (Coming Soon)
- Horizontal-Split Sample Space Federated Learning
 - Secure Logistic Regression
 - Secure Boosting Tree (Coming Soon)
 - Secure DNN/CNN (Coming Soon)
- Secure Federated Transfer Learning

WeBank

48

这是目前 FATE 项目中的算法和案例，会根据需求，不断增加各种各样的算法。

最后分享下我们联邦学习的官网：

<https://www.fedai.org.cn/cn/>

今天的分享就到这里，谢谢大家。



构建端到端的联邦学习 Pipeline 生产服务

WeBank

构建端到端的联邦学习Pipeline生产服务

曾纪策
微众银行人工智能系统架构师
jarviszeng@webank.com



<https://www.fedai.org/>
<https://github.com/WeBankFinTech/FATE>

分享嘉宾：曾纪策 微众银行 架构师
编辑整理：Hoh Xil
内容来源：微众银行 & DataFun AI Talk
出品社区：DataFun
注：欢迎转载，转载请在留言区内留言。

导读：本次分享的主题为构建端到端的联邦学习 Pipeline 生产服务。联邦学习的优势在于能够保证参与各方在数据不出本地，保持数据独立性的情况下，多方共建模型，共同提升机器学习效果。联邦机制下，安全隐私有了优势，但技术上也会面临更多挑战。作为一个工业级的框架，端到端的联邦学习 Pipeline 致力于完成高弹性、高性能的联邦学习任务，主要包括建模、训练、模型管理、生产发布和在线推理几个方面。本次将为大家分享如何灵活调度管理复杂的联邦学习任务、可视化联邦建模的实现以



及在线联邦推理服务的思考与实践，解决实验性机器学习到实际生产应用落地的难点。

主要分享 4 个方面：

- 背景介绍
- 高弹性联邦学习 Pipeline 调度
- 联邦学习任务可视化
- 高性能联邦学习在线推理服务

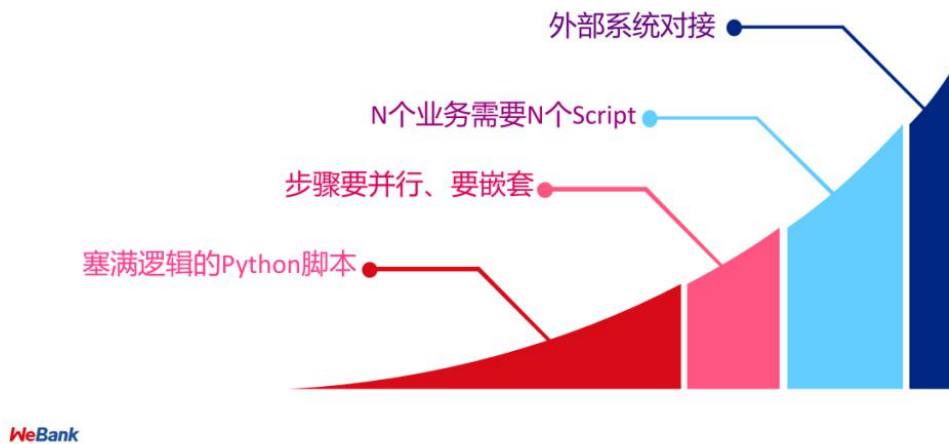


我们在日常建模过程中，会遇到哪些需求和痛点？

1. 机器学习任务编排



机器学习任务编排



我们是如何编排机器学习任务的：

- 塞满逻辑的 Python 脚本。把机器学习的一些步骤，如特征工程、预处理、验证等，写成一个符合逻辑的 Python 脚本。
- 步骤要并行、要嵌套。通过多线程、多进程等手段来完成。
- N 个业务需要 N 个 Script。生产中我们有非常多的业务，每个业务采用的算法也不同，所以会有多个 Script。
- 外部系统对接。一般其他平台在发起自动化机器学习平台任务时，系统对接上是非常困难的，毕竟还没有那么智能，那么如何更好的写好 Python 脚本呢？

2. 机器学习任务状况观察



机器学习任务状况观察



WeBank

当我们千辛万苦的把任务运行起来，还需要不断观察任务的运行情况，运行到哪一步？每一步的数据输出是什么样的？指标输出是什么样的？Loss、AUC 等指标的趋势？然后，任务跑完了吗？

我们该如何观察到各种趋势，来尽可能做参数的调整等操作。这里要大家可以想想，平时是如何观察机器学习任务状态的呢？





有没有需求是看日志不能解决的？

如果有，那就加几行日志！



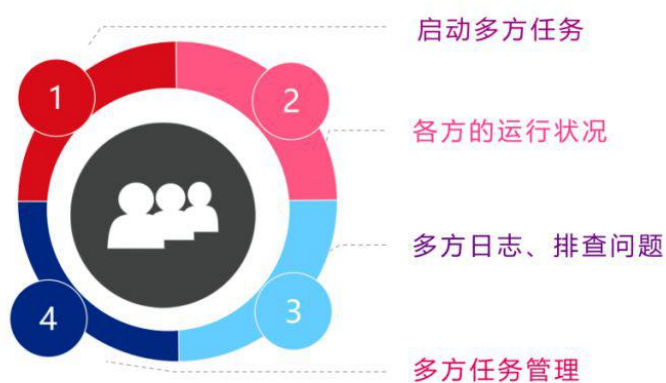
WeBank

如果是我，可能就会看日志。那有没有需求是看日志不能解决的？如果有，那就加几行日志！^_^

3. 联邦学习任务协同建模



联邦学习任务协同建模



WeBank



刚刚介绍的都是一般机器学习任务中会遇到的挑战，在联邦机器学习下，因为涉及多方协同，会遇到更多的挑战：

- 启动多方任务。如需要跟多方协调启动任务，时间难同步。
- 各方的运行状况。需要通过通信工具，同步任务进度。
- 多方日志、排查问题。如日志是散落的，各个操作系统遇到的问题也不一样。
- 多方任务管理。可能非联邦学习的建模只需要 1 个人，而联邦机制下需要 3 个人或者更多。

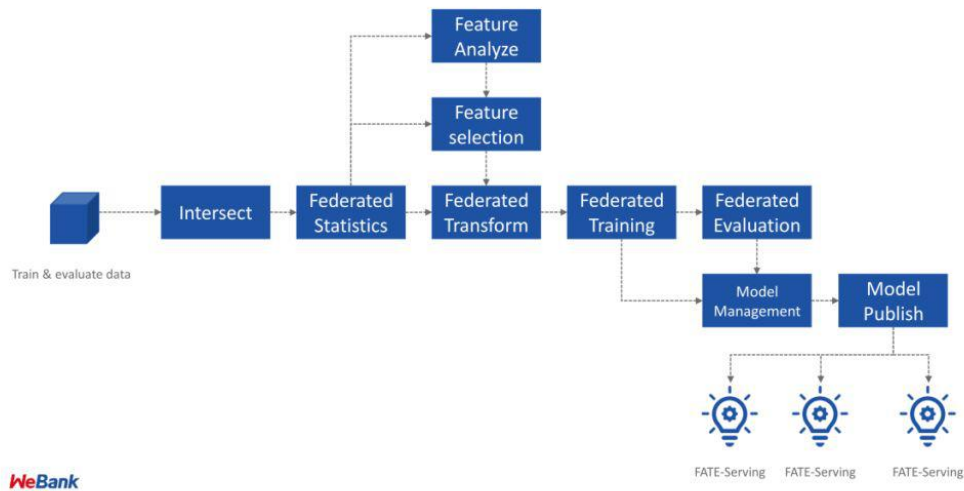


接下来分享下 FATE 是如何尝试解决这样的问题，以及当前的方案和 Pipeline 调度。

1. 端到端的联邦学习 Pipeline



端到端的联邦学习Pipeline



这是一个比较典型的纵向联邦学习 Pipeline 的例子：

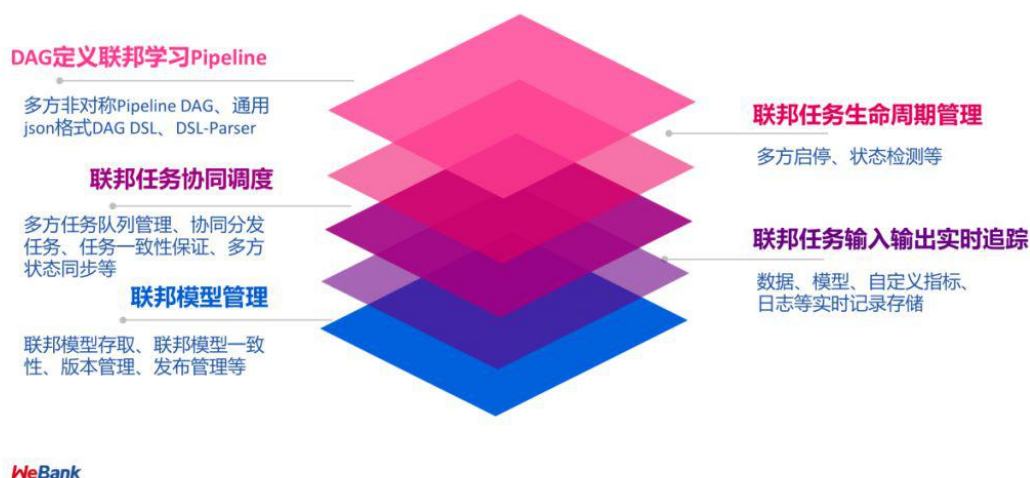
交集 (intersect) -> 联邦统计 -> 联邦特征处理 -> Training -> 验证
-> 模型保存 -> 模型发布 (发布到线上服务 FATE-serving ,后面会介绍)

2. FATE-Flow：端到端的联邦学习 Pipeline 调度平台

① FATE-Flow 功能



FATE-Flow:端到端的联邦学习Pipeline调度平台



FATE-Flow 是我们自研的联邦学习调度平台，主要有 5 项功能：

- 用 DAG 定义联邦学习 Pipeline。DAG 具有比较灵活、弹性的特点，是业界工作流调度系统常用的方式；在联邦学习场景下，DAG 是有难点的，由于协作的机制，我们的 Pipeline 是非对称的；我们在设计 DSL 时，考虑更多系统化对接的情况，所以我们采用了 json 格式的 DAG DSL，然后再采用 DSL-Parser 进行解析。
- 联邦任务生命周期管理。对应刚刚所说的痛点，对联邦任务生命周期进行管理，如多方启停、状态检测。
- 联邦任务协同调度。包括：多方任务队列、分发任务、状态同步等协同调度。
- 联邦任务输入输出实时追踪。这个功能比较普遍，一般的调度平台都会有，包括：数据、模型、自定义指标、日志等的实时记录存储。



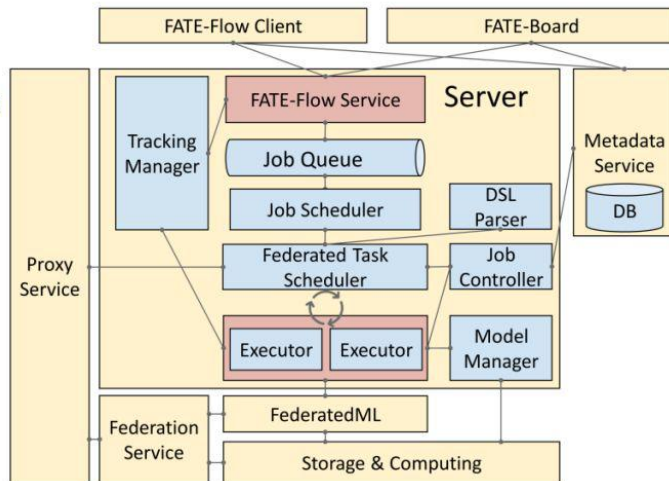
- 联邦模型管理。联邦学习模型的管理存在一个很大的问题，尤其在纵向联邦学习场景下，就是如何保证多方模型的一致性。

② FATE-Flow 架构

FATE-Flow架构

- DSL Parser: 非对称DAG DSL解析器
- Job Scheduler: 联邦任务DAG调度器
- Federated Task Scheduler: 联邦任务DAG节点协同调度中心
- Job Controller: 联邦任务控制器
- Executor: 联邦任务执行节点，Operator容器
- Tracking Manager: 任务输入输出实时追踪器
- Model Manager: 联邦模型管理器

WeBank



FATE-Flow 架构：

- DSL Parser :是调度的核心 ,通过 DSL parser 可以拿到上下游关系、依赖等。
- Job Scheduler :是 DAG 层面的调度 ,把 DAG 作为一个 Job ,把 DAG 里面的节点 run 起来 ,就称为一个 task。
- Federated Task Scheduler : 最小调度粒度就是 task ,需要调度多方运行同一个组件但参数算法不同的 task ,结束后 ,继续调度下一个组件 ,这里就会涉及到协同调度的问题。



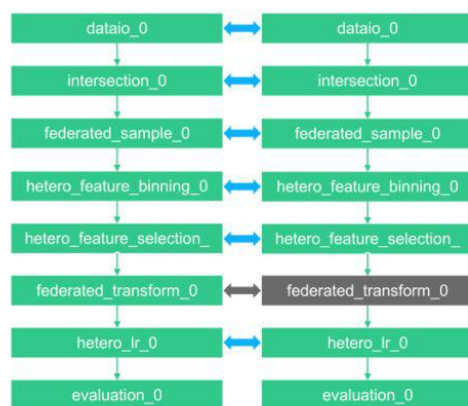
- Job Controller：联邦任务控制器
- Executor：联邦任务执行节点，支持不同的 Operator 容器，现在支持 Python 和 Script 的 Operator。Executor，在我们目前的应用中拉起 FederatedML 定义的一些组件，如 data io 数据输入输出，特征选择等模块，每次调起一个组件去 run，然后，这些组件会调用基础架构的 API，如 Storage 和 Federation Service（API 的抽象），再经过 Proxy 就可以和对端的 FATE-Flow 进行协同调度。
- Tracking Manager：任务输入输出的实时追踪，包括每个 task 输出的 data 和 model。
- Model Manager：联邦模型管理器

3. DAG 定义联邦学习 Pipeline

DAG定义联邦学习Pipeline

```
"components": {  
  "dataio_0": {  
    "module": "DataIO",  
    "input": {  
      "data": {  
        "args.train_data"  
      }  
    },  
    "output": {  
      "data": {  
        "train": "dataio_0",  
        "model": "dataio_0"  
      }  
    },  
    "need_deploy": true  
  },  
  "intersection_0": {  
    "module": "Intersection",  
    "input": {  
      "data": {  
        "dataio_0_train"  
      }  
    },  
    "output": {  
      "data": {  
        "train"  
      }  
    }  
  }  
}
```

WeBank



左边为我们的 DSL ,它的结构比较简单 ,我们可以定义一串 Component ,通过 Parser 解析出 DAG 图(如右图 ,可以清晰地看到整个算法流程的架构)。

Runtime DSL

配置运行DSL，只需三步

- module: 模型组件, FATE当前支持11个模型组件
 - Input:
 - data: 数据输入
 - model: 模型输入
 - isometric_model: 异构模型, 当前只用于Feature Selection
 - Output
 - data: 数据输出
 - model: 模型输出

```
{
  "hetero_lr_0": {
    "module": "HeteroLR",
    "input": {
      "data": {
        "train_data": ["hetero_feature_selection_0.train"]
      }
    },
    "output": {
      "data": ["train"],
      "model": ["hetero_lr"]
    }
  },
}
```

WeBank

构建 DSL 只需要三步：

① Module：模型组件，FATE 当前支持 11 个模型组件，基本满足当前 FATE 所支持的所有算法。

② Input：

- data：数据输入
- model：模型输入
- isometric_model：异构模型，当前只用于 Feature Selection

③ Output：



- data：输出输出
- model：模型输出

可参考下面的例子：

```
},
"hetero_lr_0": {
  "module": "HeteroLR",
  "input": {
    "data": {
      "train_data": ["hetero_feature_selection_0.train"]
    }
  },
  "output": {
    "data": ["train"],
    "model": ["hetero_lr"]
  }
},
},
```

构建 DSL 示例

DAG DSL Parser

1. 根据DSL定义和任务配置，解析每个Component运行参数
2. 分析DSL定义data、model输入输出，提取依赖关系

1. 构建依赖关系邻接表
2. 拓扑排序进行DAG依赖检测

1. 实时输出Component无依赖上下游
2. Component依赖度自动递减

1. 剔除预测阶段无用Component数据，模型依赖传递，推导预测DSL



WeBank



DSL 怎么工作的呢？它是一个非常酷模块，就像人类的大脑，它是 FATE-Flow 的中心：

- 组件初始化：

- ① 根据 DSL 定义和任务配置，解析每个 Component 运行参数
- ② 分析 DSL 定义 data、model 输入输出，提取依赖关系

- DAG 图：

- ① 构建依赖关系邻接表
- ② 拓扑排序进行 DAG 依赖检测，因为用户定义的 DSL 不一定是有效的

- 调度协作：

- ① 实时输出 Component 无依赖上下游
- ② Component 依赖度自动递减

- 预测 DSL 推导：

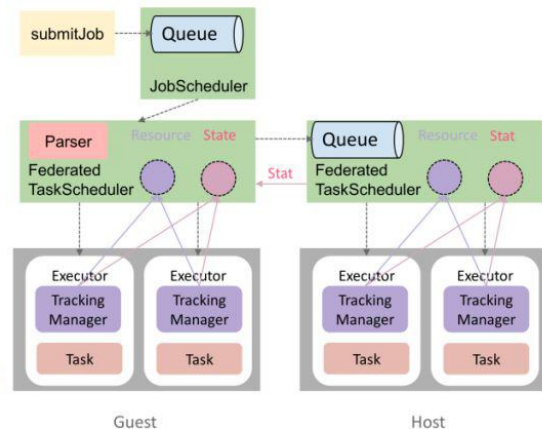
剔除预测阶段无用 Component 数据，模型依赖传递，推导预测 DSL

4. 联邦学习任务多方协同调度



联邦学习任务多方协同调度

- Federated Scheduler
 - Run DAG as Job, Run Component as Task
 - Initiator, 为调度控制方
 - 支持all_succss,all_done,one_success等策略
 - 支持rerun, specified_task_run等特定运行
- 一个Pipeline DAG Component Task为最小调度单位
 - Initiator JobScheduler从队列取出一个Pipeline DAG Job, 分发到TaskScheduler.
 - Federated TaskScheduler 从Parser取得N个无依赖的 Component, 启动多个Executor 执行, 并同步任务指令到联邦其他参与方
 - 联邦参与方取得任务, 如果New Job, 则放入队列; 否则启动多个Executor执行
 - 联邦参与方定期调度队列中的Job, 发起执行
 - Initiator TaskScheduler会等待收集该Task在所有方的运行状态



WeBank

联邦学习任务多方协同调度的流程：

首先，是以任务提交的一种方式，提交任务到 Queue，然后 JobScheduler 会把这个任务拿出来给到 Federated TaskScheduler 调度，Federated TaskScheduler 通过 Parser 取得下游 N 个无依赖的 Component，再调度 Executor（由两部分组成：Tracking Manager 和 Task）执行，同时这个任务会分发到联邦学习的各个参与方 Host。

联邦参与方取得任务，如果是 New Job，则放入队列（参与方会定期调度队列中的 Job），否则启动多个 Executor 执行，Executor 在 run 的过程中，会利用 Federation API 进行联邦学习中的参数交互，对一个联邦学习任务，每一方的 Job id 是保持一致的，每跑一个 Component，它的 Task id 也是一致的。每个 Task 跑完 Initiator TaskScheduler 会收集各方的状态，进行下一步的调度。对于下一步的调度策略我们支持：

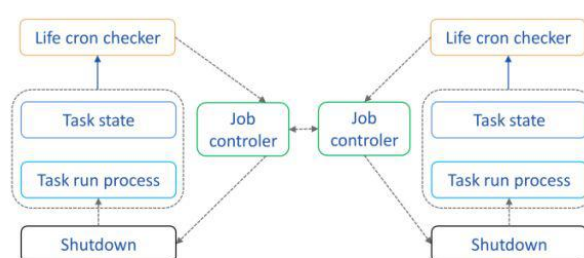


all_succss , all_done , one_succss 等策略。由于我们基于 Task 为最小的调度单位，所以很容易实现 rerun , specified_task_run 等特定运行。

5. 联邦任务多方生命周期管理

联邦任务多方生命周期管理

- Task stat: Task状态信息，如启动时间、运行状态、结束时间、超时时间等
- Task run process: Task运行进程
- Life cron checker: Task生命周期定时检测
- Job Controller: 联邦任务控制器
- 若Task运行时间超过配置超时时间或默认超时时间（一般较长），启动Shutdown
- 若Task运行进程异常终止，启动Shutdown
- 若Task正常运行终止，启动Shutdown
- Shutdown流程: kill process、清理任务以及同步指令到所有联邦参与方，保证联邦任务状态一致性



WeBank

分以下几个部分：

- Task stat : Task 状态信息，如启动时间、运行状态、结束时间、超时时间等
- Task run process : Task 运行进程
- Life cron checker : Task 生命周期定时检测
- Job controller : 联邦任务控制器



- Shutdown :kill process、清理任务以及同步指令到所有联邦参与方，
保证联邦任务状态一致性

启动 Shutdown 的条件：

- 若 Task 运行时间超过配置超时时间或默认超时时间(一般较长)，启动 Shutdown
- 若 Task 运行进程异常终止，启动 Shutdown
- 若 Task 正常运行终止，启动 Shutdown

6. 联邦任务输入输出实时追踪

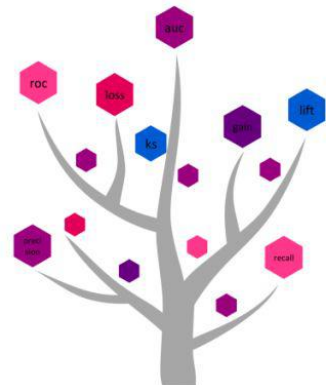
联邦任务输入输出实时追踪

- Definition
 - metric type: 指标类型，如auc, loss, ks等等
 - metric namespace: 自定义指标命名空间，如train, predict
 - metric name: 自定义指标名称，如auc0, hetero_lr_auc0
 - metric data: key-value形式的指标数据
 - metric meta: key-value形式的指标元信息，支持灵活画图
- API
 - log_metric_data(metric_namespace, metric_name, metrics)
 - set_metric_meta(metric_namespace, metric_name, metric_meta)
 - get_metric_data(metric_namespace, metric_name)
 - get_metric_meta(metric_namespace, metric_name)

WeBank



VS



联邦任务输入输出实时追踪，首先会有几个 Definition 定义：

- metric type：指标类型，如 auc，loss，ks 等等
- metric namespace：自定义指标命名空间，如 train，predict



- metric name：自定义指标名称，如 auc0，hetero_lr_auc0
- metric data：key-value 形式的指标数据
- metric meta: key-value 形式的指标元信息，支持灵活画图

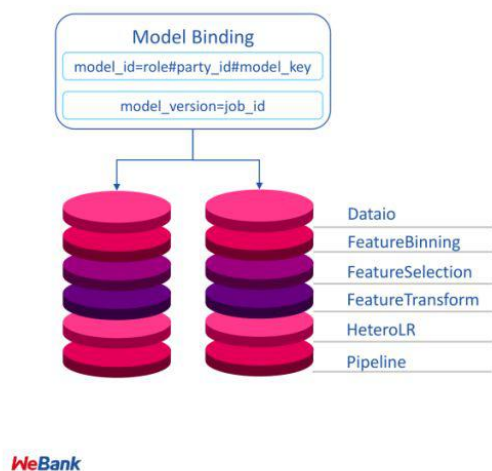
目前的 API 只有 4~5 个：

- log_metric_data(metric_namespace, metric_name, metrics)
- set_metric_meta(metric_namespace, metric_name, metric_meta)
- get_metric_data(metric_namespace, metric_name)
- get_metric_meta(metric_namespace, metric_name)

可能以前收集指标需要经过收集日志等一系列操作，任务像一座座大山一样摆在面前，现在则有可能成为我们的摇钱树，因为我们可以快速的收集各种指标，提交给需求方。

7. 联邦模型管理



联邦模型管理

- 使用Google Protocol Buffer作为模型存储协议，利用跨语言共享
- 每个算法模型由两部分组成：ModelParam & ModelMeta
- 一个Pipeline产生一系列算法模型
- 命名为Pipeline的模型存储Pipeline建模型DSL及在线推理DSL
- 联邦学习下，需要保证所有参与方模型一致性，即模型绑定
- model_key为用户提交任务时定义模型标识
- 联邦各方的模型ID由本方标识信息role、party_id，加model_key
- 联邦各方的模型版本必须唯一且保持一致，FATE-Flow直接设置为job_id

```
system = "gridpy"

package_name = websockets.get_package_name("featuredata")
system_jax_inter_classname = "FeatureDataInterFeatureData"

Message FeatureDataInterFeatureData = {
  message_type, double[] feature_values = 1;
}

Message LeftCols = {
  repeated string original_cols = 1;
  message_type, bool[] left_cols = 2;
}

Message FeatureDataInterFilterParam = {
  message_type, double[] feature_values = 1;
  message_type, FeatureDataInterHostFeatureValues = 2;
  LeftCols left_cols = 3;
  message_type, LeftCols host_left_cols = 4;
  string filter_name = 5;
}

Message FeatureDataInterFilterParamRepeated = {
  repeated FeatureDataInterFilterParam results = 1;
  LeftCols final_left_cols = 2;
}
```

[illegible]

左图中的两桶“大饼”，分别代表了某一方的模型，每一个“大饼”则代表了每个组件的 model，如：Dataio、FeatureBinning、FeatureSelection、FeatureTransform、HeteroLR、Pipeline。这里需要做个 Model Binding 模型的绑定，FATE-Flow 的做法还是比较简单的，我们会给每套模型赋予一对标志符 model_id 和 model_version 来唯一标识模型，model_id 由用户自定义的 role 和 party_id 及 model_key 拼接而成，model_version 也是可以自定义的，如果不自定义的话，会默认为 job_id。我们会有一个命名为 Pipeline 的模型存储 Pipeline 建模 DSL 及在线推理 DSL。

下面是某个算法模型数据结构的示例：



```
syntax = "proto3";

package com.webank.ai.fate.common.mlmodel.buffer;
option java_outer_classname = "FeatureSelectionParamProto";

message FeatureValue {
    map<string, double> feature_values = 1;
}

message LeftCols {
    repeated string original_cols = 1;
    map<string, bool> left_cols = 2;
}

message FeatureSelectionFilterParam{
    map<string, double> feature_values = 1;
    map<string, FeatureValue> host_feature_values = 2;
    LeftCols left_cols = 3;
    map<string, LeftCols> host_left_cols = 4;
    string filter_name = 5;
}

message FeatureSelectionParam{
    repeated FeatureSelectionFilterParam results = 1;
    LeftCols final_left_cols = 2;
}
```



```
syntax = "proto3";

package com.webank.ai.fate.common.mlmodel.buffer;
option java_outer_classname = "FeatureSelectionMetaProto";

message FeatureSelectionMeta {
    repeated string filter_methods = 1;
    bool local_only = 2;
    repeated string cols = 3;
    UniqueValueMeta unique_meta = 4;
    IVValueSelectionMeta iv_value_meta = 5;
    IVPercentileSelectionMeta iv_percentile_meta = 6;
    VarianceOfCoeSelectionMeta variance_coe_meta = 7;
    OutlierColsSelectionMeta outlier_meta = 8;
    bool need_run = 9;
}

message UniqueValueMeta {
    double eps = 1;
}

message IVValueSelectionMeta{
    double value_threshold = 1;
}

message IVPercentileSelectionMeta{
    double percentile_threshold = 1;
}

message VarianceOfCoeSelectionMeta {
    double value_threshold = 1;
}

message OutlierColsSelectionMeta {
    double percentile = 1;
    double upper_threshold = 2;
}
```

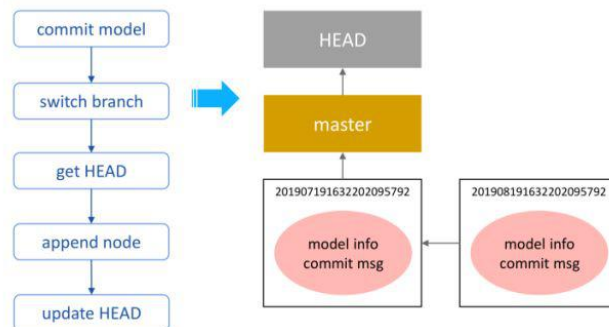
例 2

同时每个“大饼”算法模型，也由两部分组成：ModelParam 和 ModelMeta，也就是参数和元的信息。

8. 联邦模型版本管理

联邦模型版本管理

- 基于多叉树的版本记录
- 支持commit message
- 支持分支功能，如experiment, product, release
- 支持tag，如release
- 支持history查看
- 支持版本回溯



WeBank

模型版本管理我们参考了 Git 的实现思路，但是我们没有做的那么复杂，是基于多叉树版本的记录：

- 支持 commit message；
- 支持分支功能，如 experiment，product，release；
- 支持 tag，如 release；
- 支持 history 查看；
- 支持版本回溯，指定某一版本回滚。

9. 使用样例





使用样例

```
python fate_flow_client.py -f submitJob -d examples/test_hetero_lr_job_dsl.json -c
examples/test_hetero_lr_job_conf.json
python fate_flow_client.py -f jobStatus -j 20190720105134389931_1
python fate_flow_client.py -f kill -j 20190720105134389931_1
```

```
{
  "data": {
    "job_dsl_path": "/data/projects/fate/python/jobs/20190720105134389931_1/job_dsl.json",
    "job_runtime_conf_path": "/data/projects/fate/python/jobs/20190720105134389931_1/job_runtime_conf.json",
    "job_url": {
      "arbiter": [
        "http://10.10.10.10:8080/#/details?job_id=20190720105134389931_1&role=arbiter&party_id=10000"
      ],
      "guest": [
        "http://10.10.10.10:8080/#/details?job_id=20190720105134389931_1&role=guest&party_id=10000"
      ],
      "host": [
        "http://10.10.10.10:8080/#/details?job_id=20190720105134389931_1&role=host&party_id=10000"
      ]
    },
    "model_id": {
      "arbiter": [
        "arbiter#10000#arbiter-10000#guest-10000#host-10000#model"
      ],
      "guest": [
        "guest#10000#arbiter-10000#guest-10000#host-10000#model"
      ],
      "host": [
        "host#10000#arbiter-10000#guest-10000#host-10000#model"
      ]
    },
    "model_version": "20190720105134389931_1"
  },
  "jobId": "20190720105134389931_1",
  "meta": null,
  "retcode": 0,
  "retmsg": "Success",
  "created_at": "2019-07-20 10:51:34"
}
```

WeBank

上图为，FATE-Flow 的简单使用样例，主要就是使用 FATE-Flow CLI 提交一个 Job，需要提供 Job 的 DSL 描述以及配置文件，那么 FATE-Flow Server 会返回该 Job 的一些必要信息，尤其唯一 Job Id 比较重要。后面则是查询 Job 状态以及停止 Job 的操作指令，CLI 还支持许多丰富的指令，可以参考 github 上的文档。



03

联邦学习建模可视化

WeBank

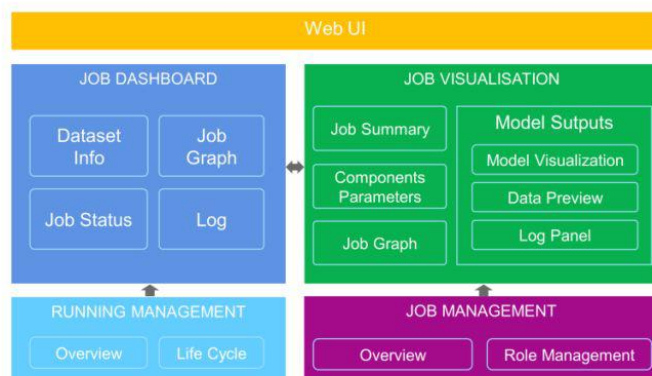
21

第三部分介绍下联邦学习建模可视化：

1. FATE-Board

FATE-Board

FATE-Board作为FATE联邦建模的可视化工具，旨在跟踪和记录联邦建模全过程的信息，并通过可视化的方式呈现模型训练过程的变化以及模型训练结果，帮助用户简单而高效地深入探索模型与理解模型。



WeBank



大体的架构如右图，有一些 Job DashBoard 和可视化，两个基本的管理和上面的 Web UI。

FATE-Board 作为 FATE 联邦建模的可视化工具，旨在跟踪和记录联邦建模全过程的信息，并通过可视化的方式呈现模型训练过程的变化以及模型训练结果，帮助用户简单而高效地深入探索模型与理解模型。

2. 建模交互及可视化

Demo

其基本步骤为：

- ① 用户配置 pipeline，建立 graph、定义 parameters 等；
- ② 用户提交 job，返回 job URL，同时启动 job 运行，进入 web 端查看 fateboard；
- ③ 观察 job 运行状态，查看运行时的统计信息，包括运行进度、日志、模型过程输出等；
- ④ 查看 job 运行完成的结果，包括模型输出、模型评分、日志等内容及可视化结果。



04

高性能联邦学习在线推理服务

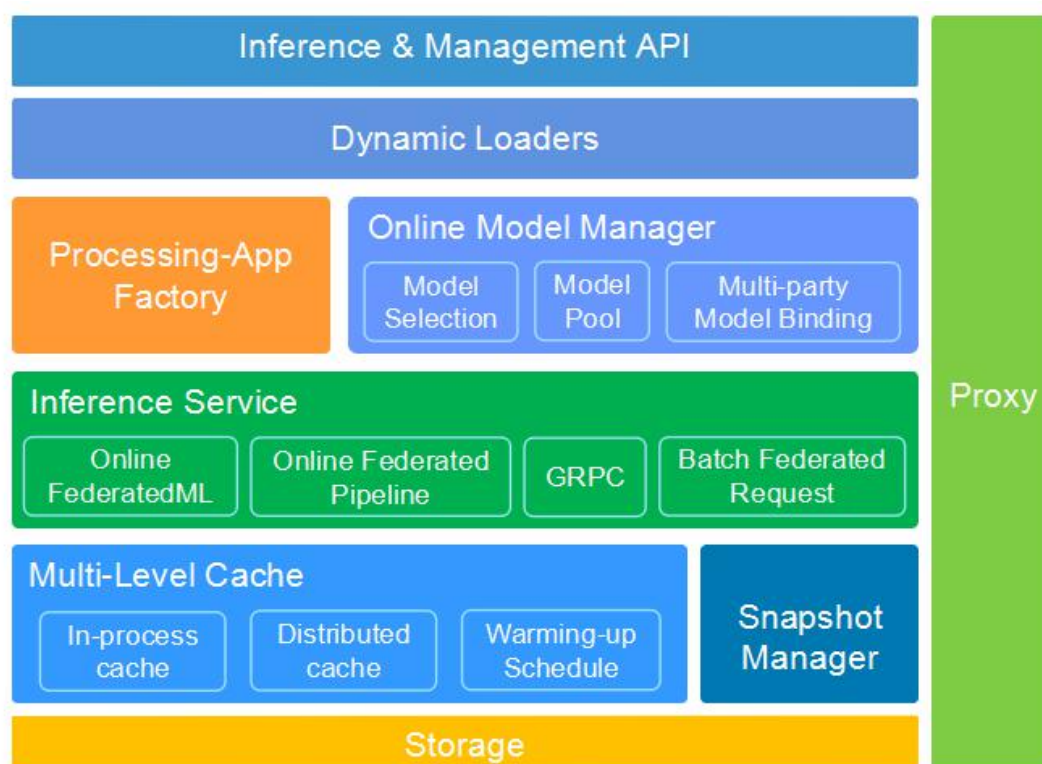
WeBank

24

第四部主要讲怎么发挥模型的最大价值，我们构建了高性能的联邦学习在线推理服务：

1. FATE-Serving





FATE-Serving 设计原则：

- 高性能，基于 GRPC 协议，批量联邦请求，联邦参与方模型结果多级缓存
- 高可用，无状态设计，异常降级功能
- 高弹性，模型 & 数据处理 App 动态加载

FATE-Serving 模块：

- Online FederatedML：高性能在线联邦学习算法包，在线和离线时性能是不一样的，在线的时候，我们没有采用 Python，当前版本是采用 Java 写的一套联邦学习算法。

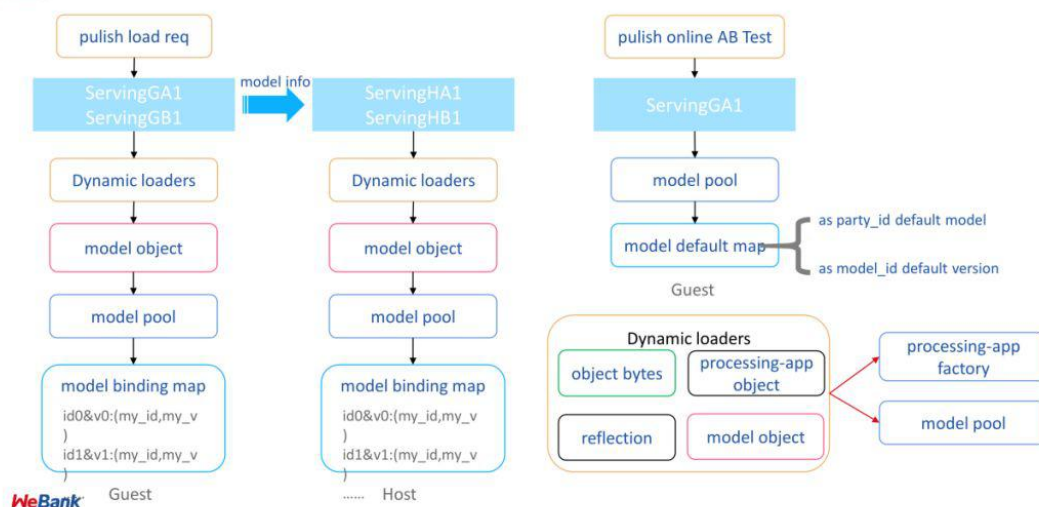


- Online Federated Pipeline：在线联邦推理 Pipeline，比较简洁高效，后面会详细介绍。
- Dynamic Loaders：推理节点动态加载器。目前支持模型和 APP 两种推理节点，节点通过训练或者编译等方式创建后是序列化后存储在分布式存储上的。对于在线服务，每个请求都需要经过每个节点，为了保证执行效率，需要把这些节点提前从存储中取出缓存在 Server 的内存空间，这也是业界在线服务的常用做法。
- Model Manager：在线模型管理器
- Processing-App Factory：数据处理节点工厂。在解决实际业务问题时，还需在模型预测的基础上在 Pipeline 中引入一些人工或者规则，打包成一个 App 发布到 Serving，在预测的时候加入一个前处理 Processing 和一个后处理 Processing。
- Multi-Level Cache：多级缓存管理器
- Snapshot Manager：快照管理，定期将在线模型、Processing-App 信息落库。

2. 在线联邦模型管理



在线联邦模型管理



在线联邦模型管理，我们提供了两个 API：publish load req 和 publish online AB Test。为什么是两个 API？因为，Publish load req 过程只是把整个模型通过 dynamic loaders 得到一个 model object 放入 model pool 中，是一个内存的池子。联邦各方的 model id 和 model version 都是保持一致的（在发布模型的时候，只需要在 Guest 侧发起就可以了，会自动走 proxy，把命令推到其他的 Host 中去），当各方都加载完之后，就可以应用了。

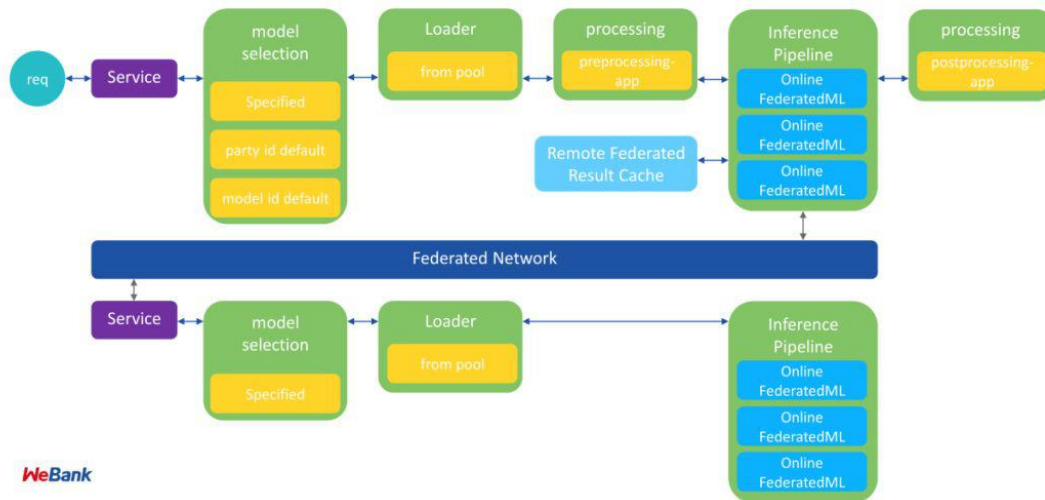
Publish online AB Test 主要是做 default binding，比如有些业务方并不知道我们的模型版本，需要有一个默认的机制。

图中右下角，为动态加载器的一个基本功能。

3. 在线联邦推理 Pipeline



在线联邦推理Pipeline



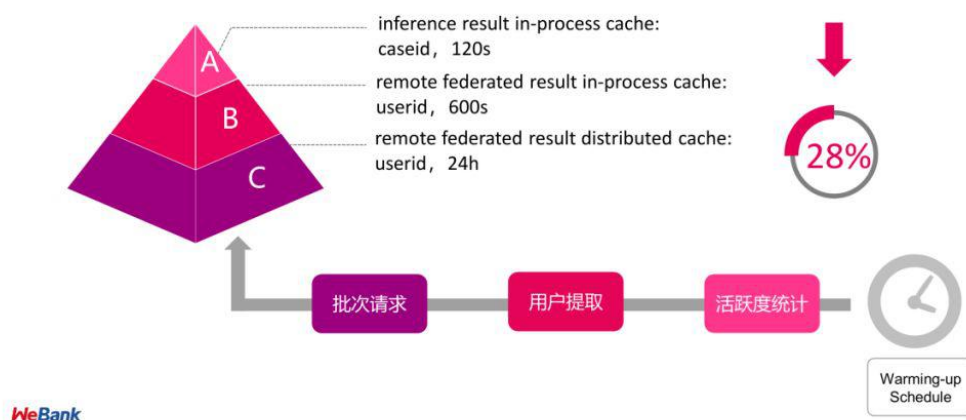
在线联邦推理 Pipeline 流程：

Req -> Service -> model selection (有三种不同策略：规定的方式和两种默认的方式) -> loader -> processing (前处理) -> inference Pipeline (是纵向 model，会经过不同的组件，这里举例是三个，然后把 user_id (用来做对齐的) 发到不同的参与方，参与方会拿自己的半模型，进行最后的预测，最后把结果发到 Federated Network) -> Processing (后处理)

4. 在线推理服务缓存



在线推理服务缓存



在线推理服务缓存分为三种：

- Inference result in-process cache : cache key 为 caseid , 缓存 120s。
- Remote federated result in-process cache :cache key 为 userid , 缓存 600s。
- Remote federated result distributed cache : cache key 为 userid , 缓存 24h , 跟 A 方的特征更新周期有很大的关系。

在业务的运行当中，按照这几个配比，一天下来发现命中远端联邦推理结果 Cache 的请求量占了 28% ,也就是说有 28%的联邦请求都不需要这部分，就会大大的增加了用户的体验。



我们还有一个缓存唤醒的热身过程，我们会跑一些离线任务，然后对活跃度进行统计，把要缓存的用户按比例提取出来，在凌晨的时候发一些批次请求，让整个缓存热身。

5. 联邦模型应用生产服务流程

联邦模型应用生产服务流程



WeBank

一个联邦模型应用到生产服务的大致流程：

- ① 全量加载联邦模型，通过 `publish load req` 全量加载模型，每方的 Serving 都加载这个模型。
- ② 灰度上线联邦模型，`publish online AB test` 可以指定多少用户上线 online 模型。（这时就会体现出为什么要做 `publish load req`，因为在 Guest 方灰度上线时，这个 Serving 请求的 `user_id` 落到其他参与方，需要找到对应的模型，但是通常情况下 Serving 都是做负载均衡的，如



果灰度这部分的 Serving 已经上线了新的模型，其它 Host 方没有预先加载这个模型，就会找不到这个模型，也就是说模型是被所有 Serving 所 load 的，但是并不是所有 Serving 当前生效的都是这个模型。）

③ 联邦模型效果验证

④ 全量上线联邦模型

最后分享下我们联邦学习的官网：

<https://www.fedai.org.cn/cn/>

以及 FATE 的 Github 地址：

<https://github.com/FederatedAI/FATE>



风控建模流程：以京东群体感知项目为例



分享嘉宾：曾军威 博士 京东

编辑整理：白云飞

内容来源：百融云创&DataFunTalk

出品社区：DataFun

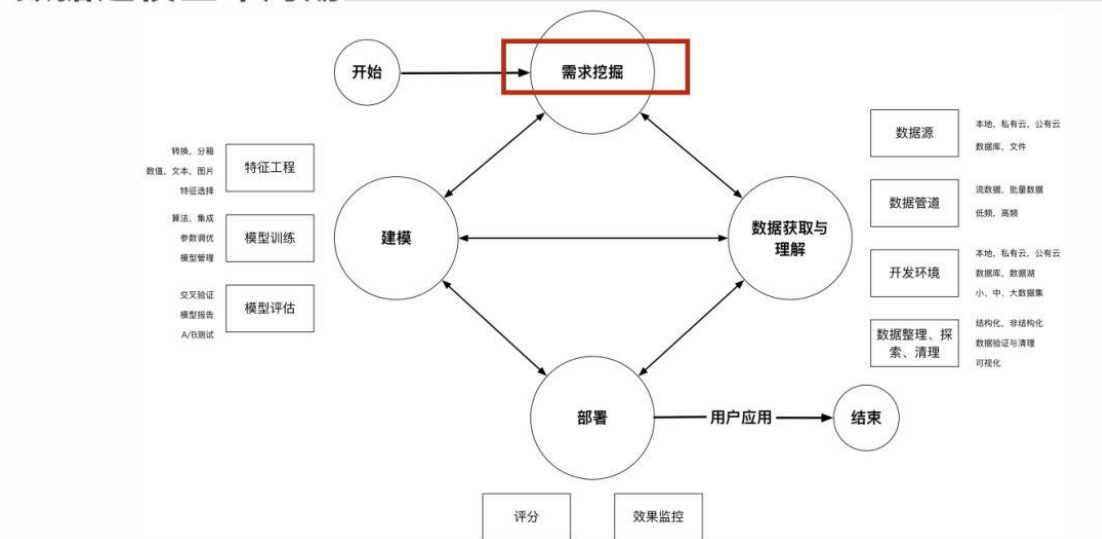
注：欢迎转载，转载请注明出处

导读：本次分享将以群体风险感知为例，从**需求挖掘、数据挖掘、建模再到最终的模型部署应用**，详细介绍全流程的风控建模方案。下面将从这几个方面出发，详细的讲解具体流程中的概念，同时会给出每个流程中的**目标、实现方法、交付物**，让同学们在具体实践的时候有目标，有方法。

■ 业务需求挖掘



数据建模生命周期



任何项目的起源，都是从需求挖掘开始的，不可能一蹴而就，会经过多次循环。这体现了作为数据分析师和建模人员的功力，可能有的业务要 A，你就给他 A，而没有往下深入几个层次进行考虑，**因为有的时候，可能业务的需求自己也没有想清楚需要的是什么，这和传统的软件工程是一样的，所以宁可在这里多花一点时间，也好过到后面踩坑之后，在回来重新挖掘需求。**

① 目标

- 找出关键变量（与判断项目成功与否的指标相关，最好的方法就是定义几个场景，比如做风控，是坏账率下降，还是机注量下降了，或者其他指标），并把关键变量作为模型目标
- 识别业务已经使用或者需要获取的数据源

② 实现方法



- 定义目标，与用户以及其他利益相关方一起理解，识别业务问题，并构建多个问题(这些问题要能定义出业务目标同时也能作为建模的目标)
- 识别数据源：基于问题找出能帮助回答的数据源，越多越好

③ 交付物

- 项目需求文档（比如 A、B、C 三个场景，机注量下降 2%，或者坏账率下降 1%等等）
- 数据源梳理，有几张宽表，有几个数据源，都需要弄清楚

④ 业务方的诉求与抱怨

群体风险感知

业务方的诉求与抱怨

多 快 准 省

业务方的诉求与抱怨，抽象来说就是多、快、准、省：

- **多**：在风控反欺诈领域，对于抓黑产，召回率越高越好。
- **快**：实时反应，不能等到 T+1 时间过去了才反应，比如拼多多的风控事件，不仅仅需要做到实时，最好做到事前，并且可以发出告警，提示哪个环节被攻击。



- **准**：准确率（auc，roc 曲线）大家都比较熟悉，尤其对于金融风险，准确率要求更高。
- **省**：省心，傻瓜式操作，不需要告诉具体指标的含义。对于模型，只需提供给业务方 0 或 1 即可。

所以，对于业务的诉求和抱怨，就在这四个字之中，当我们接到群体风险感知这个 Case 时，我们首先需要做的就是：定义群体风险感知（什么叫群体而不叫团伙？），明确业务场景，找出业务痛点。

两个场景：

- 一堆人来买我的货，我不会认为他们是坏的人；
- iphone 新品首发，黄牛带货，对我的品牌或者收益造成损失，这时，我们需要进行控制打压。

所以我们交付的东西，需要是中性的，**把模型看作一把刀，具体刀用来做什么，由业务来决定。**

⑤ 需求分析



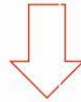
群体风险感知（需求分析）

一、未知风险可能出现的状态：

- 少量交易造成巨大损失
- 大量交易造成巨大损失

二、黑灰产的特征

- 逐利性（不断升级、不断寻找漏洞）
- 群体性（短期、快速、大量获利）



以黑灰产群体作为特征指标，发现交易链条上可能存在的漏洞
增强防控能力，提高黑产作弊成本

风险未知可能出现的状态：

- 少量交易造成巨大损失（例如：网络舆情事件，诈骗，网络攻击等）
- 大量交易造成巨大损失（例如：拼多多事件，上万个账号薅优惠券，然后快速的变现）

黑灰产的主要特征：

- 逐利性（不断升级，不断寻找漏洞）：由于他们是在为自己牟利，没有 kpi 没有 996，甚至一天可以更新迭代 4 个版本，他们自己干的很起劲。
- 群体性（短期，快速，大量获利）目前成熟的电商，基本上不会让一个账号短时间内下很多单了，现在的黑产为了达到短期快速大量获利，一般使用群控手机，即用软件控制几百台手机，尤其在 6·18 和 11·11 时，给我们造成的压力也是比较大的。

项目的定义：



以黑灰产群体作为特征指标，发现交易链条上可能存在的漏洞增强防控能力，提高黑产成本（黑产本质上是生意，风控人员不是为了消灭黑产，是为了让他们这门生意做不下去，明面上是技术的对抗，实际上是**成本的对抗**）。

⑥ 项目目标

监控指标体系与异常发现（输出结果形式）：

- **线索账号的线索**：某个特定群体的行为（观察某个小白鼠在迷宫里的行为特征）
- **群体维度线索**：多个群体聚集情况（观察一群小白鼠在迷宫里的群体共同行为）

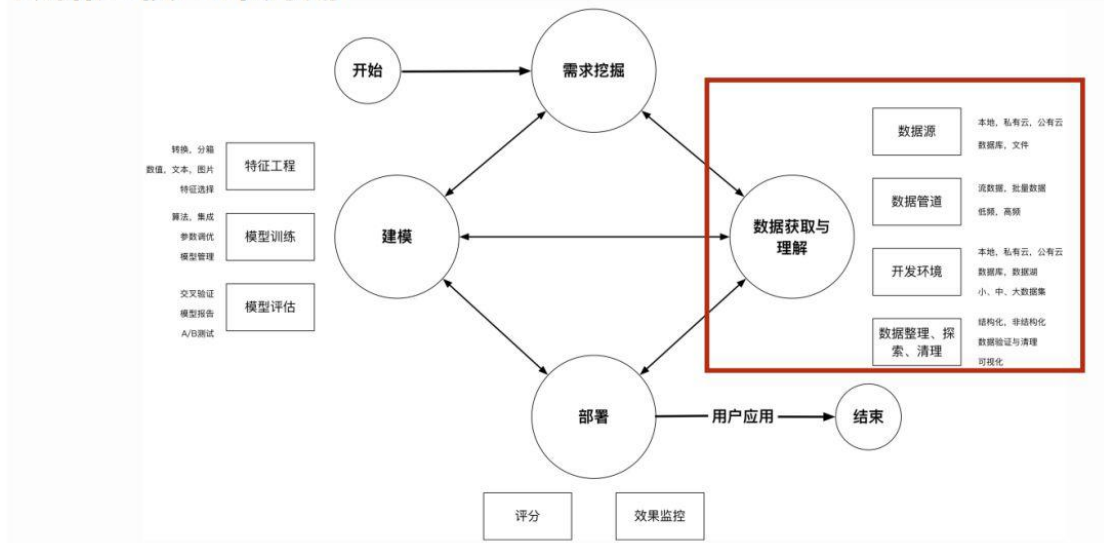
监控指标体系与异常发现（输出结果内容）：

- **聚集性指标**，灵活制定规则（告诉业务方，商品、库存、优惠券是不是被攻击了）
- **提前发现风险点**，预警后续环节（黑产领券之后，会不会有什么大动作，需要提前预警）
- **SKU，优惠券，订单，相关账号**（我们有一个评分，一旦被群体模型识别，就打一个黑标签，让风险值变高）
- **新的套利**，作弊方式，交易漏洞（将黑产规律输出给分析师，分析黑产的作弊方式以及交易环节的漏洞）

■ 数据获取与理解



数据建模生命周期



数据获取与理解，主要分四个部分：数据源（在哪分布，可能需要做数据同步，也有可能需要买一些服务，定期做 API 的推送，需要把数据源定义清楚），数据管道（如何做清洗、加工，如何接入到模型中），开发环境（根据数据规模，选择开发环境），数据整理、探索、清理：结构化、非结构化，数据验证与清理，可视化等等。

① 目标

- 产出一个干净、高质量的数据集，充分的理解数据集和项目目标变量的关系
- 把数据集放置在一个适合开发环境以便进行后续建模
- 开发数据管道的解决方案，定期对数据进行刷新

② 实现方法

- 将数据导入目标分析环境



- 研究数据以确定数据质量是否足以回答项目要解决的问题（PS：前段时间比较火的生物探针技术，可以获取一些黑灰产的特征，但这项技术落地的时候，回收的数据缺失率很高，很难得到应用）
- 设置数据管道来抽取新的数据或者定期刷新数据

③ 交付物

- 数据质量报告
- 数据管道架构
- 可行性评估

④ Tity Data (with Pandas)

具体地址：<http://pandas.pydata.org> 包含了语法的介绍，重塑，聚合，缺失数据处理，窗口函数，可视化等等。



Data Wrangling with pandas Cheat Sheet

<http://pandas.pydata.org>

Syntax – Creating DataFrames

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = [1, 2, 3])
```

Specify values for each column.

```
df = pd.DataFrame(
    [[4, 7, 10],
     [5, 8, 11],
     [6, 9, 12]],
    index=[1, 2, 3],
    columns=['a', 'b', 'c'])
```

Specify values for each row.

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
df = pd.DataFrame(
    {"a": [4, 5, 6],
     "b": [7, 8, 9],
     "c": [10, 11, 12]},
    index = pd.MultiIndex.from_tuples(
        [('d', 1), ('d', 2), ('e', 2)],
        names=['n', 'v']))
```

Create DataFrame with a MultiIndex

Method Chaining

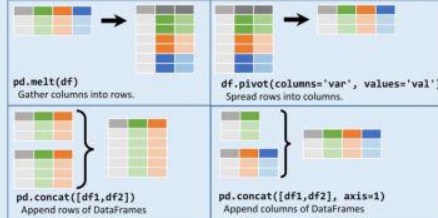
Most pandas methods return a DataFrame so that another pandas method can be applied to the result. This improves readability of code.

```
df = (pd.melt(df)
      .rename(columns={
          'variable': 'var',
          'value': 'val'})
      .query('val >= 200'))
```

Tidy Data – A foundation for wrangling in pandas



Reshaping Data – Change the layout of a data set



```
df.sort_values('mpg')
Order rows by values of a column (low to high).

df.sort_values('mpg', ascending=False)
Order rows by values of a column (high to low).

df.rename(columns = {'y': 'year'})
Rename the columns of a DataFrame

df.sort_index()
Sort the index of a DataFrame

df.reset_index()
Reset index of DataFrame to row numbers, moving index to columns.

df.drop(['Length', 'Height'], axis=1)
Drop columns from DataFrame
```

Subset Observations (Rows)

```
df[df.Length > 7]
Extract rows that meet logical criteria.

df.drop_duplicates()
Remove duplicate rows (only considers columns).

df.head(n)
Select first n rows.

df.tail(n)
Select last n rows.

df.sample(frac=0.5)
Randomly select fraction of rows.

df.sample(n=10)
Randomly select n rows.

df.iloc[10:20]
Select rows by position.

df.nlargest(n, 'value')
Select and order top n entries.

df.nsmallest(n, 'value')
Select and order bottom n entries.
```

	Logic in Python (and pandas)	
<	Less than	<=
>	Greater than	df.column.isin(values)
==	Equals	pd.isnull(obj)
<=	Less than or equals	pd.notnull(obj)
>=	Greater than or equals	obj.isna(), df.any(), df.all()

Subset Variables (Columns)

```
df[['width', 'length', 'species']]
Select multiple columns with specific names.

df['width'] or df.width
Select single column with specific name.

df.filter(regex='regex')
Select columns whose name matches regular expression regex.
```

regex (Regular Expressions) Examples	
\"	Matches strings containing a period '.'
\"length\"	Matches strings ending with word 'length'
\"Sepal\"	Matches strings beginning with the word 'Sepal'
\"^\"	Matches strings beginning with 'v' and ending with 1,2,3,4,5
\"^\"	Matches strings except the string 'Species'

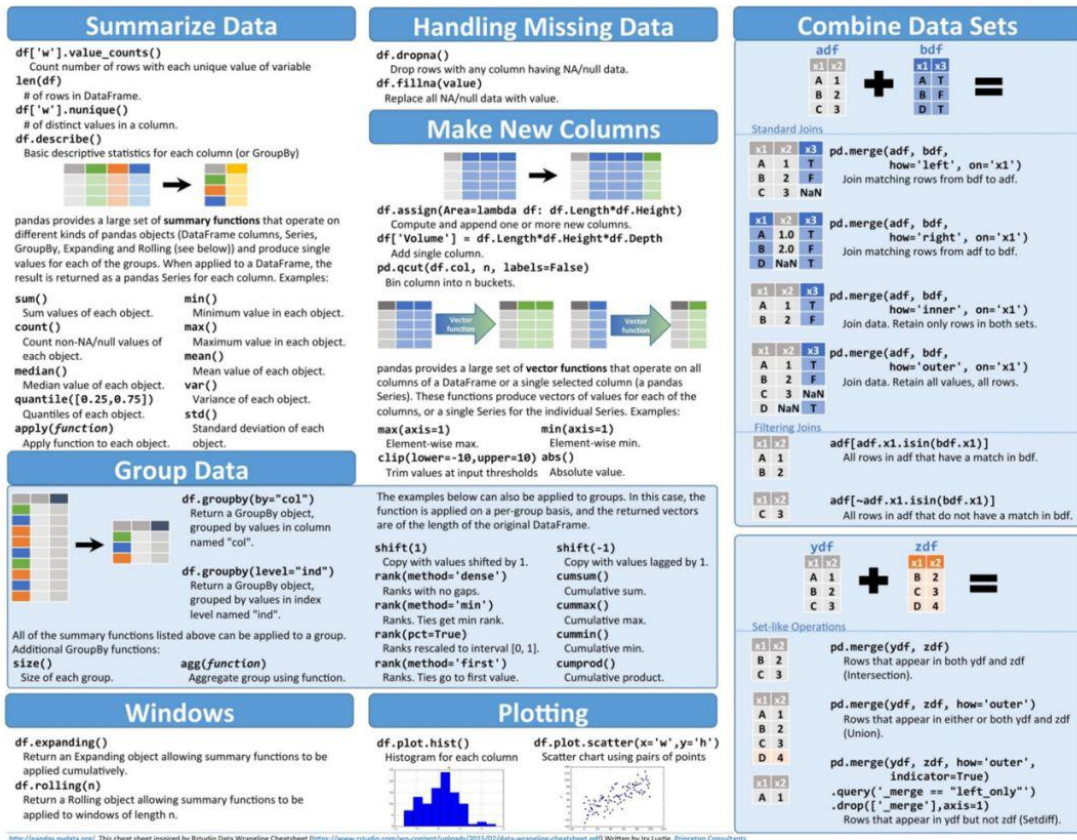
```
df.loc[:, 'x2': 'x4']
Select all columns between x2 and x4 (inclusive).

df.iloc[:, [1, 2, 5]]
Select columns in positions 1, 2 and 5 (first column is 0).

df.loc[df['a'] > 10, ['a', 'c']]
Select rows meeting logical condition, and only the specific columns.
```

<http://pandas.pydata.org> This cheat sheet inspired by Kinoshita Data Wrangling CheatSheet <https://www.kinoshita.com/data-wrangling-cheat-sheet/> Written by the author, <https://www.kinoshita.com/data-wrangling-cheat-sheet/>





⑤ 数据维度

群体风险感知数据维度



我们从静态数据和动态行为数据两个角度来构建立体多维群体行为画像：



- **静态数据**

- 注册信息画像：时间，设备，ip，渠道
- 登录信息画像：时间，设备，ip，版本

- **动态行为数据**

假设 A、B、C 三个账号由同一个人控制，那么每次有活动他们就一起来了，可知道他们是一伙的，如何把这三个人每次活动一起来这件事儿，通过数据来描述好加工出来，供后面的模型使用，这是我们当时研究的重点。

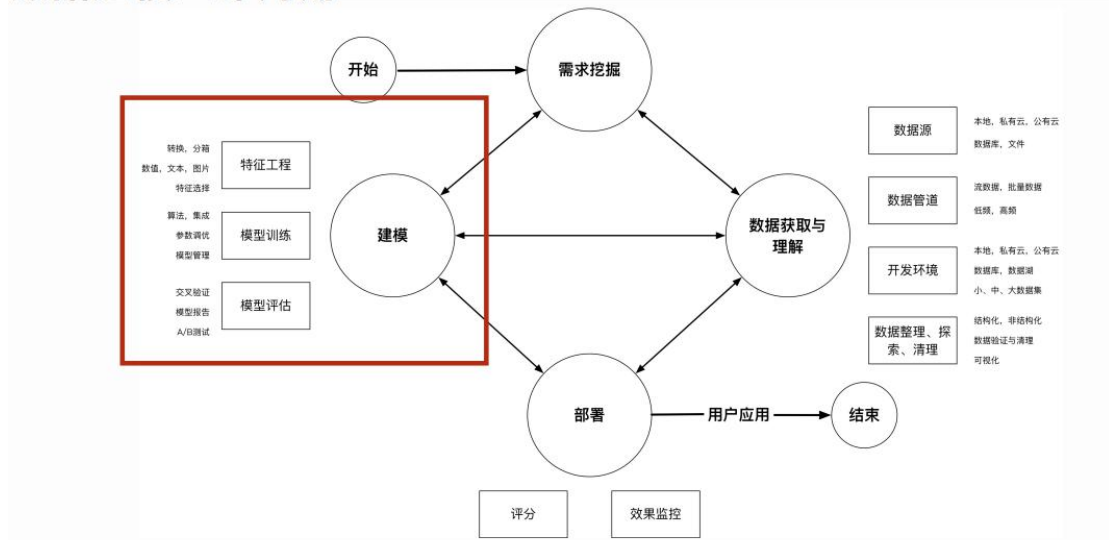
动态行为数据包括：

- **浏览行为画像**：时间，设备，ip，渠道（同静态相同，但是是账号之间的比较）
- **下单行为画像**：商品信息，时间信息，设备，ip，地址，优惠券，支付
- **领券行为画像**：频率，使用周期

■ 建模



数据建模生命周期



建模的三个主要模块：

- **特征工程**：特征处理，数值，文本，图片，特征选择
- **模型训练**：参数调优，模型管理
- **模型评估**：交叉验证，模型报告，A/B 测试

① 目标

- 确定要输入模型的数据特征
- 选择一个最能满足项目要求的模型（如果这个项目以准确率为最核心的 KPI，可能 A、B、C 三个模型都各有所长，有的效率高、有的速度快、有的召回率高，这时就需要按要求选择）
- 模型参数调优，针对生产环境做优化（A/B TEST）

② 实现方法

- 特征工程



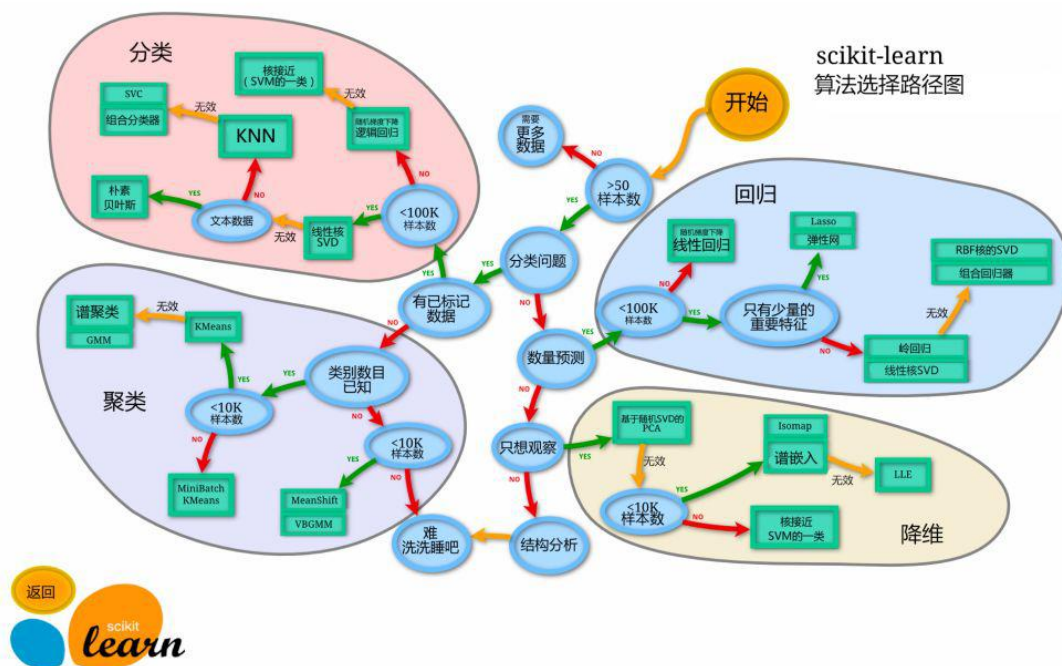
- 模型选择（分类、聚类、有监督、无监督、深度学习、强化学习）
- 模型训练（如何调优）
- 评估模型在生产环境的适应性

③ 交付物

- 特征集
- 模型报告

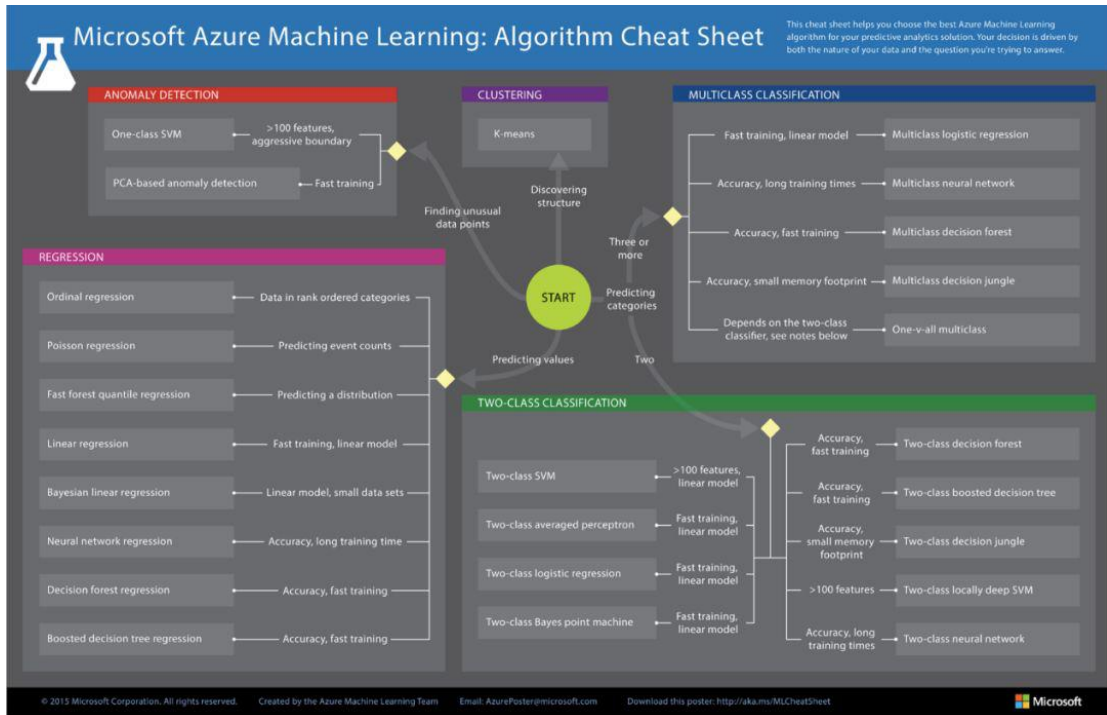
④ 算法模型路径选择

- Sklearn 算法选择路径图（这是目前网上比较流行的模型选择路径，大家可以根据业务的具体需求来选择模型）；



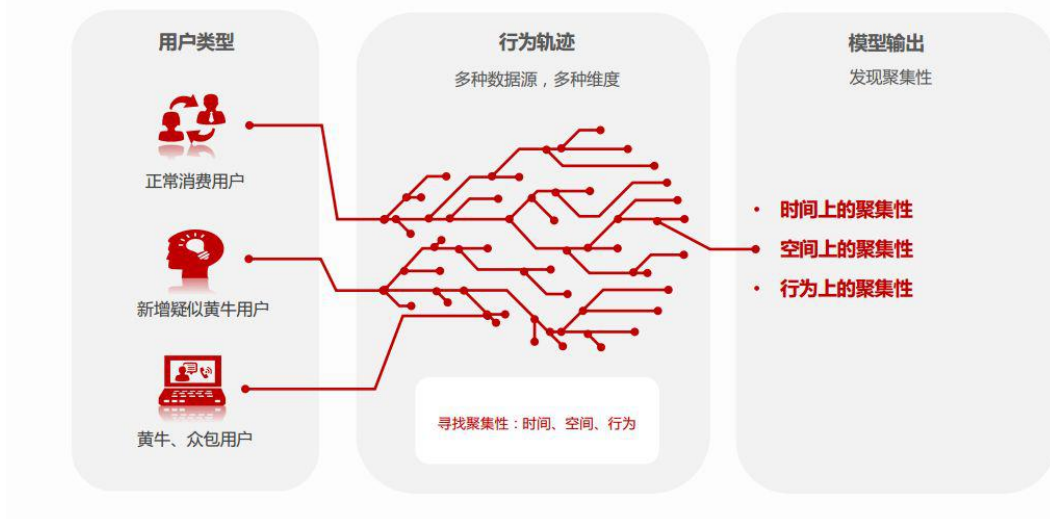
- 微软的模型选择路径：





⑤ 群体风险识别建模

群体风险识别建模



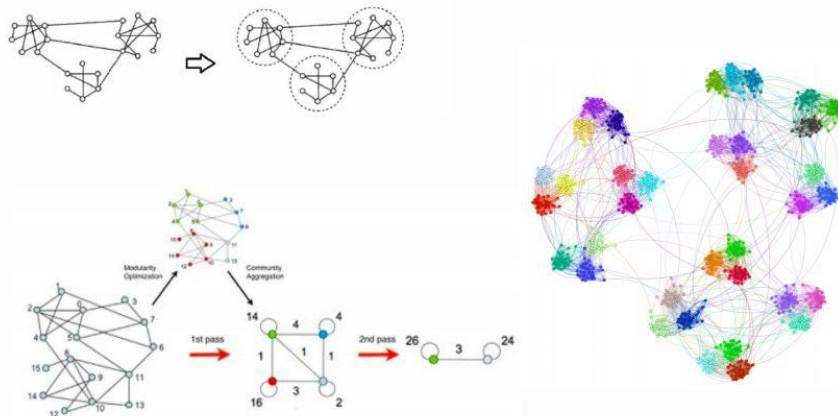
- 。 **用户类型**：正常消费，新增疑似黄牛用户，黄牛&众包用户



- **行为轨迹**：多种数据源，多种维度，寻找在时间，空间，行为的聚集性
- **模型输出**：发现时间，空间，行为上的聚集性

⑥ 静态特征识别

静态特征识别



- 关系图 聚类 :Louvain ,Fast Greedy ,Markov Cluster ,Info Map , Walktrap。
- 图计算的社区发现（前人的一些经验）：

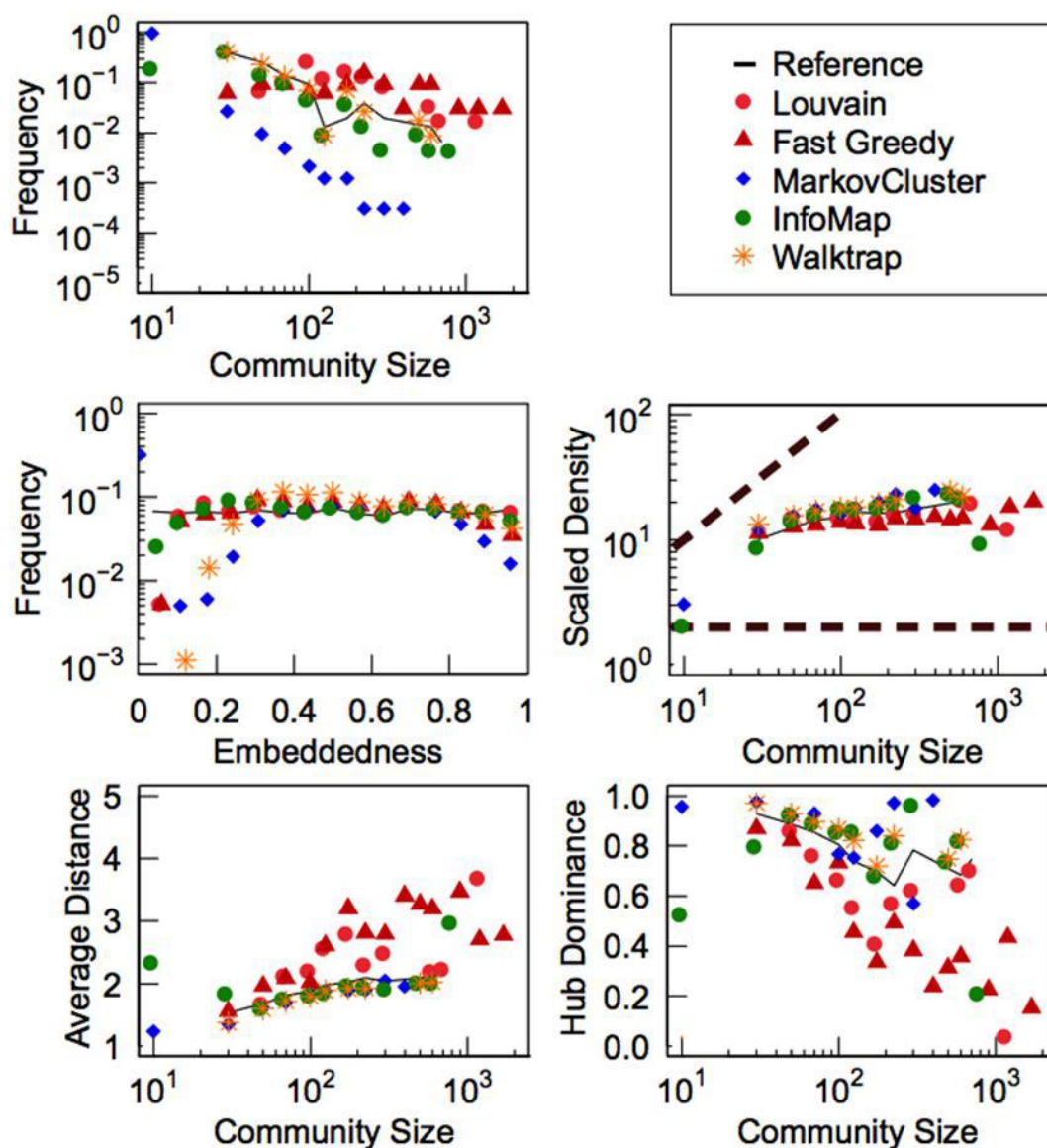


Algorithm	Year	Complexity
CFinder	2005	
LPA	2007	$O(m)$
LFM	2009	$O(n^2)$
EAGLE	2009	$O(n^2 s)$
GIS	2009	$O(n^2)$
HANP	2009	$O(m)$
GCE	2010	$O(mh)$
COPRA	2010	$O(v m \log(\frac{vm}{n}))$
NMF	2010	$O(K n^2)$
Link	2010	$O(n k_{max}^2)$
SLPA	2011	$O(T m)$
BMLPA	2012	$O(n \log n)$

。 常用算法比较：



Algorithm	$n = 10^4$	$n = 10^5, k_{max} = 10^3$	$n = 10^5, k_{max} = 3 \times 10^3$
Louvain	0.80	0.78	0.80
Fast Greedy	0.59	0.66	0.67
MarkovCluster	0.83	0.87	0.80
InfoMap	0.88	0.93	0.91
Walktrap	0.77	0.79	0.78



⑦ 动态特征识别



动态特征识别

时间	活动场景	下单账号
2019-05-21 11 : 00-12 : 00	A品类B商品C优惠	pin1,pin2,pin3,pin4,pin5,pin6,pin7
2019-05-21 12 : 00-13 : 00	A品类B商品C优惠	pin0,pinA,pinB,pinC,pinD
2019-05-21 13 : 00-14 : 00	A品类B商品C优惠	pinF,pinE,pinH,pinG,pinB,pinM,pinL,pinX
2019-05-22 08 : 00-09 : 00	X品类Y商品Z优惠	pin3,pinE,pinB,pinH
2019-05-22 09 : 00-10 : 00	X品类Y商品Z优惠	pin1, pinC,pinX,pinV,pinZ,pinR,pinU,pinI
2019-05-23 00 : 00-01 : 30	M品类L商品N优惠	pin4,pinC,pin3,pinX,pinD,pin9,pin4,pinF
2019-05-23 01 : 30-02 : 30	M品类L商品N优惠	pin23,pin5,pinG,pinH,pinJ
.....		

这是我们归结出来的数据底层形式，包括：时间，活动场景，下单账号几个维度。

群体划分(聚类问题)：距离的选择对聚类的结果影响巨大

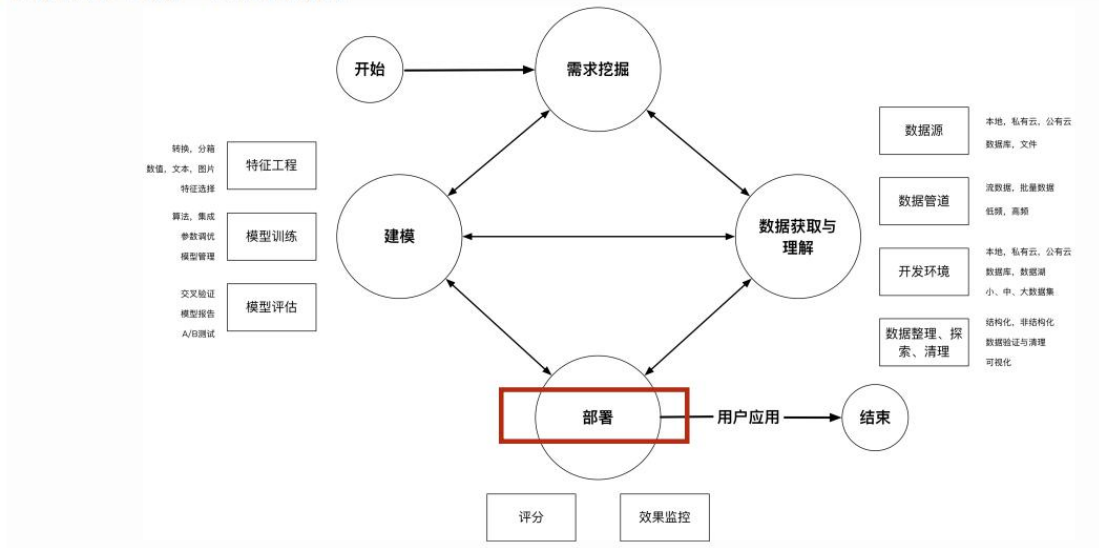
动态特征识别（群体划分）

算法名称	算法参数	范围	距离度量
K-means	簇数	大样本、聚类数目中等	点之间的距离公式
spectral clustering	簇数	样本数中等、聚类数目小	图距离
ward hierarchical clustering	簇数	大样本、聚类数目大	点之间的距离
agglomerative clustering	簇数、链接类型、距离	大样本、聚类数目大	任意成对点线图之间的距离
DBSCAN	半径大小、最低成员数	样本数中等、聚类数目中等	最近点之间的距离
birch	分支因子、阈值、可选全局集群	大样本、聚类数目大	点之间的欧式距离

部署



数据建模生命周期



部署的时候需要考虑的两个关键因素：**评分和效果监控**。

群体风险感知

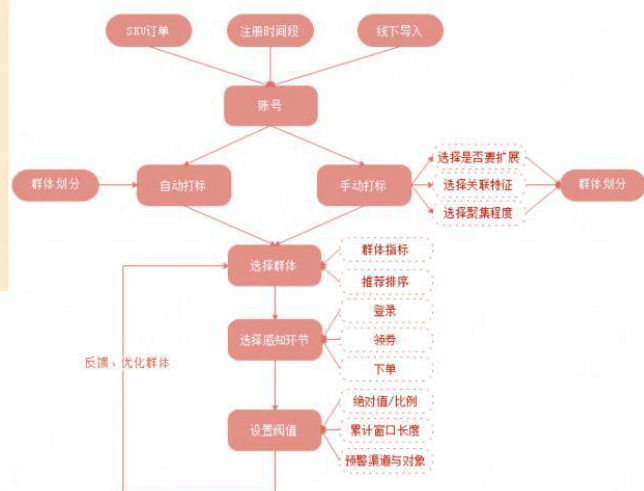
监控指标体系与异常发现

输出结果形式

- 线索账号的线索：某个特定群体行为
- 群体维度的线索：多个群体聚集活跃

输出结果内容

- 聚集性指标，灵活制定规则
- 提前发现风险点，预警后续环节
- sku、优惠券、订单、相关账号
- 新的套利、作弊方式、交易漏洞



- 目标：把模型和配套的数据管道部署到生产环境
- 实现方法：**推荐 api 接口形式部署模型**



- 交付物：监控报表，模型说明文档（输出和部署细节），数据管道说明文档

■ 用户应用

① 目标

- 完成项目交付

② 实现方法

- 验证部署好的系统的功能性和准确率，满足业务需求
- 所有文档都要完成并进行审阅
- 项目交接工作可能是交给 IT 运维部分或者是用户的数据团队甚至可能是业务团队

③ 交付物

- 项目交付报告

④ 业务应用架构：



业务应用架构

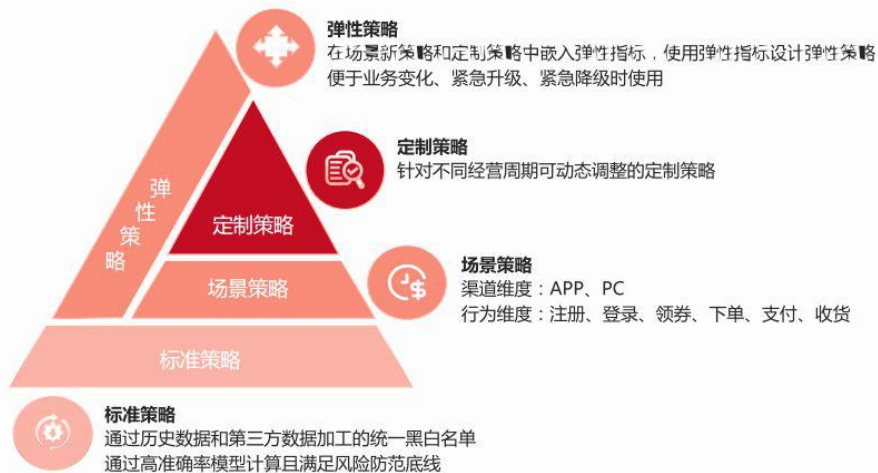


用规则引擎（实时计算、多重规则、实时指标），模型引擎（模型管理、模型部署、模型运算），决策引擎（策略管理、策略可视化、策略路由）连接数据中心和用户行为。

⑤ 设计多层次风控场景策略



设计多层次风控场景策略



- **标准策略**：标配，黑白名单，评分
- **定制策略**：针对不同的经营周期动态调整策略
- **场景策略**：针对渠道，行为维度
- **弹性策略**：场景策略和定制策略中增加弹性指标，用于业务变化，
紧急升级和降级

今天的分享就到这里，谢谢大家。



信贷业务风控策略简介



分享嘉宾：韩士渊 百融云创 高级风控总监

编辑整理：张祥

内容来源：百融云创 & DataFun Talk

出品社区：DataFun

注：欢迎转载，转载请注明出处

导读：大家好，今天分享的主题是信贷业务风控策略。风控业务主要经历了几个阶段：

- 规则：直接判断通过，或不通过。
- 数据：可以通过客户的资产，流水，来判断客户的资质优劣。



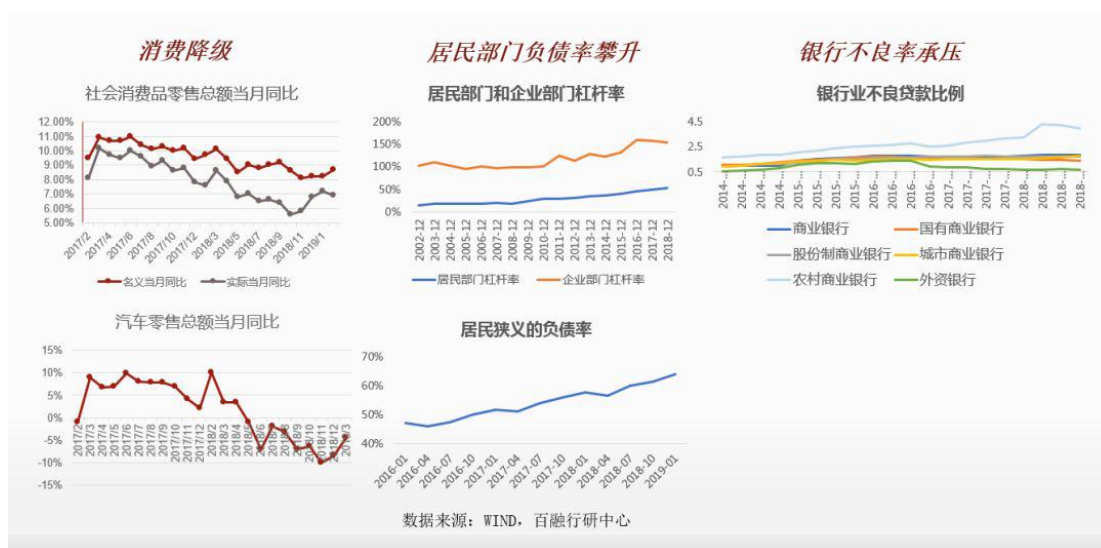
- 。模型：通过数据分析、数据挖掘，找到相应的规律，识别出人工难以找到的部分人群。

但是，数据是有限的，成本很高，会限制风控的上限；同时，如何有效的结合数据、规则、模型，来实现业务目标，这就需要风控策略来完成。

本次分享，将介绍如何在信贷业务中利用数据、规则、模型等完善风控策略，包括原有风控流程及规则优化、定价策略、额度策略等内容。

背景

1.1 消费信贷行业背景

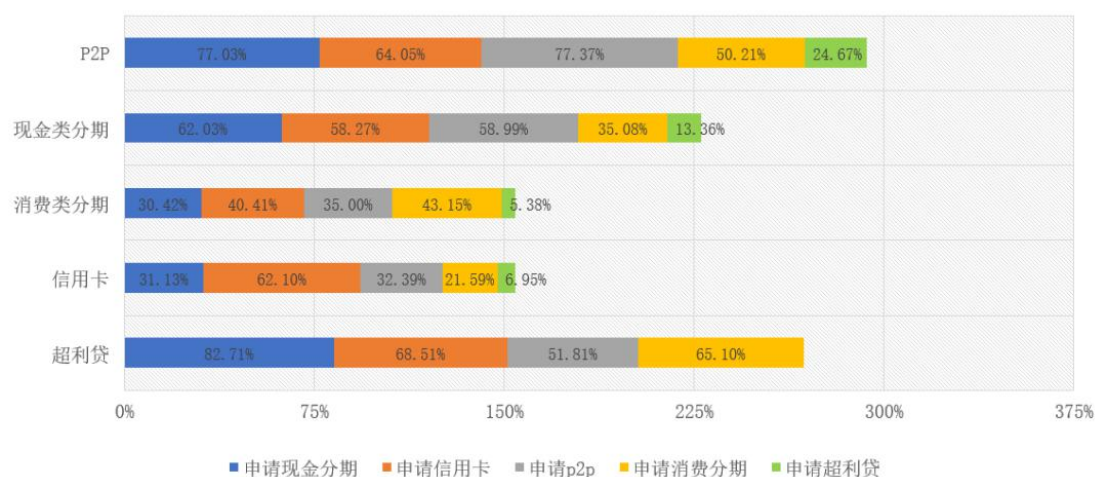


首先给大家普及下消费信贷行业的情况，目前的消费信贷市场和 2 年前的不太一样。2 年甚至 3 年以前，它是一个蓬勃发展的阶段，但是现在出现了一个情况，就是消费降级，社会消费品零售总额、汽车零售总额增长放缓，居民狭义的负债率攀升，以及银行不良贷款率增长态势。信贷市场相



较前几年的市场发生了变化，前几年属于资产红利，风控的价值本质上是资产获利的多少；目前而言，风控的价值是有效降低风险，减少亏损。因此，对于成本的控制，对于风控的效果，需要有更精细化的应用和实践。

1.2 消费信贷行业客户渗透

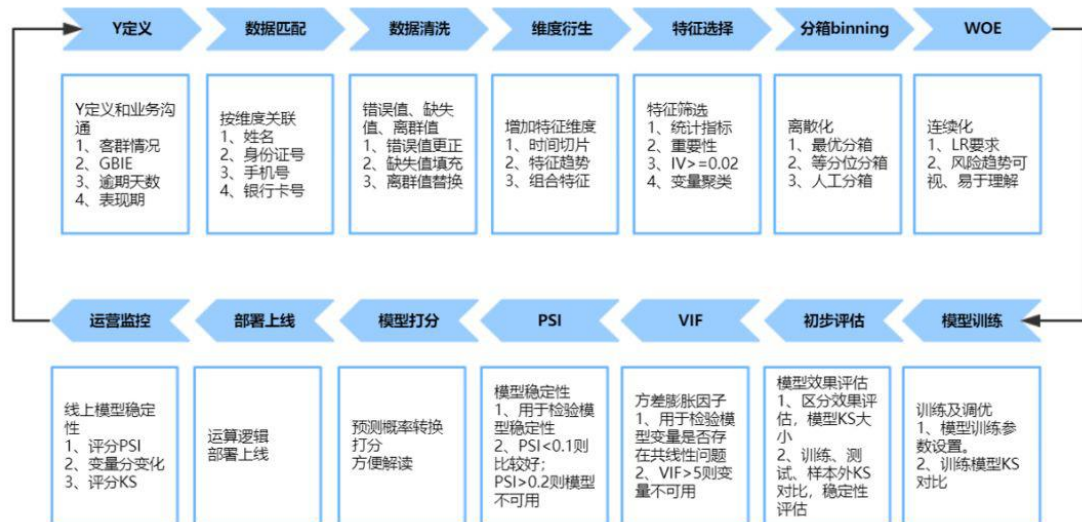


客户渗透率

横轴表示渗透占比，横轴越长，渗透率越高；单个颜色越长，表明用户群重合也越高。当某个产品受到监管，会导致其客户流失，随之带来金融风险问题。以 P2P 和超利贷为例，正常情况下，P2P 的客户应该比超利贷好，但是由于市场监管的变化，二者的客户质量已经差不多了。所以，当市场规则发生变化时，原来的规则和模型都失效，因此，风控策略就成为了最后一道关卡。

1.3 传统评分卡的开发流程





评分卡开发流程

① 目标定义

- 定义风控业务的目标，要求熟悉业务逻辑，好坏客户的定义。

② 数据的整合加工

- 数据：包括用户的姓名、身份证号、手机号、银行卡号，购物记录等
- 清洗：错误值、缺失值、离群值
- 衍生变量的处理：引入其它维度的数据，以及做特征组合

③ 特征选择、调优、效果评估

- 特征选择：利用统计显著性、变量重要度、IV 指标，变量聚类等算法来挑选重要的变量，通常还需要采用分箱的方式，将连续值离散化，增加变量的鲁棒性，增加了变量的整体稳定性。
- 模型调优：在训练集上调整模型参数，对比 KS 值的变化，选择 KS 最大的模型。



- 效果评估：对比训练集、测试集、样本外数据集的 KS 值，查看模型是否稳定。
- 共线性：此外，还需要检验模型变量之间是否存在共线性问题，一般计算 VIF 值，如果 $VIF > 5$ ，表明存在共线性问题，该变量不可用。
- 模型稳定性：计算 PSI 值，如果 $PSI < 0.1$ ，则模型稳定性较好，如果 $PSI > 0.2$ ，则模型稳定性较差，不可用。

④ 模型打分、部署、监控

- 对变量每个分箱都打一个分值，形成一张评分卡（相较于机器学习，优点是可解释性强），最后上线部署。

1.4 机器学习模型开发流程



机器学习开发流程

相较于评分卡开发流程，差异如下：



- ① 机器学习中人工介入少；
- ② 可解释性差；
- ③ 重点是调整参数，避免模型过拟合。

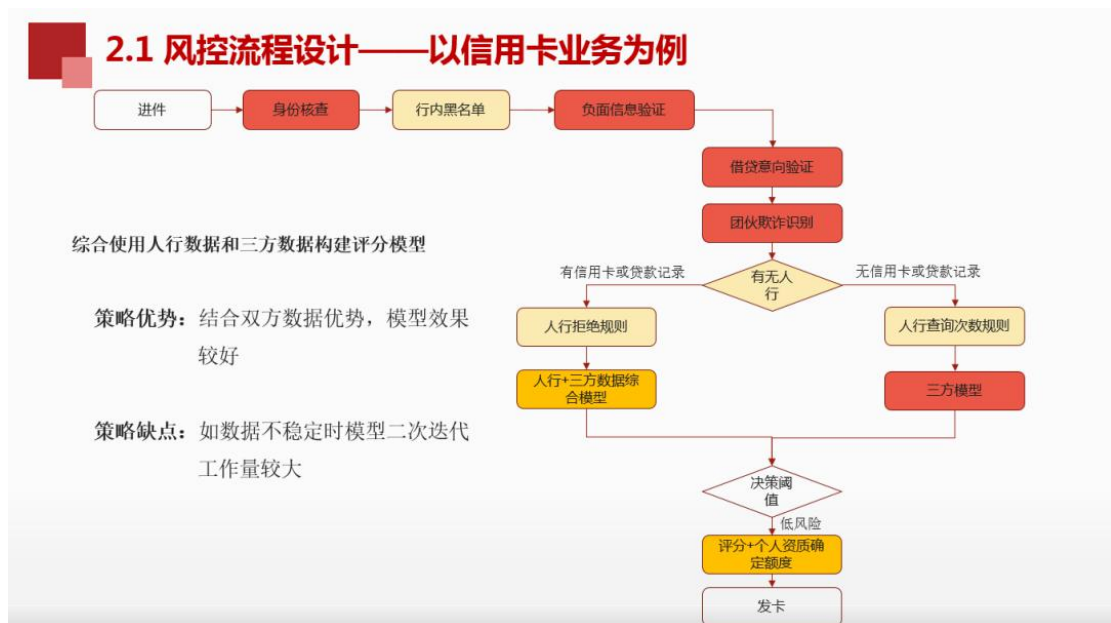
■ 贷前风控流程与策略

2.1 风控流程设计

目标是发现风险点（包括：信用卡欺诈、团伙窝案、高危用户等），降低风险；同时降低成本、提升效率。

银行的风控流程，以某四大行信用卡业务为例。

案例 1：



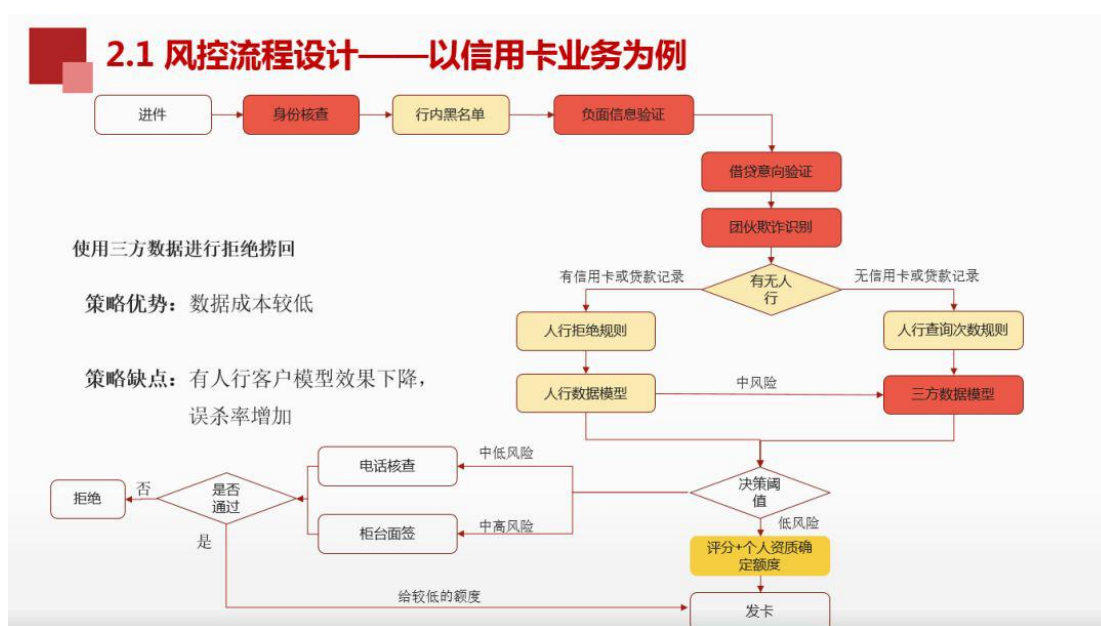
- ① 身份核查：验证身份的真实性，是否为本人、是否有欺诈等行为。
- ② 剔除其它点影响：行内黑名单、负面信息验证。



③ 将成本低的借贷意向验证放在成本高的团伙欺诈识别前面，达到节省成本的作用。

④ 结合人行拒绝规则、人行数据以及第三方数据建模，其优点是结合双方数据优势，模型效果会较好。但数据不稳定时，模型二次迭代工作量较大。

案例 2：

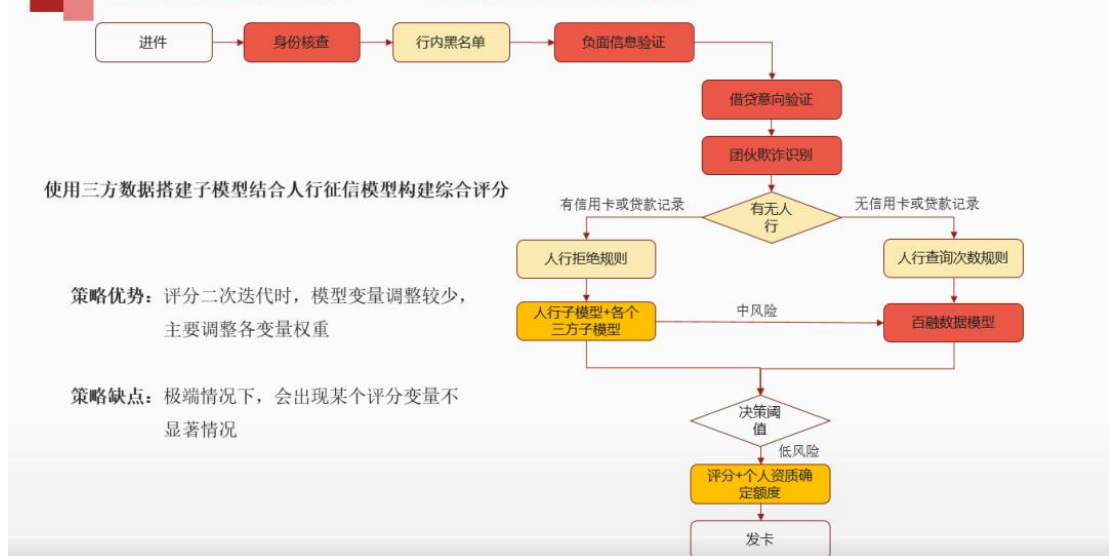


相较于案例 1，案例 2 较为保守，没有将人行数据和三方数据结合起来做来模型，对于拒绝的用户再次做人工审核，将符合的用户再次捞起来。这样做数据成本低，但是有人行客户模型效果下降，误杀率增加。

案例 3：



2.1 风控流程设计——以信用卡业务为例



同案例 1 的区别：将各个数据构建子模型，然后整合为一个综合模型，如收入数据、支出数据建模。差异在于客群建模的差异，将不同的客群用户分开做模型。这样在评分二次迭代时，模型变量调整较少，主要调整各变量权重即可。但是在极端情况下，会出现某个评分变量不显著的情况。

综合上述三个案例，风控策略并不是完全依赖于成本，还要依赖于实际业务情况和业务目标。很多时候，风控流程是根据业务情况来进行的。

2.2 利率策略和额度策略



2.2 利率策略&额度策略

在完成模型构建后，对每位客户进行打分，统计每个分段内的坏账率，可得到以下决策表：

评分区间	通过率	人数	区间坏账率 (pi)
<551	100%	3000	28.50%
[551, 582)	95%	3000	17.72%
[582, 601)	90%	3000	13.23%
[601, 617)	85%	3000	10.67%
[617, 631)	80%	3000	8.76%
[631, 644)	75%	3000	7.08%
[644, 655)	70%	3000	6.93%
[655, 667)	65%	3000	6.18%
[667, 679)	60%	3000	5.04%
[679, 691)	55%	3000	4.71%
[691, 701)	50%	3000	3.69%
[701, 717)	45%	3000	3.36%
[717, 730)	40%	3000	3.03%
[730, 745)	35%	3000	2.46%
[745, 760)	30%	3000	2.28%
[760, 777)	25%	3000	1.80%
[777, 794)	20%	3000	1.44%
[794, 818)	15%	3000	1.14%
[818, 849)	10%	3000	0.90%
>=849	5%	3000	0.51%

完成模型构建后，对每位客户打分，统计每个分段内的坏账率，以控制收益与成本。怎么给合适的利率和额度呢？

① 利率策略：

假设：

- 预期收益率： r_0
- 利率在区间中变化时，不同分段的坏客户率变化不显著
- 评分有较好的排序能力，每一分段对应的坏客户占比有显著差异，第*i*段评分区间对应的违约率为 p_i

可能的收益-可能的损失=预期收益

$$A_0 \times r \times (1 - p) - A_0 \times p = A_0 \times r_0$$

$$A_i \times r_i \times (1 - p_i) - A_i \times p_i = A_i \times r_0$$

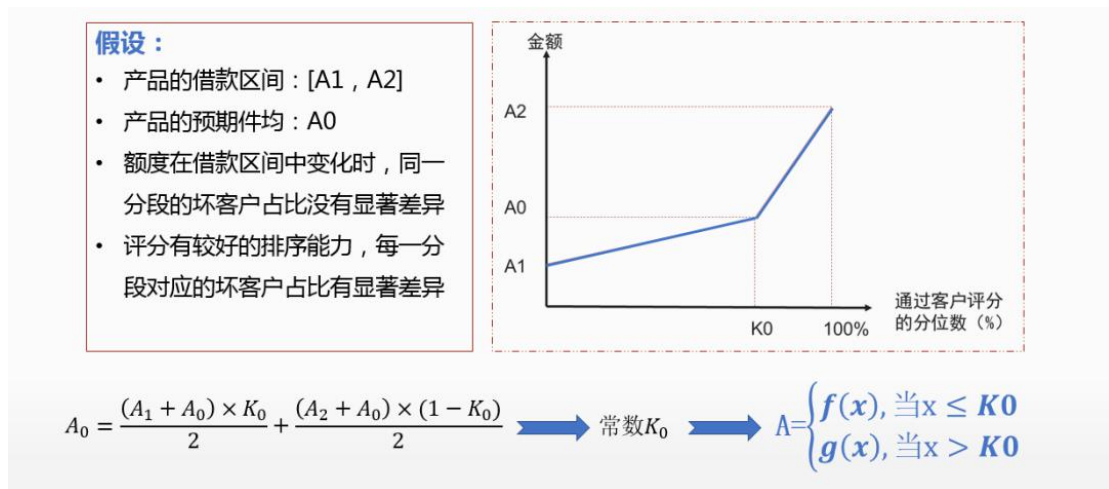
$$r_i = \frac{r_0 + p_i}{1 - p_i}$$

风险与利率计算公式

A 表示额度，r 表示预期收益率，p 表示坏账率，对每个评分段分别计算预期收益 r_i ，但通常情况下，利率是固定的，当分数在某个阈值时，就直接拒绝掉。



② 额度策略：



风险情况与平均每件额度

额度策略本身受限于产品设计、客户需求及竞品情况，结合自身成本和风险偏好，可初步确定产品的额度区间 $[A_1, A_2]$ 和件均 A_0 。由于右图中两个梯形的面积应该是相等的，因此，可以得到关于 A_0 的计算表达式，由于 A_0 、 A_1 、 A_2 都是已知的，因此我们能计算出 A_0 对应的常数 K_0 ，这样就可以把右图中蓝色的折线拟合出来，即相对最优的一个解。可以实现，根据不同的分位数，给不同的额度。那么，这么做合理吗？它是需要满足一定的假设条件的：

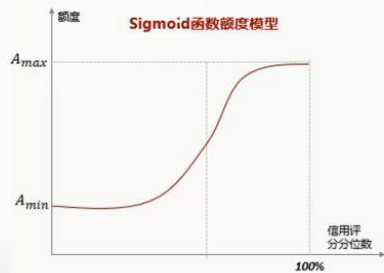
- 额度在借款区间中变化时，同一分段的坏客户占比没有明显差异；
- 评分有较好的排序能力，每一分段对应的坏客户占比有显著差异。

③ 额度策略优化：



信用水平结合客户收入状况，从**违约风险**、**偿债能力**两方面对客户进行差异化定額

- 基于产品的额度区间、件均、信用评分，采用额度模型确定**基础风险额度**



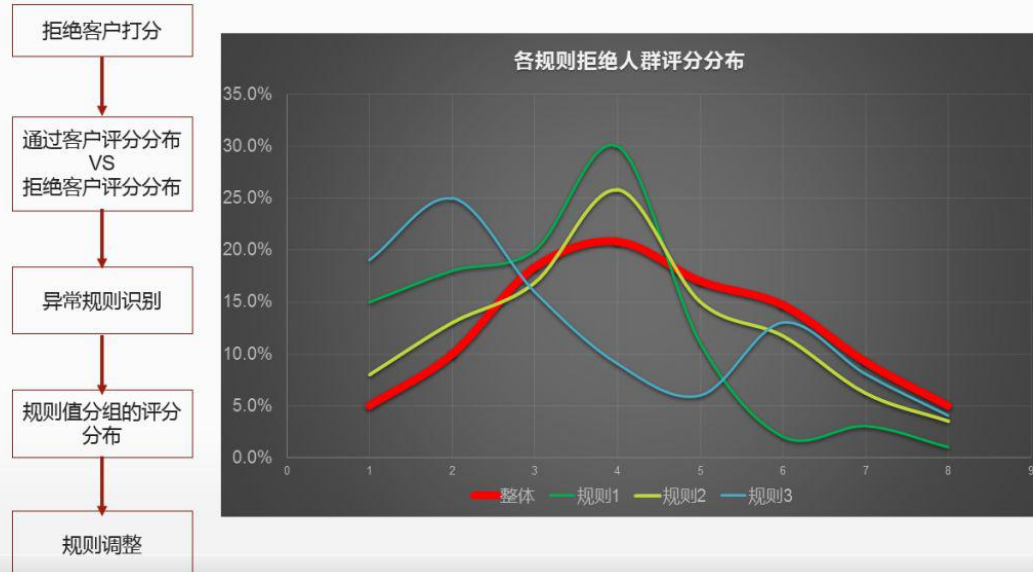
- 综合考虑风险评分与个人收入/资产/流水，基于收入差异化调整基础额度

月收入(千)	0	1	2	3	4	5	6	7	8	9	...
基础风险额度	0	1.2	2.4	3.6	4.8	6	7.2	8.4	9.6	10.8	...
1.5	[600,630]	1.5	1.2	1.5	1.5	1.5	1.5	1.5	1.5	1.5	...
3.1	[630,660]	1.5	1.9	3.1	3.1	3.1	3.1	3.1	3.1	3.1	...
5.4	[660,690]	1.1	3.2	3.9	4.7	5.4	5.4	5.4	5.4	5.4	...
8.6	[690,720]	1.3	5.2	5.7	6.3	6.9	7.4	8	8.6	8.6	...
12.8	[720,750]	1.3	7.7	8.3	8.8	9.4	10	10.5	11.1	11.7	...
17.2	[750,780]	1.3	10.3	10.9	11.4	11.9	12.4	13	13.5	14	...
21.1	[780,810]	1.2	12.7	13.2	13.7	14.3	14.8	15.3	15.9	16.4	...
24.6	[810,840]	1.4	14.8	15.3	15.8	16.3	16.9	17.4	17.9	18.4	...
27.4	[840,870]	1.5	16.5	17	17.5	18	18.5	19.1	19.6	20.1	...
29.2	[870,900]	1.5	17.5	18	18.5	19	19.5	20	20.5	21.1	...
29.9	[900,1000]	1.5	17.9	18.5	19	19.5	20	20.5	21	21.6	...

额度策略优化

采用 sigmoid 来替代分段函数，确定基础风险额度。对于大额借贷，还是考虑用户的偿债能力，即收入，资产，流水等指标，先算出基础风险额度，再结合收入等指标，差异化调整基础额度。

2.3 风控规则有效性诊断



各规则拒绝人群评分分布

怎么确认规则的有效性和调整？

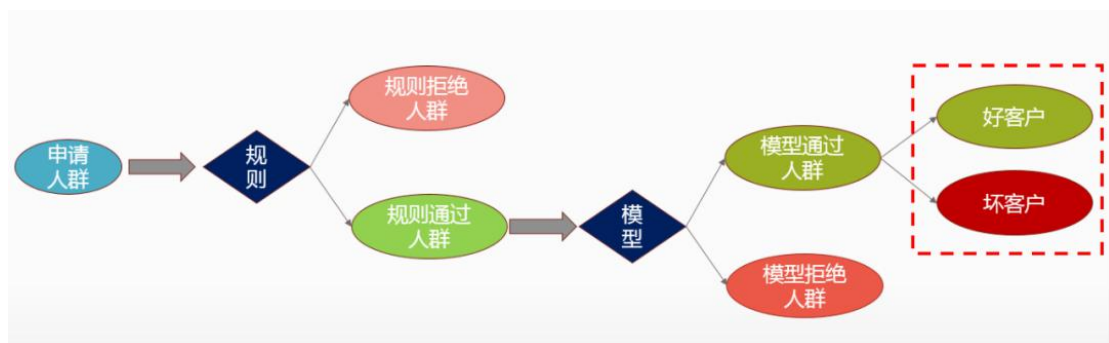


首先对拒绝客户进行打分，然后对比‘通过客户评分分布’和‘拒绝客户评分分布’，找出异常规则，再把各规则进行分组评分分布对比，最后进行规则调整。

如上图，对拒绝的客户重新打分，与整体样本中拒绝用户评分对比：

- 规则 1，效果最好，最低分段的用户拒绝率高，高分段的用户拒绝率。
- 规则 2，效果不明显，与整体数据的分布差异不大，效果不明显。
- 规则 3，太过于波动，不稳定。

2.4 信用评估模型的构建与优化——模型优化



风控建模流程图

基础的风控建模流程如上。在进行模型迭代时，主要利用通过样本进行模型构建，被迭代的模型拒绝率较高；如不考虑模型拒绝人群，进行拒绝推断，则新模型的应用效果会有明显下滑，且多次迭代后，新模型的效果提升比例会越来越小。因此重要的一点在于，如何做拒绝推断，找到之前被规则淹没的特征。



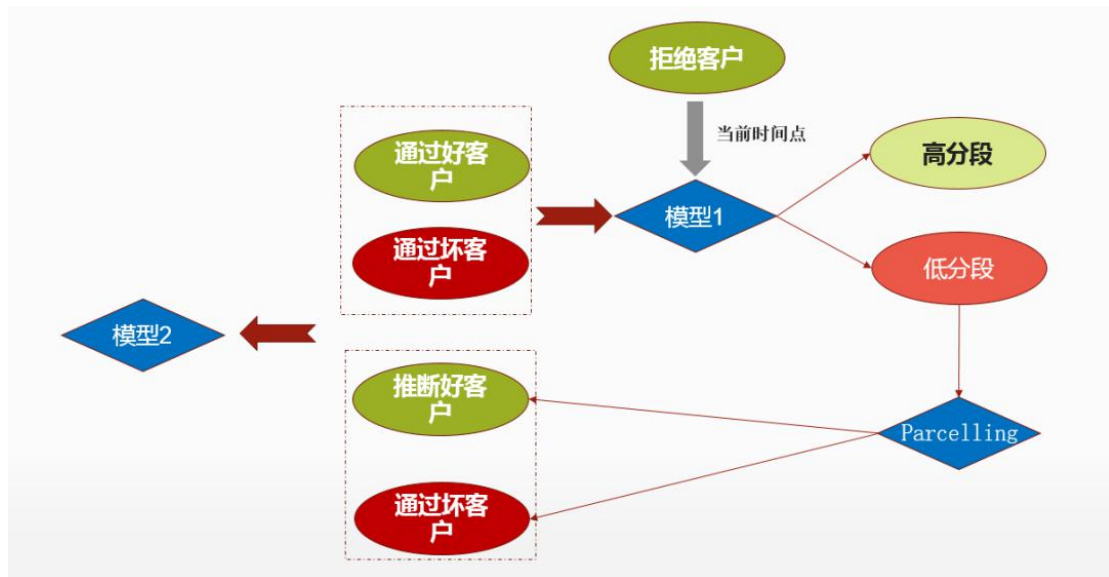


解决样本淹没问题的三种方案

为了解决样本淹没问题，将规则拒绝的样本加入模型训练阶段，有三种方案：

- ① 比例分配：将拒绝对象随机划分为“好”和“差”的账户，再次带入评分流程中，构建一个模型；
- ② 简单增强法：对拒绝客户打分，并选取某个 cutoff 点进行区分，cutoff 点的两边分别为“好”和“坏”客户，代入模型中迭代；
- ③ Parcelling：是结合比例分配和简单增强法，对拒绝客户打分，对每个分段按照比例进行好坏客户划分，再次代入模型进行迭代。





模型优化

我们的做法是使用通过的样本和通过的坏客户构建模型 1，拿当前时间点跑客户评分，分为高分段和低分段，把原来拒绝客户跑完模型后高分段的样本剔除掉，对低分段特征做 parcelling，然后推断好客户和坏客户，合在一起再做模型 2，然后不断的迭代这个模型，这时跑出来的模型，要比刚刚介绍的模型 1 方法要好。但是这种拒绝推断是没办法从样本上解决样本有偏的问题，只能从某种程度上解决。

以上就是今天要分享的内容，谢谢大家。



机器学习在信贷风控建模中的优势和挑战



机器学习在信贷风控建模中的 优势和挑战

罗灿



移动金融智选平台

北京融联世纪信息技术有限公司

地址：北京市海淀区中关村丹棱街甲1号互联网金融中心21层

邮编：100086

电话：(86-10) 82625755

电话：(86-10) 82625755-8080

文章作者：罗灿 融 360

编辑整理：郑智献

内容来源：DataFun AI+ Talk

出品社区：DataFun

注：欢迎转载，转载请注明出处

在国内国外金融风控领域大致分为两个流派，其中一派为具有统计学背景的人，分布在银行、金融消费公司等传统的金融领域，偏好评分卡进行建模。另外一派则是具有互联网背景的新兴探索者，将机器学习、深度学习等方法运用于金融风控领域。融 360 是 ToB 的金融风控企业，不仅跟传统的金融公司有合作，还和京东、百度金融、滴滴等互联网背



景的公司有交集。本次演讲内容为这两种流派优劣势对比及机器学习、深度学习在金融风控领域的实践应用。

一、信用评分卡模型

评分卡模型虽然是一个简单的权重线性加和的回归模型，但在传统金融风控领域使用了上百年。

● 评分卡模型处理



评分卡模型的本质是线性回归模型

$$Y = \sum_{i=0}^n \theta_i x_i + b$$

特征权重设定：

- 根据经验设定
- 根据单特征区分能力设定
- 设定loss，求解最优化问题

1、评分卡模型中的特征筛选、参数求解

评分卡模型对特征筛选有以下偏好：

- 1) 特征覆盖率高，通常达到 70%以上；
- 2) 特征与逾期率为线性强相关；
- 3) 特征随着时间迁移，其分布保持稳定；



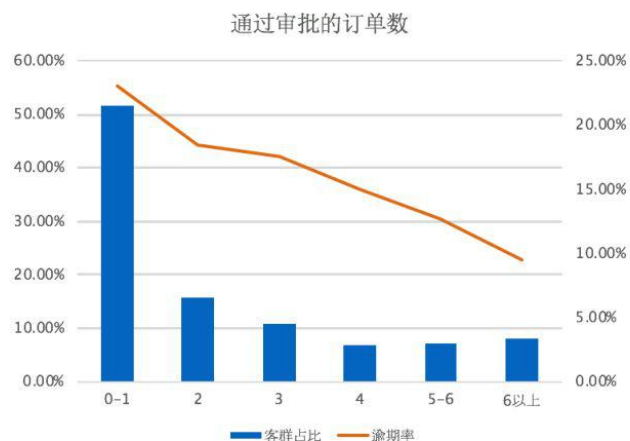
4) 特征变量与风险趋势的相关性有明显的可解释性，并且入模变量少，通常为 8-12 个。

以上特征筛选的偏好是为了保证入模的特征变量的稳定性、有效性，比如会有黑名单、负债、资产等特征变量。总的来说，在少样本量及强特征的情况下，使用评分卡这种简单的模型能够很好的保证稳定性和有效性。

● 特征筛选

融360
Rong360.com

- 特征覆盖率比较高
- 逾期率线性
- 强相关
- 随时间迁移分布稳定
- 特征需要可解释
- 入模特征8-12个



关于参数求解，在机器学习中通常设定损失函数，将其转化为凸问题，再进行求解。但是在评分卡模型中，若行业样本非常稀缺，对参数训练收敛有一定的困难。若当前样本量为 0，可根据专家的历史经验进行权重设定。若当前样本为几百个，可根据单特征区分能力比如 KS / IV 值等进行权重设定。

2、评分卡模型的非线性 / 交叉特征处理



线性回归模型有一定的劣势 ,比如不能解决非线性相关特征及交叉特征的问题。

2.1 非线性特征处理 — woe / 分桶

对于非线性特征线性化有 woe 处理和分桶两种方式。

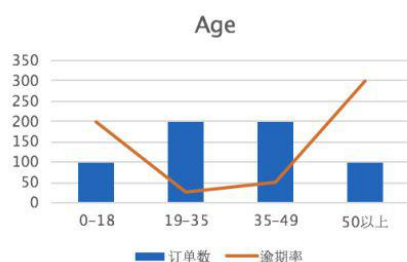
● 对非线性特征的处理

融360
Rong360.com

- 使用woe处理

$$woe_i = \ln \left(\frac{p_{y1}}{p_{y0}} \right) = \ln \left(\frac{\#B_i / \#B_T}{\#G_i / \#G_T} \right)$$

Age	bad	good	订单数	逾期率	WOE
0-18	40	60	100	40.00%	0.879733
19-35	10	190	200	5.00%	-1.65924
35-49	20	180	200	10.00%	-0.91203
50以上	60	40	100	60.00%	1.690663
all	130	470	600	21.67%	-1.2852



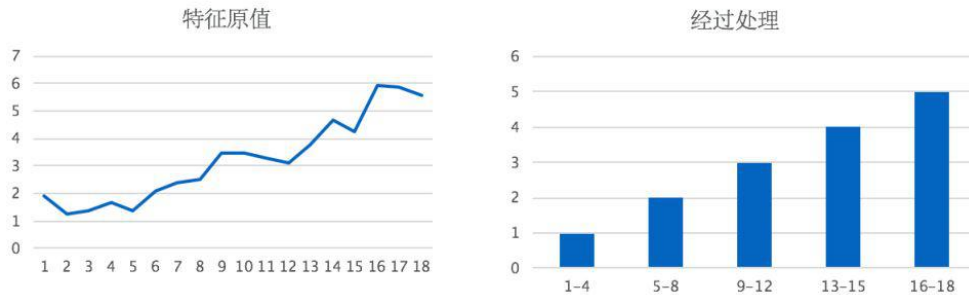
1) 左图中年龄与逾期率呈非线性相关性 ,对年龄特征做 woe 特征变换后 ,在右图中年龄与逾期率呈现了一定的线性相关性。再将变换后的年龄特征带入线性回归模型 ,就能比较好地体现和刻画年龄特征的价值。



● 对非线性特征的处理

融360
Rong360.com

- 使用分桶的方式处理
 - 根据分布分桶
 - 根据odds分桶
 - 根据IV值分桶



2) 对有一定跳变的连续值特征进行分桶，其优势在于弱线性特征转化为强线性特征，并且可以增加模型的鲁棒性。

2.2 交叉特征处理 — 分群

● 分群

融360
Rong360.com

根据各种客群的基本属性对客群进行细分，在每一个细分客群上单独建模

- 提升特征的区分能力，从而提升模型的区分能力，本质是一种交叉特征的体现
- 提升特征的稳定性，从而提升模型的稳定性

对于交叉特征的问题，可使用客户分群的方式。根据基本客群的基本属性划分，比如是否有征信，是否为VIP等基本属性划分，再对细分客



群进行单独建模，拟合风险。以上在一定程度上相当于做了交叉特征，该过程类似于决策树的建立，根据经验和业务指标建立决策树，将人群的划分作为不同的叶子节点。

最后，结合以上特征处理的技巧，在小样本量上进行线性回归就可以得到稳定且区分度高的模型。

3、总结



总结



优势

- 稳定
- 高可解释
- 可以实现冷启动
- 快速适应市场变化

劣势

- 需要强相关信息
- 特征提取/处理、分群非常依赖建模者的个人经验
- 能够挖掘的信息价值相对局限



评分卡模型的优势有稳定、高可解释性、可实现冷启动以及快速适应市场变化。其中冷启动优势的表现在于不论初始样本量的多少就能快速建模。其劣势为特征要求强相关，且非常依赖建模的个人经验，挖掘的信息价值相对有限。

二、机器学习的出现

随着机器学习的出现，传统金融逐渐向互联网金融转变。



1、传统金融和互联网金融的差异及新要求

● 业务场景的变化



传统金融

订单数少
订单金额高
借贷期限长
客群资质好
风控预算高

VS

互联网金融

订单数多
订单金额低
借贷期限短
客群资质差
风控预算低

传统金融的订单数少、订单金额高、贷款期限长、客群资质好、风控预算高。互联网金融订单数多、订单金额低、贷款期限短、客群资质差、风控预算低。风控的预算高低主要体现在订单金额的高低和期限的长短，其共同决定了贷款利率的高低。每一笔订单金额越高、期限越长，每一笔收获的利息越高，即风控预算越高。

针对互联网金融独有的特性，对风控提出了新的要求。

1) 目前具有央行征信的个体覆盖占全国的 50%左右，对于剩下 50%客群资质较差的个体，央行征信的风控系统无法进行覆盖。互联网金融的服务客群就是这剩下的 50%。所以要使得互联网金融信贷用户下沉，



就得要求风控模型覆盖的人群更广,准确地说覆盖大量传统风控不能评估的个体。

2) 互联网金融借贷的期限短,模型要求有较短预测有效期。

3) 互联网金融风控成本低,要求风控数据价格低。

2、金融风控数据

基于以上风控的新要求,对风控数据进行如下整体分析。

数据是根本,建模方法要根据数据进行不断地调整。将风控数据分为四部分,资质数据、信贷数据、消费数据、行为数据。其与逾期率的相关性从前往后逐渐减弱。

1) 资质类数据包括该个体是否有房车、教育程度、工资水平等个人属性相关的数据;

2) 信贷类数据包括历史的借贷额度、借贷次数、还款情况、是否逾期等借贷相关的数据;

3) 消费数据包括在电商平台的消费记录等流水相关的数据;

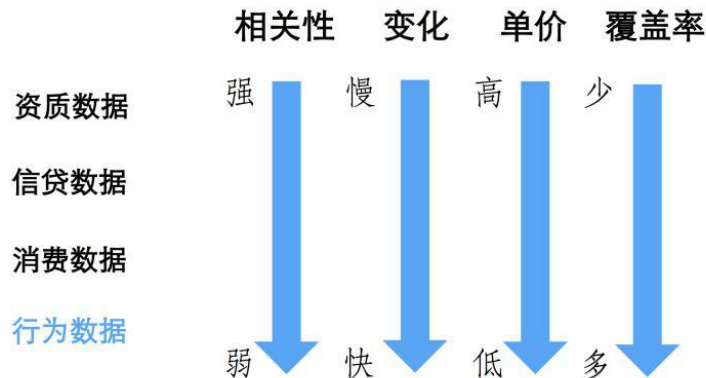
4) 行为数据包括个体之间的联系、手机 App 的使用列表、手机型号、浏览内容类别等个人行为数据。

互联网金融中资质较差的客群基本没有资质数据以及合规的信贷数据,虽然有一定的消费数据但是该类数据大多掌握在巨头电商平台手中,所以所能获取到的就是开放度高的互联网行为类数据。行为类数据的优势为覆盖广、成本低,缺陷为相关性弱、变化快。





风控数据的发展



3、机器学习的优势及问题

互联网金融的数据特征相关性弱，那么要求模型的挖掘能力要强，所以机器学习的重要性就显现出来了。

机器学习有以下优势：



机器学习的优势



机器学习技术引入了对特征拟合能力更强的模型，对非线性特征和交叉特征有较好的处理能力

决策树/带核函数SVM/神经网络-深度学习

同时，模型融合算法对进一步提升模型拟合能力和稳定性帮助较大。

Stacking/Bagging/Boosting

不同的基础模型结合不同的融合方式，衍生出了很多复杂的模型结构

rf/gbdt/xgboost/lightGBM

本质上来说是提升了模型的复杂度，模型对特定pattern的识别能力增强了，降低了对特征工程的要求，加快建模速度，让建模人员能处理更多的数据



1) 特征的拟合能力更强，对非线性和交叉特征有更好的能力；



2)能够基于不同的基础模型融合得到拟合更强和更稳的模型,比如 rf / gbdt / xgboost / lightGBM 。本质上,提高了模型的复杂度,模型对固定 pattern 的识别能力增强。总的来说,机器学习能从海量的、贫瘠的数据中挖掘出数据的价值。

机器学习在商业上应用比较成功的领域有广告、推荐等,但是与信贷领域有较大的差异:



机器学习的问题



机器学习之前成功的商业领域,比较典型的是广告、推荐等领域,这些领域与信贷领域存在较大差异

- 样本不同,广告推荐拥有亿级的近期样本,而信贷风控只有百万级的历史样本
- 预测时间不同,推荐广告只需要预测一个人下一秒会对什么内容感兴趣,而信贷风控需要预测一个人一个月后、一年后会不会还钱
- 模型作用周期不同,推荐广告模型训练非常频繁,一般至少是一天或者一小时重训一次,甚至使用 online learning,但是因为金融的滞后性,风控模型一般需要几个月甚至几年才会升级
- 泛化能力要求的不同,推荐广告一般只要求模型在特定的用户、特征的场景下起作用,但是征信机构的模型需要适用于各种客群和场景



1) 样本量级不同,广告推荐亿级近期样本,信贷风控只有百万级历史样本;

2) 预测时间不同,广告推荐实时预测,信贷风控是未来长周期的预测;

3) 模型更迭周期不同,广告推荐能短时间内进行模型更迭,能捕获到短时间内的样本特征回馈。信贷风控模型具有滞后性,样本回馈需要较长的周期,建模的训练数据与预测数据存在着较大的时间跨度;



4) 泛化能力不同，广告推荐面向的客群比较特定，信贷风控模型需要适用于广泛的客群和场景。

想要在信贷风控的场景成功地利用机器学习，必须要做一些迁移和适应。

三、机器学习技术的突围

在信贷风控领域关于机器学习技术的探索主要分为三个方向。

第一个方向，既然构造复杂模型存在着不稳定的风险，最稳妥的方式为使用机器学习 / 人工智能增加新特征，再使用评分卡模型。

第二个方向为传统风控为体，机器学习为用。即特征筛选的标准和规则仍不变，仅替换评分卡模型为复杂模型比如 xgboost 等。

第三个方向为大规模样本结合机器学习，保证模型的稳定和泛化。



三个方向

融360
Rong360.com

- 使用机器学习/人工智能的方式加工新的特征
- 传统风控为体，机器学习为用
- 使用大规模样本和模型保住稳定和泛化



1、使用机器学习 / 人工智能加工特征、风控为体-机器学习为用

关于机器学习 / 人工智能加工特征，做了以下的实践尝试。



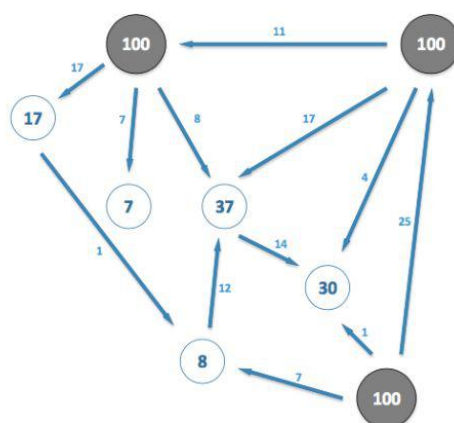
1.1 社交关系图 / 知识图谱

将用户看为一个点，用户之间的联系为边。基于图结构构造特征，比如图的出 / 入度，度中黑名单的个数，黑用户 randomwalk 将度权重传导至相邻的边等衍生特征。



社交关系图

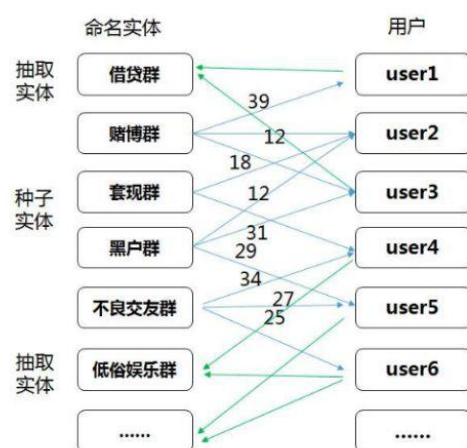
融360
rong360.com



1.2 二部图游走

以下为通过二部图来衍生特征，首先建立用户与实体的二部图，实体可以为 QQ 群，App，浏览链接等等。用户与实体特性相关则连接一条权重边，将有反面特性的实体作为种子实体，向用户侧游走，用户侧又向实体侧游走，直到游走平衡进入稳态可以得到稳定的权重。将稳定的权重作为特征，来标识反面特性的程度。





1.3 word2vec

word2vec 主要应用于 NLP 领域，在风控领域中的运用鲜少。目前该方法应用于页面埋点动作的特征提取，将点击和浏览动作看作词，用户完整周期内的所有动作行为组成了段落和文章，再使用 word2vec 方法，将用户的点击浏览行为隐含表征为一定维度的向量，再将向量作为用户的特征输入 xgboost 进行建模，单个特征有 1-2 个 ks 点的提升，效果还是不错的。

1.4 订单序列模型

用时序的模型如 LSTM 处理用户某一个订单的相关特征，如期限、金额、时间、到期账单等，其中一个订单表示一个时序事件，用户生命周期所有订单构成完整的时序链。将历史用户周期内的订单特征灌入 LSTM 得到 $(1, N)$ 维特征，再将这些特征输入 LR 模型进行训练。



该模型对于跨时间不是太稳定，但是相较之前还是有较大的提升，后续还需要持续优化。

2、模型和数据的减压、模型监控

以传统的风控为体，机器学习为用的方式，虽然使用了机器学习方法但还是向传统靠拢。该方式强化了人工干预和信息抽象，弱化模型的复杂度和数据，以此来提升模型特征的信息含量和稳定。比如以高要求（高 IV 低 PSI）筛选入模特征，或者 LDA 文本主题词浓缩大量文本信息等方式来给模型减压。



给模型和数据减压



不能把太多的工作交给模型和数据，在特征处理上使用无监督的方式做人工干预和信息抽象，提升进模型特征的信息含量和稳定性

- 使用APP类别替代独立的APP
- 使用文章主题替代单个的关键词
- 高要求筛选入模特征，高IV，低PSI



为了保证模型的稳定性，可加强对模型的监控即加强数据和特征的分布及中间结果的监控。同时可加快模型的更新效率，建设模型自动重训练机制。





加强模型监控



- 因为金融建模只能使用历史样本，必须使用回溯的方式提取特征，回溯数据和线上实时数据的分布很难保证绝对一致；互金行业在高速发展，客群和数据源都在快速变化，也加快了数据分布的变化速度
- 监控线上实时预测过程中所有特征和模型中间结果、最终结果，和建模样本做对比，才能了解模型在其作用周期中的效果
- 加快模型更新效率，建设模型自动重训机制



以传统风控为体，机器学习为用的方式能够使得模型构建过程更加透明，特征构建更加灵活。也就是机器学习技术仅仅用于新特征的挖掘，最终仍以高要求筛选特征，入模评分卡。



向传统低头



- 机器学习技术仅仅用于新特征的挖掘，最终的训练模型仍然选用稳定的评分卡模型。
- GBDT的叶子节点、DNN的中间结果作为特征
- 所有统一使用评分卡模型的特征选择方式进行筛选，保证特征的稳定性，不符合筛选条件的特征暂时放弃



3、大规模机器学习



在去年 9 月份开始尝试大规模机器学习模型 — XGB-LR / deepFM / Deep&wide ，大规模机器学习有以下要求：

1) 百万级别的样本。对样本的一致性要求不是很高，可将时间跨度大的订单样本作为输入。

2) 数万维的特征，数十万参数的模型。大容量模型意味着多参数，多特征。其中输入特征为所有特征，即不经过高要求（高 IV 低 PSI）筛选入模。其主要风险点为线下线上模型效果差异大，即线下建模效果佳，完全拟合了线下分布，但是上线后可能由于过拟合等原因导致效果不佳。所以到目前为止，该模型一直处于线下陪跑测试阶段，监控其稳定性。

传统风控为体-机器学习为用方式建模的模型效果，对比大规模机器学习的模型效果，大部分时间内，其效果要优于传统风控为体-机器学习为用方式，并且有一定的稳定性。

4、总结





总结



- 机器学习技术在信贷风控领域的应用在还处于相对初期的阶段，远不如其在广告推荐、语音图像、自然语言等领域的成熟度
- 在引入新技术的同时需要参考应用场景，在传统的建模原则上循序渐进的优化，同时实时监控效果
- 整体来看机器学习在信贷风控领域的发展还有很长的路要走



在个人技术成长的角度有以下总结：

- 1) 机器学习技术在信贷风控领域远不如广告推荐、语音图像、自然语言等领域成熟，成长期大概在 5-6 年，还有非常多的机器学习的点值得探索。
- 2) 在引入新技术同时需要参考应用场景，比如新技术在广告领域 auc 达到了 0.99 等，但是在风控领域其实不然。所以需要充分考虑风控领域的特性，将机器学习的优势迁移过来。
- 3) 机器学习技术在信贷风控领域还有广阔的发展空间和前景。



度小满智能获客系统实践

度小满智能获客系统实践

度小满金融 信贷产品营销部 费浩峻



分享嘉宾：费浩峻 度小满金融 主任架构师

编辑整理：李婷婷

内容来源：百融云创&DataFun AI Talk

出品社区：DataFun

注：欢迎转载，转载请注明出处

大家好，今天分享的主题是：度小满金融信贷业务的智能获客系统实践。

根据过去两年的工作经验，将其总结沉淀为渠道、效率、数据理解和度小满智能获客系统这四个环节，其中前三个环节是对获客的理解，渠道指明了获客方向，效率好比获客的发动机，数据则是获客的燃料。

渠道

1. 都有哪些渠道？



01 渠道-从运筹帷幄到决战千里



获客时首先会思考这样的问题：获客的体量有多大？获客的人群是什么样的？获客的渠道有哪些？获客的成本是多少等等。实际工作中，我们应先充分开发公司内部数据，逐步接入外部数据。常见的获客渠道有：直接触达，竞价广告、流量合作和线下渠道。

- 直接触达，依托公司内部数据，具有高转化率，低接入成本，高可操作性，但受限于客户规模；
- 竞价广告，公司外部主要获客渠道，克服了客户规模弊端，但是市场竞争激烈易产生恶劣竞争、市场变动快、影响大，成本高；
- 流量合作，是一种较为稳妥的方式，客户规模大，增强公司壁垒，但接入成本高，可操作性弱；
- 线下渠道，复杂多样，各个公司都在尝试，但还未形成成熟的获客环境。



2. 如何选择渠道？

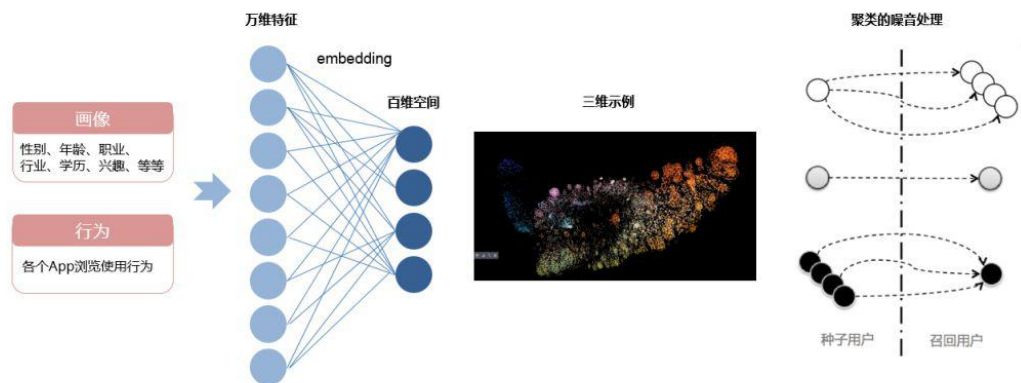
01 渠道-从感官到数据驱动的选择



不同渠道各有利弊，如何选择渠道？从认知角度来看，产品初期，以感官认知为主，依赖产品的定位，从而确定渠道；产品发展期，分析认知逐渐重要，用户画像、用户标签帮助我们更好确定目标客户，选择合作渠道；产品成熟期，分析认知很难提高效果，转为大数据驱动的数据模型认知，模型可以更好地细分客户，具有高信息利用率、判别准确率和鲁棒性。



01 基于人群DNA的lookalike



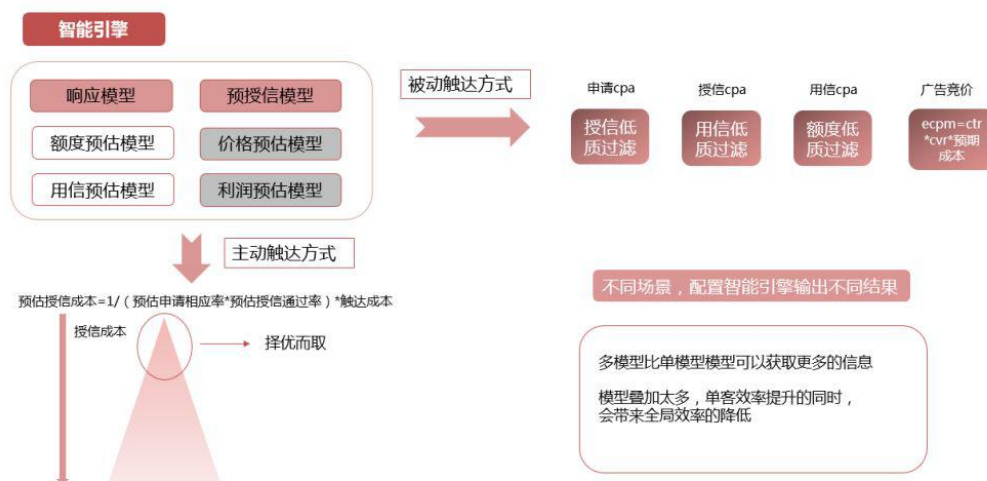
主要做法是把已知的行为和用户画像，抽象成特征，这时我们的数据集有了万维特征，然后在把特征 Embedding 成百维特征空间里，当需要投放时，会寻找目标客群，以及相似的空间向量（相似客群）进行投放。

PS：在寻找相似人群的时候，需要考虑噪音的问题，会把单目标客群到多目标客群去掉，voting 是必要的。

效率



02 效率-流量价值的深度量化



1. 单客户的效率，如何更精准的获客

智能引擎可以提高单客户的效率，在风控模型调用前使用，它由响应模型、预授信模型、额度预估模型、价格预估模型、用信预估模型和利润预估模型组成。

- 以主动触达方式为例，对授信成本进行预估，将响应模型和预授信模型拟合在一起，从而得到客户授信成本的排序，在选择客户时，优先选择预授信成本低的客户进行广告投放；
- 以被动触达方式为例，在授信、用信、额度、广告竞价环节中，均使用模型过滤掉质量低用户，这种方式会尽量最大化单客户效率。

可能大家会疑惑为什么需要建立多环节的模型，原因有二：

一是，不同阶段用户流失是不同的，建立多个环节的模型可以有选择的使用，确保不是一味的提升单客效率而降低了全局的效率；

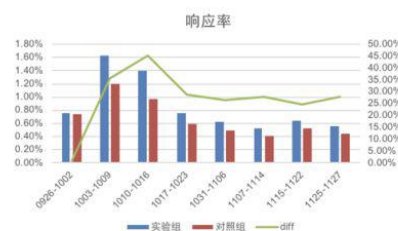
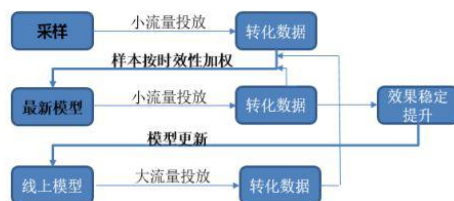


二是，相同的特征在不同的模型中表现差异性很大，可以更有效使用特征。

2. 整体的效率，模型如何快速迭代

02 效率-实验平台快速找到正确的方向

Test & learn



严格的AB test对比机制

避免产品调整以及市场波动带来的影响，数据说话排除主观臆测

实验和模型的迭代是继续往前走的很重要的保障，以实验流程缩减流程图为例，线上同一时间运行四个模型，① 线上最优模型，② 根据新样本每天更新的模型，③ 一系列的观察模型，④ 随机采样的过程；采样过程可以做很多事情，包括 MAB 的方法或自己设定方向的探索，比如下一版模型中，新样本应占总体样本的比例，样本的时效性处理等等。模型对比应做到严格的 AB test 对比机制，避免产品调整，市场波动带来的影响，排除主观臆测。

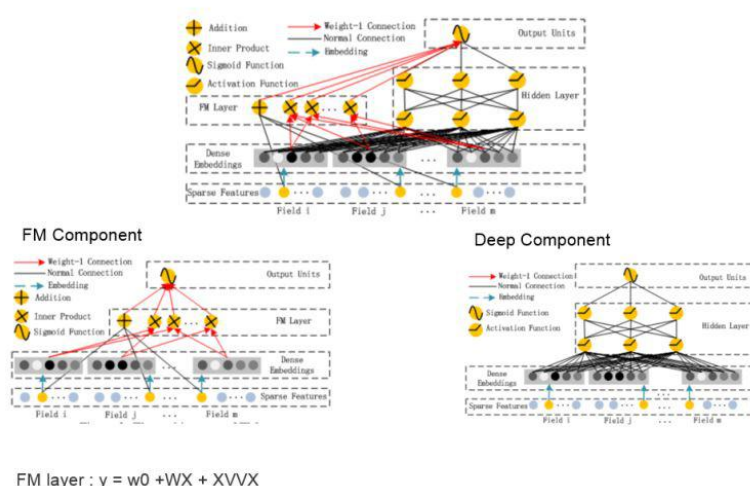
数据理解



第三部分是讲数据，我们前面提到，数据是获客的燃料，模型做的好不好，数据理解是最重要的，数据理解主要包括两个方向：特征理解（如何更好的挖掘理解特征）和样本理解（如何理解样本是否为目标样本）。

1. 特征理解，deepFM 更好的理解特征

03 数据理解-deepFM加强对特征的理解



将不同的特征进行融合，解释这样的特征时，我们一般采用两种方式，一种是从广度上理解，如矩阵式的；另一种是从深度上的理解，如深度神经网络，梳理各个特征之间存在的隐藏关系。DeepFM 是我们去年开始调研部署上线的模型，整体模型结构是非常简洁的，DeepFM 就分两个模块，一个是 FM 模块，一个是 DNN 模块。基本上线上线下在同样的数据特征、评测集要比 XGBoost 的 AUC 提升 1~2 个点。

2. 样本理解，建立全生命周期数据构造



03 数据理解-全生命周期数据构造加强对样本理解



构造全链路的转换数据是做好转换归因的第一步

尽量避免使用受影响太大的样本

找到关键的动作

营销行为复杂，最后一次行为无法归因到转换动机

构造全链路的转换数据是做好转换归因的第一步，以客户触达的为例，客户前后共经过 feed 流广告展示、短信触达、抖音广告展示、下载 app，搜索广告展示、电话外呼、爱奇艺会员入口展示，用户申请。很难确定哪一种获客方式获得该客户，在做分析时，尽量避免使用受影响太大的样本。总而言之，我们需要找到用户的关键动作并对之进行分析，但营销场景可能非常复杂，我们不能将用户的关键动作归因到具体的获客渠道策略时，全生命周期的数据管理可以帮助我们解决这样的问题。

度小满智能获客平台



04 度小满智能获客平台



下面介绍度小满获客平台现在的架构，主要分为数据层、技术层、模型层、系统层、功能层和渠道。

① 从数据来讲，主要的数据来源是用户数据，第二个是广告数据、第三个是用户的生态数据；

② 在系统层，主要为智能引擎、创意引擎、广告投放平台、内容辅写平台、实验平台这五个部分：

- 智能引擎主要配置响应模型、授信预估等，使用常规的机器学习模型一般为 xgboost，deepFM，stacking 等；
- 创意引擎，指的是获客的千人千面的创意，实现方式有文本创意和图片创意，文本创意以 GAN 深度学习模型为主，图片创意现在做到提取图片关键因素，还未实现完全图片的自动生成；
- 广告投放平台是主要体现技术的部分，投放时间预估、投放频度等；



- 内容辅写平台以 nlp 技术为主，同时也包括文章质量的预测、热门文章预测等；

- 实验平台主要是加速实验的进程和舆情的监控。

③ 上面介绍了智能获客平台的底层架构，主要是为了实现上层功能，也就是多目标客群的识别、千人千面创意和快速迭代反馈，从而为客户匹配合适的渠道，更适合的触达方式和触达内容。



金融风控反欺诈之图算法

分享嘉宾: 张行军 Abakus 高级风控算法经理

内容来源: DataFun AI+ Talk

出品社区: DataFun

注：欢迎转载，转载请注明出处。

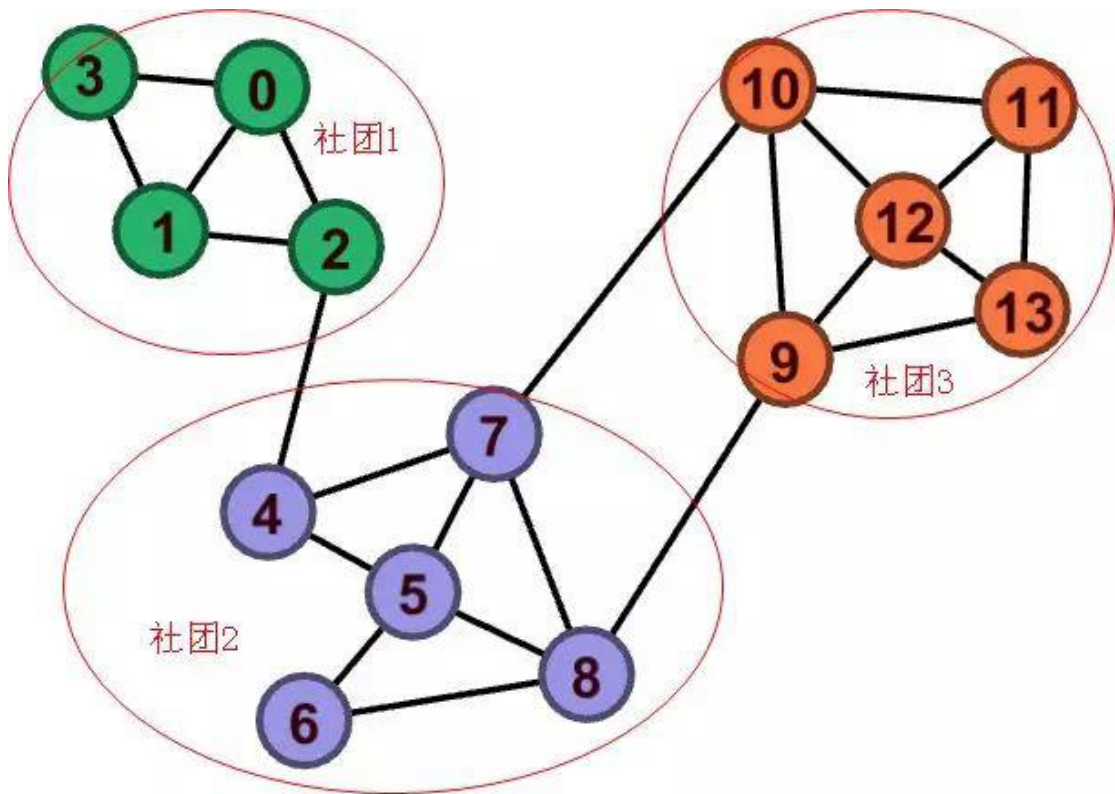
先介绍下金融借贷业务流程：用户前来申请借贷，会先经过欺诈识别，把欺诈团伙和主观欺诈的个人拒绝掉，然后对通过的人做信用评估，最后根据额度模型，算出利润最大化时放款金额。

刚才提到了团队欺诈，举个真实的例子。宜人贷在他们的财报中公布的，他们被一个团伙成功撸走了 **2000** 多单，当时宜人贷的件均 **4w**，一下损失了 **8000w**!! 那么如何防范这种风险呢。这就是今天要分享的图算法。图可以将这些一个个有良好记录的个体关联起来，一网打尽。

再举一些团伙欺诈的行为。比如一个团伙，注册真实的淘宝商家，然后刷出良好的淘宝购物记录。或者来回转账，刷出良好的银行流水。

刚才前两位老师都没有提到额度模型，简单介绍下，如果只给用户放款 **5000**，可能坏账风险很小，但是利息也少，如果放款 **10000**，利息虽然收到利息多了，但是坏账风险高岭，所以需要做个权衡





Graph 简介

$G=(V,E)$ $G=(V,E)$

- **V: vertex set**
- **E: edge set** (有向, 无向, 有权重和没有权重)

举例, 两个人之间的联系, **A** 给 **B** 买了东西, **A** 和 **B** 之间的通话次数时长多于 **A** 和 **C** 之间。

- 度中心性 (**Degree Centrality**) - 表示连接到某节点的边数。在有向图中, 我们可以有 **2** 个度中心性度量: 流入和流出。一个节点的节点度越大就意味着该节点在网络中就越重要。
- 接近中心性 (**Closeness Centrality**) - 从某节点到所有其他节点的最短路径的平均长度。反映在网络中某一节点与其他节点之间的接近程度。
- 介中心性 (**Betweenness Centrality**) - 某节点在多少对节点的最短路径上。介数中心性是比较能体现节点在图中桥梁作用的中心性度量方法。介数反映了相应的节点或者边在整个网络中的作用和影响力, 具有很强的现实意义。例如, 在交通网络中, 介数较高的道路拥挤的概率很大; 在电力网络中, 介数较高的输电线路和节点容易发生危险。

社团发现算法一般有:



- 最小割，正则化割:通过计算图的最小割，即将网络划分为预定的分组数，并使连接各分组的边的条数最少。
- 非负矩阵分解:基本原理是将原始矩阵分解得到社区指示矩阵和基矩阵
- 基于模块度的社区划分
- 基于节点相似性的社区划分

最小割算法广泛应用在分布式计算的负载均衡中，对集群节点的分组有利于减少不相关节点之间的通信。然而由于该算法限定了网络最终分组的个数，而不能通过算法“发现”节点间的内在联系并自然地构成若干个社区，因此最小割算法应用较为局限。

本文主要分享这两类的主要算法，基于模块度的 **louvain** 和基于信息熵 **Infomap**，基于相似度的 **node2vec**

模块度(Modularity)公式及简化

优化目标：一般认为社团内部的点之间的连接相对稠密，而不同社团的点之间的连接相对稀疏。

所以模块度也可以理解是社区内部边的权重减去所有与社区节点相连的边的权重和，对无向图更好理解，即社区内部边的度数（内部的连线数）减去社区内节点的总度数。



$$\begin{aligned}
Q &= \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \\
&= \frac{1}{2m} \left[\sum_{i,j} A_{ij} - \frac{\sum_i k_i \sum_j k_j}{2m} \right] \delta(c_i, c_j) \\
&= \frac{1}{2m} \sum_c \left[\sum_{in} - \frac{(\sum_{tot})^2}{2m} \right] \\
&= \sum_c \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 \right] \\
&= \sum_a [e_c - a_c^2]
\end{aligned}$$

$$\delta(u, v) = \begin{cases} 1 & \text{when } u == v \\ 0 & \text{else} \end{cases}$$

模块度公式的解释

A_{ij} 节点 i 和节点 j 之间边的权重，网络不是带权图时，所有边的权重可以看做是 1；

$k_i = \sum A_{ij}$ 表示所有与节点 i 相连的边的权重之和（度数）；

c_i 表示节点 i 所属的社区；

$m = \frac{1}{2} \sum_{i,j} A_{ij}$ 表示所有边的权重之和（边的数目）。



其中 $\sum_{in}$ 表示社区 c 内的边的权重之和， $\sum_{tot}$ 表示与社区 c 内的节点相连的边的权重之和，即社区 c 节点的度之和（包含与其他社区相连边的度）。
从概率的角度去看：

e_c 表示实际情况下， c 社区内产生边的概率。

a_c^2 表示在一种理想情况下，给定任意节点 i 的度 k_i ，对节点 i 和节点 j 进行随机连边，边属于社区 c 的概率期望。

于是上式就表示了社区内连边数与随机期望的一个差值。连边数比随机期望值越高，表明社区划分的越好。

一般使用后面简化的公式，简化后的公式删除了判断两个节点是否划为同一个社区的函数，所以在一定程度上大大减少了 Q 值计算量。

louvain

louvain 算法的思想很简单：

- 将图中的每个节点看成一个独立的社区，此时社区的数目与节点个数相同；
- 对每个节点 i ，依次尝试把节点 i 分配到你每个邻居节点所在的社区，计算分

配前与分配后的模块度变化 ΔQ ，并记录 ΔQ 最大的那个

邻居节点，如果 $\max \Delta Q > 0$ ，则把节点 i 分配

ΔQ 最大的那个邻居节点所在的社区，否则保持不变；

- 重复 2，直到所有节点的所属社区不再变化；
- 对图进行压缩，将所有在同一个社区的节点压缩成一个新节点，社区内节点之间的边的权重转化为新节点的环的权重，社区间的边权重转化为新节点间的边权重，然后重复 2，3；
- 重复 2~4，直到整个图的模块度不再发生变化。



第一阶段称为 **Modularity Optimization**，主要是将每个节点划分到与其邻接的节点所在的社区中，以使得模块度的值不断变大；

第二阶段称为 **Community Aggregation**，主要是将第一步划分出来的社区聚合成为一个点，即根据上一步生成的社区结构重新构造网络。重复以上的过程，直到网络中的结构不再改变为止。

移动

$$Q_1 = \left[\frac{E_c}{m} - \left(\frac{\sum_{tot}}{2m} \right)^2 \right] + \left[\frac{o}{m} - \left(\frac{k_i}{2m} \right)^2 \right]$$

$$Q_2 = \frac{E_c + E_{i,c}}{m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2$$

$$\begin{aligned} \Delta Q &= Q_2 - Q_1 \\ &= \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right] \\ &= \left[\frac{k_{i,in}}{2m} - \frac{\sum_{tot} k_i}{2m^2} \right] \end{aligned}$$

$k_{i,in}$ 是社区 c 内节点与节点 i 的边权重之和，再乘以 2

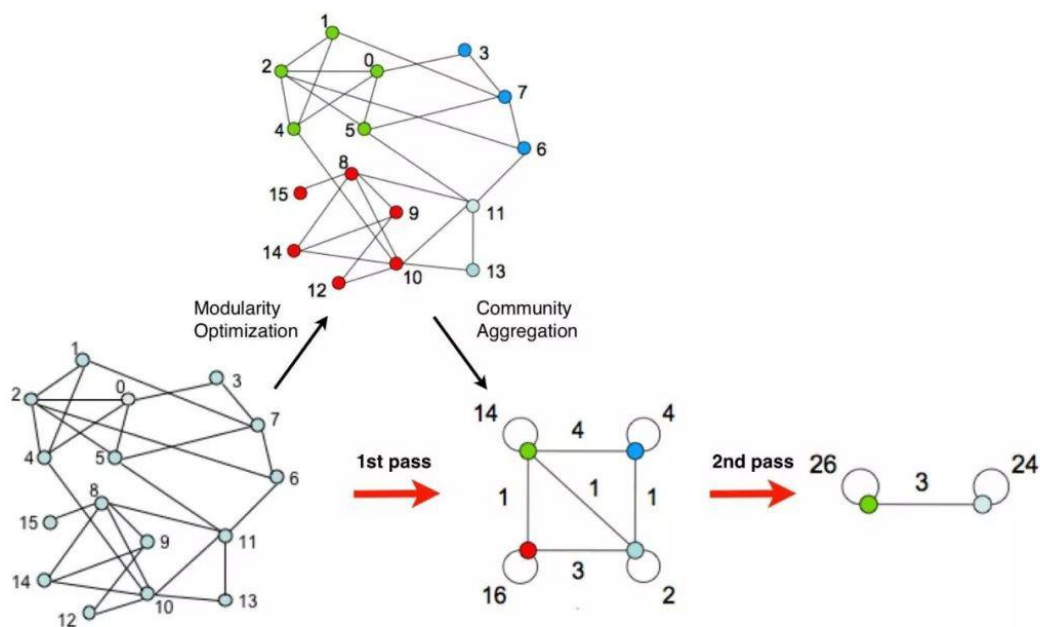
前面部分表示把节点 i 加入到社区 c 后的模块度，后一部分是节点 i 作为一个独立社区和社区 c 的模块度

1. 模块度与 **louvain** 社区发现算法

<http://www.cnblogs.com/fengfenggirl/p/louvain.html>

2. **Spark GraphX** 分布式图计算实战





Infomap

从信息论的角度出发，假设一个 **random worker** 在图上进行随机游走，那么怎么用最少的编码长度来表示其路径呢？

如果节点存在社区结构，那么社区内的节点就可以共享社区的 **bit** 位码，可以得到更小的平均比特，所以社区划分的越好，那么表示任意一条随机游走的路径所需的平均比特就越小。

如果我们能够计算出每个节点的到达概率，就可以依据信息熵的公式来量化平均比特了：

$$L(P) = H(P) \equiv - \sum_{\alpha} p_{\alpha} \log p_{\alpha}$$

怎么计算每个点的到达概率呢？

一个暴力的办法是在图上进行长时间的随机游走，最后统计每个节点的出现概率。太暴力了。

利用 **pagerank** 思路，初始化了每个节点的到达概率之后，就可以不断地迭代更新每个节点的到达概率，这个结果会很快趋于收敛。

其实这过程就是一个马尔科夫随机过程，随机初始化起始值，然后随机游走就相当于不停地用概率转移矩阵相乘，最后就可以达到马尔科夫稳态。

把随机游走事件归为三类：进入某个社团，离开某个社团，再社团内部游走。

定义清楚各类事件的发生概率，依据信息熵公式，就可以得到此时编码所需的平均比特了，其本质就是从信息论的角度出发。



$$L(M) = q_{\sim} H(Q) + \sum_{i=1}^m p_{\cup}^i H(P^i)$$

$L(M)$ 将图划分成 m 个社区，编码随机游走路径时的平均比特；

$q_{\sim} = \sum_{i=1}^m q_{i\sim}$ 产生进入社区这种事件的总概率；

$H(Q) = -\sum_{i=1}^m (q_{i\sim}/q_{\sim}) \log(q_{i\sim}/q_{\sim})$ 进入社区事件发生的信息熵；

$p_{i\cup} = \sum_{\alpha \in i} p_{\alpha} + q_{i\sim}$ random worker 在社区 i 内部的概率，这里包括了在社区内跳转和离开该社区两类事件；

$H(P^i) = -(q_{i\sim}/p_{i\cup}) \log(q_{i\sim}/p_{i\cup}) - \sum_{\alpha \in i} (p_{\alpha}/p_{i\cup}) \log(p_{\alpha}/p_{i\cup})$ random worker 下一步事件发生在社区 i 内部的信息熵；

Infomap 算法的迭代过程

1. 初始化，对每个节点都视作独立的社区；
2. **while** 平均比特的值不再下降；
3. 对图里的节点随机采样出一个序列，按顺序依次尝试将每个节点赋给邻居节点所在的社区，取平均比特下降最大时的社区赋给该节点，如果没有下降，该节点的社区不变。

参考链接

1. The map equation

<http://www.mapequation.org/apps/MapDemo.html>

2. <https://mp.weixin.qq.com/s/qUxMesOA-edSyHeudORRGa>

3. DEEP GRAPH INFOMAX 阅读笔记

<https://zhuanlan.zhihu.com/p/58682802>

Graph embeddings

Deepwalk

1. 使用随机游走(RandomWalk)的方式在图中进行节点采样获得节点共关系，
2. 使用 **skip-gram**，根据步骤 1 中生成的节点序列学习每个节点的向量表示。
skip-gram 就是根据给定输入的节点，预测上下文节点。

Deepwalk 有多不足，比如泛化能力，有新节点加入时，它必须重新训练模型以表示该节点。



其中一个就是采样，从其邻居中随机采样节点作为下一个访问节点，是一种可重复访问已访问节点的深度优先遍历算法。

node2vec 是一种综合考虑 **DFS** 邻域和 **BFS** 邻域的 **graph embedding** 方法

node2vec

优化目标：

$$\max_f \sum_{u \in V} \log Pr(N_S(u) | f(u))$$

条件独立假设：

$$Pr(N_S(u) | f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i | f(u))$$

特征空间的对称性：

$$Pr(n_i | f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}$$

优化目标：

$$\max_f \sum_{u \in V} [-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u)]$$

$$Z_u = \sum_{v \in V} \exp(f(u) \cdot f(v))$$
 计算

量非常大，所以论文采用负采样(**negative sample**)进行近似计算。

这个 **node2vec** 优化目标函数，因为它跟大名鼎鼎的 **word2vec** 是一样。



我们最初是用一个 **Python** 写的包，跑一遍算法需要一周。后来想，既然优化目标是一样的，那能不能用 **word2vec** 包，因为 **word2vec** 用 **c** 写的，而且还采用了 **Hierarchical Softmax**, **negative sampling** 加速。

然后在网上找到了一个套用 **word2vec** 实现的 **node2vec** 包，速度快很多。

随机游走的方式

复杂网络处理的任务其实离不开两种特性，前面也提到过：一种是同质性，就是之前所说的社区。一种就是结构相似性，值得注意的是，结构相似的两个点未必相连，可以是相距很远的两个节点。

能不能改进 **DeepWalk** 中随机游走的方式，使它综合 **DFS** 和 **BFS** 的特性呢？所以本文引入了两个参数用来控制随机游走产生的方式。

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\alpha_{pq}(t, x) \cdot w_{vx}}{Z} & \text{if } (v, x) \in E \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

Z 是分子的归一化常数

如果已经采样了 **(t, v)**，也就是说现在停留在节点 **v** 上，那么下一个要采样的节点 **x** 是哪个？作者定义了一个概率分布，也就是一个节点到它的不同邻居的转移概率：

直观的解释一下这个分布：

- 如果 **t** 与 **x** 相等，那么采样 **x** 的概率为 $\frac{1}{p}$ ；
- 如果 **t** 与 **x** 相连，那么采样 **x** 的概率为 1；
- 如果 **t** 与 **x** 不相连，那么采样 **x** 的概率为 $\frac{1}{q}$ 。

参数 **p**、**q** 的意义分别如下：

返回概率 **p**：



- 如果 $p > \max(q, 1)$ ，那么采样会尽量不往回走，对应上图的情况，就是下一个节点不太可能是上一个访问的节点 t 。
- 如果 $p < \min(q, 1)$ ，那么采样会更倾向于返回上一个节点，这样就会一直在起始点周围某些节点来回转来转去。

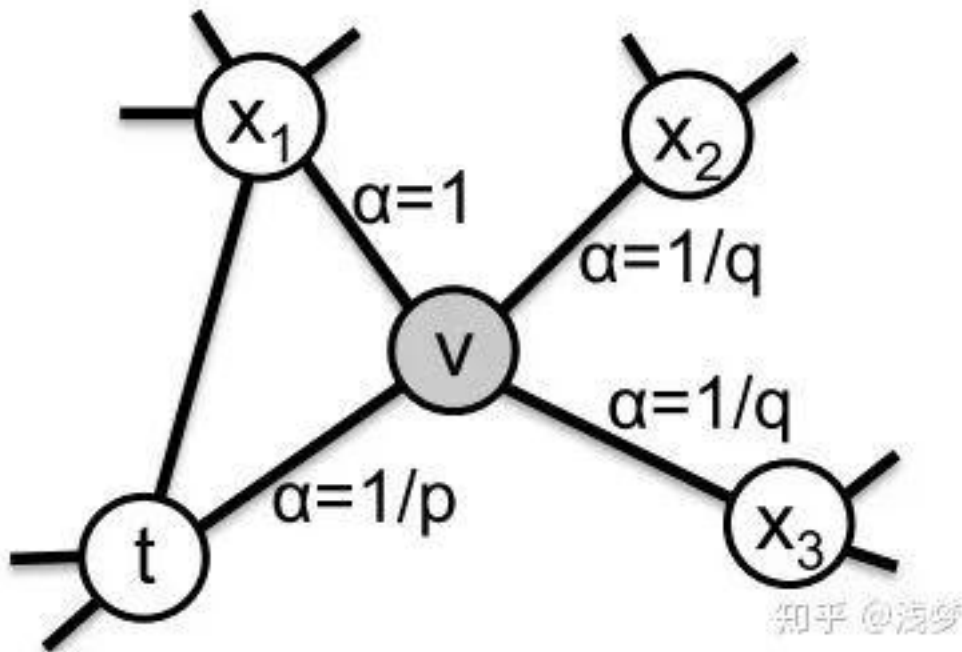
出入参数 q :

- 如果 $q > 1$ ，那么游走会倾向于在起始点周围的节点之间跑，可以反映出一个节点的 **BFS** 特性。
- 如果 $q < 1$ ，那么游走会倾向于往远处跑，反映出 **DFS** 特性。
- 当 $p=1$, $q=1$ 时，游走方式就等同于 **DeepWalk** 中的随机游走。

简而言之：

参数 p 控制重复访问刚刚访问过的顶点的概率，

参数 q 控制着游走是向外还是向内，若 $q > 1$ ，随机游走倾向于访问和 t 接近的顶点(偏向 **BFS**)。若 $q < 1$ ，倾向于访问远离 t 的顶点(偏向 **DFS**)。



缺点

1. 慢
2. 先 **embedding** 再聚类，感觉这两个过程很割裂！！融合一下

come

Graphembedding 得到向量后，可以做很多事情，在我们这个主题可以简单的通过聚类来讲节点分组。



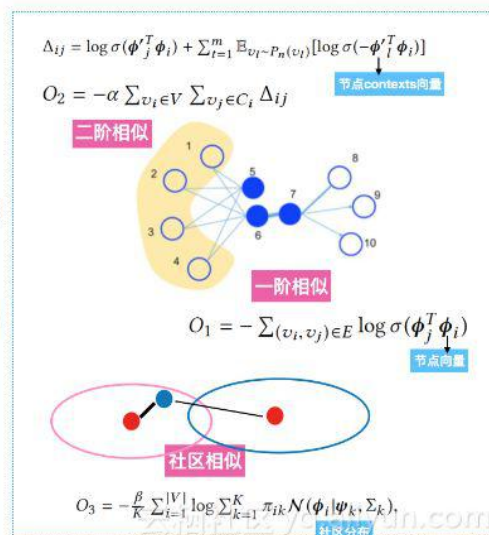
但是这个过程比较割裂，先优化 **node2vec**，然后再优化聚类。能不能整体上一次性优化完呢。

comE 这个算法优化目标中加入了社区的检测和嵌入。通过一个混合高斯模型将节点划分开。

$$\begin{array}{ccc} \text{一阶LOSS} & \text{二阶LOSS} & \text{社区LOSS} \\ \downarrow & \downarrow & \downarrow \\ \mathcal{L}(\Phi, \Phi', \Pi, \Psi, \Sigma) = O_1(\Phi) + O_2(\Phi, \Phi') + O_3(\Phi, \Pi, \Psi, \Sigma) \end{array}$$

Our final optimization problem becomes:

$$(\Phi^*, \Phi'^*, \Pi^*, \Psi^*, \Sigma^*) \leftarrow \arg \min_{\forall k, \text{diag}(\Sigma_k) > 0} \mathcal{L}(\Phi, \Phi', \Pi, \Psi, \Sigma)$$



<1>.社区检测和社区嵌入。

这篇论文比较创新的一点，就是将社区嵌入定义为低维空间中的多元高斯分布。即对于社区k来说($k \in \{1, \dots, K\}$)，它在d维空间下的社区嵌入是一个多元高斯分布 $\mathcal{N}(\psi_k, \Sigma_k)$ ，其中 $\psi_k \in R^d$ 表示的是社区内节点的平均向量（即中心点）， $\Sigma_k \in R^{d \times d}$ 表示的是协方差矩阵。

整篇文章最大的难点在于，如何用统一的目标函数来使得社区嵌入、社区检测和节点嵌入三者问题得到最优化的结果。由于社区嵌入是由多元高斯分布来表达的，因此就可以通过高斯混合模型来完成。

假设每一个节点的嵌入向量 ϕ_i 可以由该节点所在社区的多元高斯分布产生，这样一来，就可以得到一个似然目标函数：

$$\prod_{i=1}^{|V|} \sum_{k=1}^K p(z_i = k) p(v_i | z_i = k; \phi_i, \psi_k, \Sigma_k) \quad (1)$$

其中， $p(z_i = k)$ 表示的是节点 v_i 属于社区k的概率。方便表示，可以将这个概率表达为 π_{ik} ：

$\pi_{ik} \in [0, 1], \sum_{k=1}^K \pi_{ik} = 1$ 。在社区发现问题中，这个概率是未知的。

而 $p(v_i | z_i = k; \phi_i, \psi_k, \Sigma_k) = \mathcal{N}(\phi_i | \psi_k, \Sigma_k)$ 表示的是多元高斯分布。在社区嵌入问题中， ψ_k, Σ_k 是未知的。

综上，将公式1对上述三个未知变量求导，就可以同时实现社区检测和嵌入。

优化目标中前面两项跟 LINE 定义的相似度相似：



1st

对于每一条无向边 (i, j) ，定义顶点 v_i 和 v_j 之间的联合概率为：

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-\vec{u}_i^T \cdot \vec{u}_j)}, \quad \vec{u}_i \text{ 为顶点 } v_i \text{ 的低维向量表示。 (可以看作一个内积模型, 计算两个 item 之间的匹配程度)}$$

$$\text{同时定义经验分布 } \hat{p}_1 = \frac{w_{ij}}{W}, W = \sum_{(i,j) \in E} w_{ij}$$

$$\text{优化目标为最小化: } O_1 = d(\hat{p}_1(\cdot, \cdot), p_1(\cdot, \cdot))$$

$d(\cdot, \cdot)$ 是两个分布的距离，常用的衡量两个概率分布差异的指标为KL散度，使用KL散度并忽略常数项后有

$$O_1 = - \sum_{(i,j) \in E} w_{ij} \log p_1(v_i, v_j)$$

1st order 相似度只能用于无向图当中。

2nd

这里对于每个顶点维护两个embedding向量，一个是该顶点本身的表示向量，一个是该点作为其他顶点的上下文顶点时的表示向量。

对于有向边 (i, j) ，定义给定顶点 v_i 条件下，产生上下文(邻居)顶点 v_j 的概率为

$$p_2(v_j | v_i) = \frac{\exp(\vec{c}_j^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} \exp(\vec{c}_k^T \cdot \vec{u}_i)}, \quad \text{其中 } |V| \text{ 为上下文顶点的个数。}$$

优化目标为

$$O_2 = \sum_{i \in V} \lambda_i d(\hat{p}_2(\cdot | v_i), p_2(\cdot | v_i)), \quad \text{其中 } \lambda_i \text{ 为控制节点重要性的因子, 可以通过顶点的度数或者PageRank等方法估计得到。}$$

经验分布定义为: $\hat{p}_2(v_j | v_i) = \frac{w_{ij}}{d_i}$, w_{ij} 是边 (i, j) 的边权, d_i 是顶点 v_i 的出度, 对于带权图, $d_i = \sum_{k \in N(i)} W_{ik}$

使用KL散度并设 $\lambda_i = d_i$, 忽略常数项, 有

$$O_2 = - \sum_{(i,j) \in E} w_{ij} \log p_2(v_j | v_i)$$

1. <https://blog.csdn.net/u012151283/article/details/87013915>

2. learning Community Embedding with Community Detection and Node Embedding on Graphs

<https://zhuanlan.zhihu.com/p/36924789>

3. learning Community Embedding with Community Detection and Node Embedding on Graphs

<https://github.com/vwz/ComE>

4. <http://sentic.net/community-embedding.pdf>



评价指标

Modularity

标准化互信息 NMI (Normalized Mutual Information)

$$NMI(U, V) = \frac{MI(U, V)}{\sqrt{H(U)H(V)}}$$

$$\begin{aligned} MI(U, V) &= \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} P(i, j) \log \left(\frac{P(i, j)}{P(i)P'(j)} \right) \\ &= \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} \log \left(\frac{N |U_i \cap V_j|}{|U_i| |V_j|} \right) \end{aligned}$$

假设对于 n 个样本点的两种标签划分为 U 和 V 。熵为划分集的不准确性

$$H(U) = - \sum_{i=1}^{|U|} P(i) \log(P(i))$$

$$H(V) = - \sum_{j=1}^{|V|} P'(j) \log(P'(j))$$

$P(i) = |U_i| / N$ 表示任取一个样本划分为 U_i 的概率

$$P'(j) = |V_j| / N$$

实时社区发现

1. 基于个体稳定度博弈的动态社区发现算法研究

<http://www.deliwenku.com/p-84779.html>

2. 基于标签传播的实时社区发现算法研究



<http://www.doc88.com/p-1055432963471.html>

3. 基于 `spark streaming` 的动态社区发现及其在个性化推荐应用中的研究

<http://www.doc88.com/p-6991331093268.html>

4. 一种基于局部模块度的增量式动态社区发现算法

<http://www.doc88.com/p-9146960189736.html>

5. 基于标签传播的实时社区发现算法研究

<http://www.doc88.com/p-1055432963471.html>

6. 图流合璧——基于 `Spark Streaming` 和 `GraphX` 的动态图计算

<http://wenku.baidu.com/view/fa65268159eef8c75fbfb3a6.html?re=view>

参考佳文

1. [深度学习时代的图模型，清华发文综述图网络](#)

2. 社区发现(Community Detection)算法

<https://blog.csdn.net/cleverlzc/article/details/39494957>

3. 社区发现算法 `FastUnfolding` 的 `GraphX` 实现

<http://www.aboutyun.com/forum.php?mod=viewthread&tid=19817>

4. 谱聚类 (`spectral clustering`) 原理总结

<http://www.cnblogs.com/pinard/p/6221564.html>

5. `GraphWave`: 一种全新的无监督网络嵌入方法

6. 社团划分结果评估指标: `Q`、`ARI`、`NMI`

https://blog.csdn.net/DreamHome_S/article/details/78167267

7. 论文解读: [从乌合之众到群体智慧的一步之遥](#)

8. `2017CIKM - Network Embedding` 专题论文分享

<https://yq.aliyun.com/articles/294450>

9. `node2vec: Scalable Feature learning for Networks`

<https://www.jianshu.com/p/a9a2ed8b98be>

10. `node2vec: Scalable Feature learning for Networks`

<https://arxiv.org/pdf/1607.00653.pdf>

11. `node2vec: Scalable Feature learning for Networks` 阅读笔记

<https://zhuanlan.zhihu.com/p/30599602>

12. 基于 `PageRank` 的复杂网络社区发现

https://blog.csdn.net/qq_32284189/article/details/80827899

13. `Modularity` 的计算方法——社团检测中模块度计算公式详解

http://www.yalewoo.com/2017/03/10/modularity_community_detection/

14. 社区发现算法 - `Fast Unfolding (louvian)` 算法初探

<https://www.cnblogs.com/littleHann/p/9078909.html>



15. [论文 | 蚂蚁金服亮相数据挖掘顶会 KDD 2018，这些你不可错过！](#)



机器学习在反欺诈中应用



分享嘉宾：**陈德建** 新浪金融 高级模型算法研究员

整理编辑：**侯美娟**

内容来源：DataFun AI Talk 《**机器学习在反欺诈中应用**》

出品社区：**DataFun**

注：欢迎转载，转载请注明出处。

当前机器学习在金融行业已经运用非常广泛，尤其在金融业的信贷领域。在实践中，欺诈与信贷业务强相关，所以，反欺诈变成机器学习在信贷领域的一大应用。反欺诈一般会用到机器学习、深度学习以及图谱关系，其中，机器学习与深度学习多是针对个人欺诈风险，而图谱关系则多用于团伙欺诈风险的识别。因此，本次分享将介绍欺诈风险的特点以及机器学习、深度学习和图谱在实践中的应用。



一、关于反欺诈

（一）欺诈风险简介

在信贷领域有两类风险，一类是信用风险，一类是欺诈风险。信用风险主要是对借款人还款能力和还款意愿进行评估，而反欺诈则是对借款人的目的是否正当进行判断。

一般借款人出现信用风险，金融机构可通过风险定价和自有备付金进行防范，风险可控性较大。而当借款人在借款时便以骗贷或骗钱为目的，且金融机构未能及时识别欺诈，则会出现未能通过借款赚取利息，反而被骗走本金的情况，尤其无法识别团伙欺诈时，会在短时间内遭受非常严重的后果，金融机构面对此类风险的可控性便会非常小，所以欺诈风险是金融机构零容忍的。

（二）反欺诈生命周期简介

信贷反欺诈要从防御开始做起，所以应从用户申请到放款整个生命周期的各个阶段特点进行分析，针对各个阶段采取具有针对性的反欺诈措施。其生命周期可参见下图：





反欺诈的几个方面



如上图，设备与网络防御是反欺诈的第一步，可通过设备和网络层面的检测，防止用户利用设备对金融机构进行欺诈；在第一层进行防御了部分欺诈用户后，再从用户行为层、业务频次层及业务事件异常层识别用户是否有欺诈行为，再经过前四层之后还会通过复杂网络对团伙欺诈进行识别。

二、机器学习应用

（一）反欺诈规则的缺点

反欺诈一般通过两种方式，一种是设定规则，另一种是通过算法。

规则在反欺诈实践中应用也较多，但是缺点也明显，主要表现为：

1. 策略性较强，命中直接拒绝，而且黑名单本身的误伤性也较强；
2. 无法给出用户的欺诈风险有多大；



3. 未考虑用户从信用风险向欺诈风险的转移，尤其是在行业不景气时。

以上缺点机器学习可以进行有效的避免，如可计算用户的欺诈概率有多大，从而采取一定的措施争取客户，而不是直接拒绝，同时也可以通过模型计算用户从信用风险转移为欺诈风险的概率，从而金融机构可及时进行风险控制与准备。

（二）机器学习有监督模型

评分卡一般运用在信用风险评估，如：A 卡（申请评分卡）和 B 卡（行为评分卡）等，反欺诈也会运用有监督学习，如评分卡（F 卡），具体如下：

一般有监督机器学习问题：

假设训练数据

$X = \{x_1, x_2, \dots, x_n\}$, n 个样本

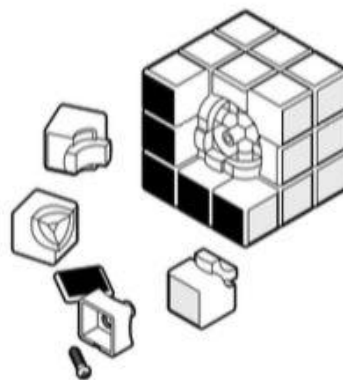
$Y = \{y_1, y_2, \dots, y_n\}$

损失函数(loss function)

$L(F(X), Y)$

目标，寻找一个 F

$F^* = \operatorname{argmin}_F L(Y, F(x))$



做模型的过程中，特征很重要，特征决定模型的效果。反欺诈模型需要从欺诈的角度来做特征，要注意与信用特征区分开，以免与 A 卡和 B 卡的耦合度过高。模型算出的多是概率，一般会将会用模型算出的概率映射到分数，具体如下图：



评分卡分数的计算

评分卡模型的最终产出是分数，且与违约概率负相关

$$\text{score} = \text{BasePoint} + \frac{\text{PDO}}{\ln(2)} (-y)$$

其中， $y = \text{logit}(p) = \log\left(\frac{p}{1-p}\right)$

Base Point: 基准分，无实际意义

PDO : Point-to-Double Odds，好坏比每升高一倍，
分数升高PDO个单位



（三）部分常规机器学习在反欺诈中应用

在反欺诈中用到的机器学习主要有下图几种。其中，iforest 通常用来做数据离群点的异常检测，在应用方面，金融公司可根据自身的规则和算法，将检测出的离群点在评分卡入模的数据方面，进行加权或算法调整。svm 通常也用来做异常检测；arima 则用来作时间序列预测分析；根据现在信贷数据坏样本较少的特点，knn 和 kmean 可以用来做聚类；随机森林则是在做异常检测时进行分类，以上机器学习可通过博客进行更多了解，此处不进行深入讲解。





三、深度学习应用

此部分对人工神经网络（ANN）和时间序列进行简单介绍。

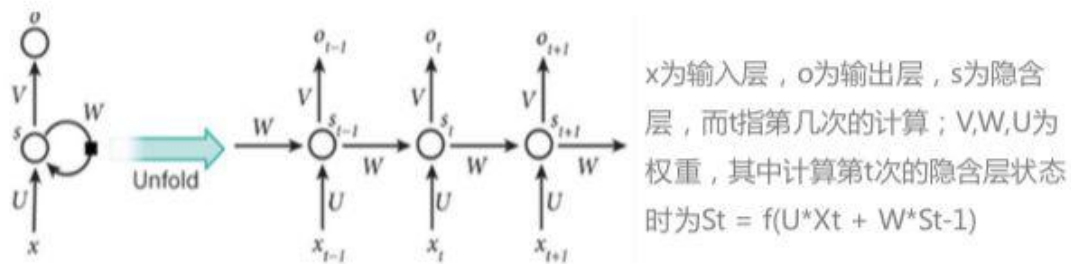
神经网络通常需要大量彼此连接的神经元，每个神经元通过特定的输出函数，计算处理来自其他响铃神经元的加权输入值。神经元质检的信息传递强度，通过加权值定义，算法会不断自我学习，调整加权值。神经网络算法的核心是：计算、连接、评估、纠错和疯狂培训。

时间序列部分介绍 RNN（循环神经网络）和 LSTM（长短记忆循环神经网络）两种算法。

LSTM 是 RNN 的优化版，在特征较多时，RNN 计算量会呈指数式增长，其计算复杂度也会增加，如下图：



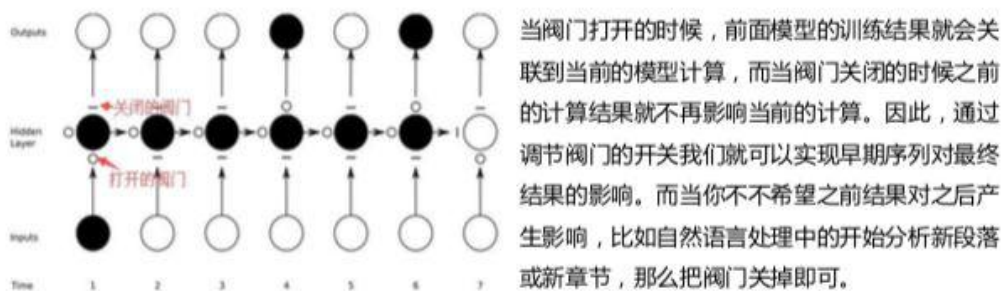
时间序列预测分析就是利用过去一段时间内某事件时间的特征来预测未来一段时间内该事件的特征



特点：依赖事件先后发生的顺序 缺陷：计算量会呈指数式增长，计算复杂度增加

$$s_t = f(U * x_t + W_1 * s_{t-1} + W_2 * s_{t-2} + \dots + W_n * s_{t-n})$$

LSTM 则是在 RNN 的结构以外加入遗忘阀门 (forget gate)、输入阀门 (input gate) 和输出阀门 (forget gate)，其通过这些阀门节点实现记忆功能，改善了 RNN 在计算过程中会出现的问题，如下图：



时间序列在信贷中有两个比较重要应用场景，一是 B 卡（行为评分卡），一是异常检测。

我们着重介绍 LSTM 在这两个场景中的应用。在行为评分卡的应用中，当用户在金融机构进行多次借款时，可以将其以往的借款行为通过统计方法或其他相关方法生成 embedding 进行 LSTM 计算。异常检测的应用可参考下图：





另外在使用 LSTM 时需注意 4 点：

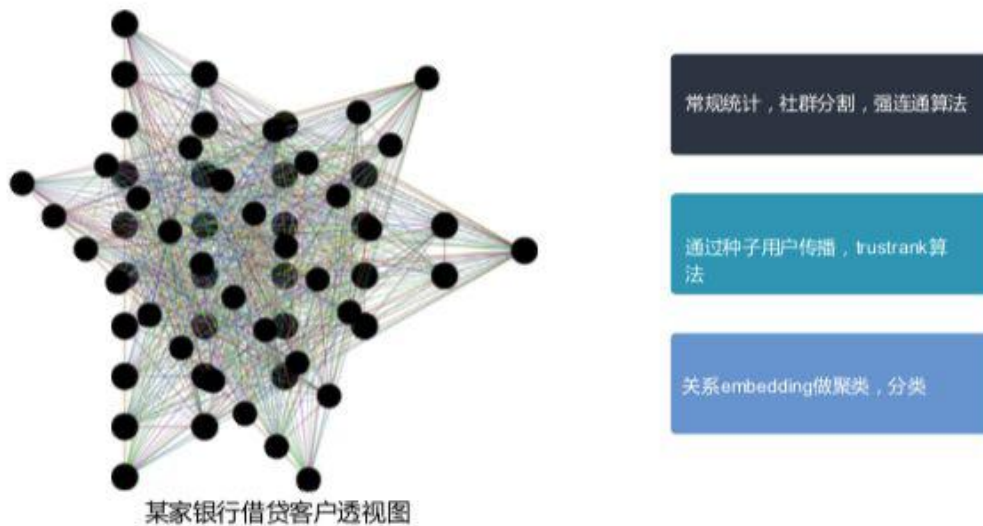
1. 应限制每一个时间序列 embedding 的长度；
2. 对缺失数据做补 0 操作；
3. 针对离散变量的 embedding 尽量不要做 onehot 处理；
4. 样本量少时，应通过仿真模型进行异常检测评估，仿真模型能够有效解决信用风险转欺诈风险的导致模型失效的问题。

四、图谱相关应用

图谱主要用来防止团伙欺诈，也可以根据用户的周边关系判断用户的好坏概率。此次介绍三种图谱关系在反欺诈中的应用。



图谱关系在反欺诈中应用



其中，常规统计一般不会直接用规则，而是将规则做成特征，再带入模型进行统计。比如一度联系人中有多少黑中介，一度联系人中的逾期人数有多少，此类特征的 KS 较高且有效。常规统计一般会用到社群分割和强连通算法。而种子传播层面则需要用到 trustrank，关系 embedding 则可以通过衍生变量将关系向量化，将向量化的关系带入模型进行进一步的分析和统计。

（一）常规统计

因为资源限制或提高投资回报率的原因，黑产一般会最大程度的利用已有资源，比如，重复的使用现有设备和信息进行多次贷款申请，这样就会出现同一手机号码、登录 IP 或硬件设备出现在多个申请信息中，形成关联网络。常规统计的运用，是通过将数据进行关联，形成



关系网络图，然后使用社会关系网络分析工具，分析关系网络图中是否有大量共用设备等拓扑结构。

（二）复杂网络 embedding 算法

有时候机器无法识别信息，需要将信息向量化（embedding），将信息向量化后才可以做后续的计算操作。embedding 的方法有很多，此次仅介绍 node2vec 一种。node2vec 的原理是前端为随机游走（random walk），后端为 word2vec。random walk 则采样，将概率最大的关系采样出来并生成类似文本的序列数据，这类序列数据相当于词的共现性，对词的共线性可以做 word2vec，这里的 word2vec 与 NLP 的 word2vec 无差异。Embedding 后会生成 50 维到 128 维的向量，之后进行聚类和分类的操作，具体如下：

复杂网络embedding算法

word2vec 其实是根据词的共现关系，算词和词之间的概率，将词映射到低纬度向量，并语料中的信息。

DeepWalk 算法其实和word2vec算法很类似，应该是借鉴word2vec算法吧或者是照搬word2vec算法，对图从一个节点开始用random walk来生成类似文本的序列数据，然后将id做为一个个词，始用skip gram训练得到向量。

node2vec

node2vc在deepwalk算法的基础上，定义了一个bias random walk 的策略生成序列，算是改进了一下算法，后面还是走的skip gram去训练

（三）trustrank 算法



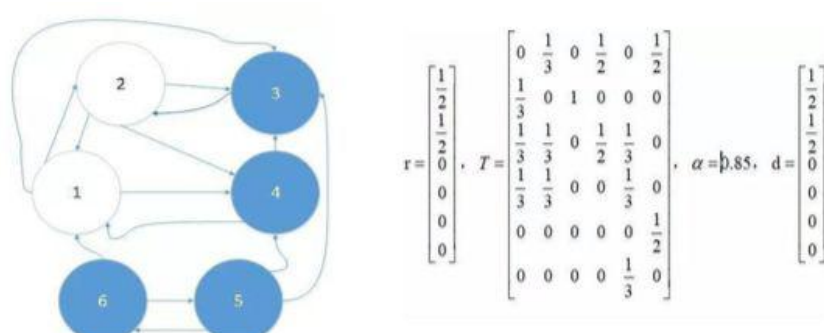
Trustrank 是 pagerank 的升级版，当前我司用的 trustrank 并不是传统的 trustrank，而是改变其中的某些算法。trustrank 是传播关系的一种算法，根据人与人的关系进行判断和识别。比如，小红和小明是情侣关系，当小红违约时，小明的违约概率会增大，根据类似传播关系用来做定额和模型的衍生变量。Trustrank 的使用需要建立起图谱关系，数据量小时，spark 的 sparkgraph 对 trustrank 的图谱关系支撑较好。



Trustrank 涉及种子用户（含白种子和黑种子）的定义，当前的大多使用中只有一种传播方法，也就是白种子只传播白用户，黑用户只传播黑用户。但是其实可以进行变量的衍生和算法的改进，比如一个用户既跟黑中介有联系，又跟高净值用户有联系。

以下图为例，trustrank 为种子用户（下图的 1 和 2）定义一个初始值，每次传播后会改变矩阵的值，最后收敛得到 trustrank 的分。



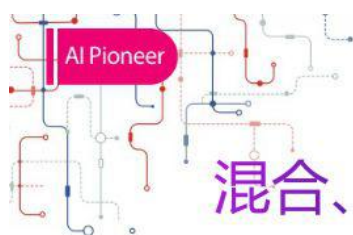


通过不停的迭代，直到都收敛。公式： $r = \alpha \cdot T \cdot r + (1 - \alpha) \cdot d$ 衰减因子，一般取值0.80或0.85，关系矩阵T,d为种子

以上图谱关系可以进行改进和升级，如在传播的过程中可以加入通讯录关系，号码通，老乡、目前所在地、兴趣、职业等，形成以用户画像传播的好坏用户，在不同的用户画像传播中做不同的权重传播。



混合、安全对抗下人工智能设计原则



2018 AI Pioneer Conference

混合、安全对抗下人工智能设计原则

谭炽烈 同盾反欺诈算法总监
chilie.tan@tongdun.cn



本文由 DataFun 社区根据同盾科技反欺诈算法总监谭炽烈老师在 2018AI 先行者大会中分享的《混合、安全对抗下人工智能设计原则》编辑整理而成。



目录

同盾公司简介

反欺诈面临的高对抗问题

人工智能应用的设计原则

反欺诈解决方案

今天分享的内容分为以下几个方面，首先是同盾公司简介，反欺诈面临的高对抗问题，然后根据同盾的实践经验讲一下反欺诈解决方案、人工智能应用的设计原则等。

一、同盾公司简介





智能风险管理引领者

成立于2013年，总部位于浙江杭州，是国内专业的第三方智能风控和分析决策服务提供商。同盾坚持AaaS(智能分析即服务)的风控理念，将人工智能与业务场景深度结合，为银行、保险、汽车金融、非银行信贷、小微企业、基金理财、三方支付、航旅、电商、O2O、游戏、社交平台、直播等多个行业客户提供高效智能的信贷风控、反欺诈及营销整体解决方案。

行业、媒体及政府认可，助力诚信体系建设



同盾科技成立于2013年主要做第三方智能风控和分析决策引擎，为金融、O2O等十来个行业提供风控、营销、反欺诈服务。专攻领域有区块链、大数据处理、网络欺诈分析、机器学习、深度学习、人工智能、自然语言处理等。

二、反欺诈面临的高风险问题



身边随时发生的欺诈



AI 遇到的问题有薅羊毛、交易里面的反欺诈，身边有很多欺诈行为，如账号盗用（钓鱼网站、伪登录网站）、垃圾注册（薅羊毛，通过注册大量账号，规整利益到一起），盗卡盗刷主要是针对银行，ATM 磁卡机侧漏获取磁条信息中的账号密码。群控机器就是有很多账号，将账号和设备进行关联，可以通过风控规则进行拦截，电信诈骗就是通过电话或者是短信的方法持续去引导和诱骗受害者，进入到紧急危险场景，骗取金钱。金融卡套现主要是卡商和机构勾结，商用风险和信用卡风险类似。



同盾对全网30亿设备进行欺诈团伙画像分析



交易欺诈已经成为一种黑色产业链



同盾对全网 30 亿设备进行欺诈团伙画像分析，发现每日欺诈访问比例 8.55%以上，增长趋势逐渐往高科技团伙作案发展，团伙规模逐渐扩大。这里有个提示是当一个欺诈团伙由小发展到大，需要尽快将其拦截，不然后期危险会越来越大。作案手段逐渐多元化，每个行业的欺诈手段和方式都不样，场景也不一样。欺诈团伙分工也是越来越来明显，早期只是对漏洞进行攻击，现在做成一个产业的分工。控制很多虚假号码，身份证无法虚假就是搜集，手机



号码等通过群控来薅羊毛，专业攻击软件实现目的，互联网时代很多营销活动初衷是好的，但是设计体系没有完善，黑产就会利用这一点，包括前段时间土耳其汇率下降。

高对抗场景下的攻与防



如果要对风险进行一个防控，要做几个事情。首先要知道黑产是什么、到底是怎么做的、目的是什么，第二个要知道系统有哪些漏洞，第三个就是有哪些工具去防控这个事情。如下图所示，疑似黑产发生攻击时，有四个时刻。第一个就是攻击发生，平台受到威胁后会进行攻击情报获取、异常预警快速识别风险。平台发现风险、策略分析、拦截风险。当黑产发现被拦截，可能会想办法绕过拦截继续进行攻击。这个过程是不断循环的，不断迭代更新，技术对抗也不断加码。从攻击发生到发现风险，平台要尽早发现并预警风险，因此需要不断完善预警系统，并收集情报；从发现风险到拦截风险，即我们需要分析数据现状以及现在攻击的态势是怎么样的，通过分析产生新的策略并实施拦截。黑产拦截有几种方法，一种就是实时，通过黑白名单机制。还有就是通过无监督或者半监督的方法，找到团伙特征来做一些分析，由于数据和时效性无法实时上线，因此需要无监督方法实现，具体包括图的计算，更高维的有监督算法做一个分类。



拦截到风险到黑产发现之前，可以做一些随机策略从而延长机制。从攻击发生到拦截，上面一部分平台是受攻击的，底下是得到防御和保护的阶段。对于系统希望上面时间越短越好，下面时间越长越好，这样平台受损就少。黑产的攻击有好多种，而且同一个平台遭受不同欺诈团伙攻击，发生时间、渠道、团伙大小、手段等都可能不同。

针对薅羊毛场景，具体分工为：商家会发布一个优惠，黑产获取信息发掘有利可图进行攻击。平台可能只有有限优惠券可以领，这样黑产会让很多人做这个事情，将其归拢一起，获取利益最后做利益分配。

三、反欺诈解决方案

上面梳理了黑产是怎样的，有哪些手段，接下来讲我们有哪些工具来进行防御。人工智能并不是无所不能的，当然也不是完全依赖人工，因此协调人工运营、决策、人机协同，达到更快、更准的理念。结合业务场景、流程、风险问题设计，通过数据的驱动获得整个人工智能算法的应用。如下图所示，在这个过程中可用的工具分为无监督和有监督两种。有监督包括分类、回归等，对于反欺诈大部分是分类，回归用的较少，分类算法有 LR、贝叶斯、随机森林或者 GBDT。无监督方法更多用于预警、关系挖掘（图挖掘 LPI、标签传播）等。



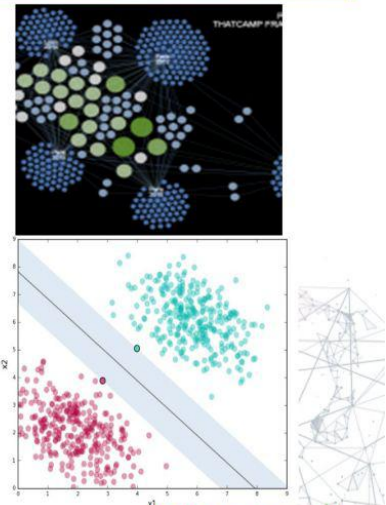
机器学习算法分类

■ 无监督——无标签

- K-means
- 密度
- 时间序列
- 图计算

■ 有监督——有标注

- 线性，逻辑回归 / 分类
- 贝叶斯分类
- 决策树
- 随机森林
- SVM
- Adboost



2018 AI先行者大会

机器学习算法开发流程如下图，(1) 首先我们要定义明确业务目标，目标明确能够很好的提升算法水平。依据业务场景特征、行为做一些对抗分析，如薅羊毛在刷单和众包里面都有，也有可能群控设备，但是其反馈是不一样的；(2) 然后数据分析，需要明确黑白样本有没有，数据缺失率、准确率如何，能否支撑目标的运行；(3) 特征选择，这决定了算法的效果。并不是所有的算法都可以使用，不同的特征只适应特定的场景，需要依据业务目标和拿到的数据选择特征；(4) 算法训练，前期准备好，时间可控，一般可收敛，如果不收敛就需要在前期工作做些调整；(5) 验证测试，有监督中验证测试比较简单，分类就统计准确率和召回率。无监督中更多需要人去确认分类是否有意义，能不能解决问题；(6) 迭代更新。



机器学习算法开发流程



四、人工智能应用的设计原则

为了尽早发现风险，应该如何做呢？

原则 1：预警及早发现风险

- 实时监控与离线挖掘结合，互为补充
- 可靠的情报侦察
- 时序分析：事件序列的模式，危险时段，特征编码
- 趋势分析，均值预测
- 统计监控变量，发现状态异常

- (1) 实时监控与离线监控都是需要的，实时监控是线上调用量的结果、时间态势，包括统计变量，但是这种无法得到多维信息。需要离线组合弥补
- (2) 可靠的情报侦察；
- (3) 时序分析：事件序列的模式是否异常，危险时段，依据事件特征编码进行挖掘；
- (4) 趋势分析，到底是否合理，均值预测；



(5) 统计监控变量，发现状态异常。

举例说明，时间序列指标的异常监控如下图所示，我们可以发现有几个攻击发生，大约 4

月 10 日开始尝试攻击、试探，监测到 4 月 16 日线上实时调用异常，发现监测指标峰值。

我们通过日常预警调用量的方案可以找出攻击发生点，能够快速对攻击事件进行拦截、分析。

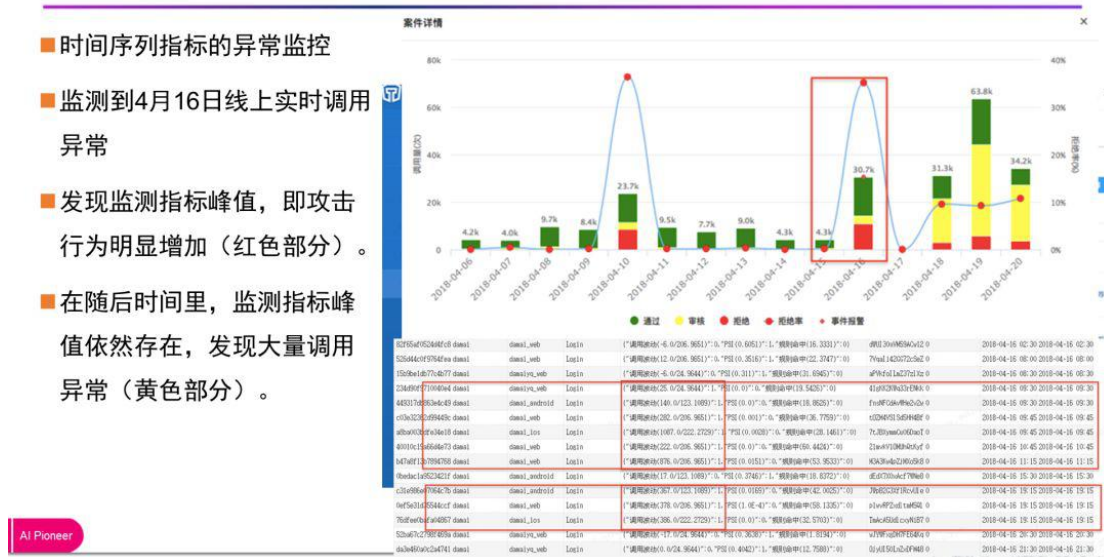
业务数据异常预警：调用量

时间序列指标的异常监控

监测到4月16日线上实时调用异常

发现监测指标峰值，即攻击行为明显增加（红色部分）。

在随后时间里，监测指标峰值依然存在，发现大量调用异常（黄色部分）。



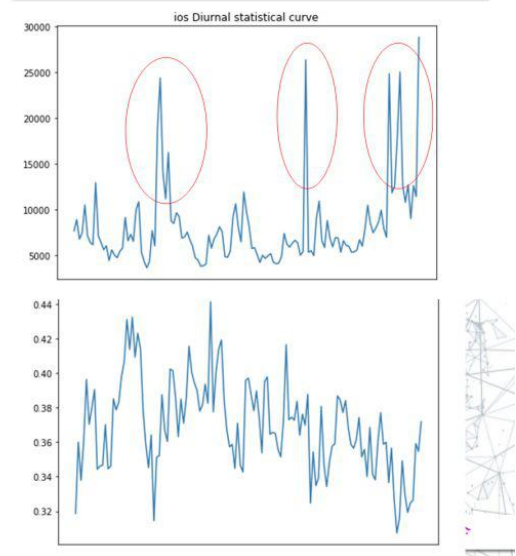
下图所示，终端设备监测指标的时间曲线。上图是用户使用某一终端系统设备参数的用户波

动情况，下图是终端设备参数的占比。从监测数据可以发现这是一个群控现象。



业务数据异常预警：终端异常数据监测

- 图中是终端设备监测指标的时间曲线（5个月）
- 上图是用户使用某一终端系统设备参数的用户数波动情况，下图是终端设备参数的占比。
- 从监测数据可发现，发生剧烈波动的几个异常波峰。



原则 2：需要多特征、多维特征补充分析准确度

- 事件前后关联分析：注册，登录，搜索，支付
- 地理空间分析，wifi，gps等地理位置，经纬度，地址文本处理
- 用户操作行为，传感器，页面操作行为，
- 画像分析，黑产用户画像，设备画像，档案画像，IP画像，手机画像
- 关联分析，团伙关联，有价值资产关联，交易双方关联，事件关联，相似特征聚类

(1) 事件前后关联分析：目前大都是单场景策略拦截，但是可以在注册和登录搜集特征，补充到后续交易和支付场景里面，整体做特征补充。

(2) 地理空间分析：wifi，gps 等地理位置、地址文本处理等相关处理；

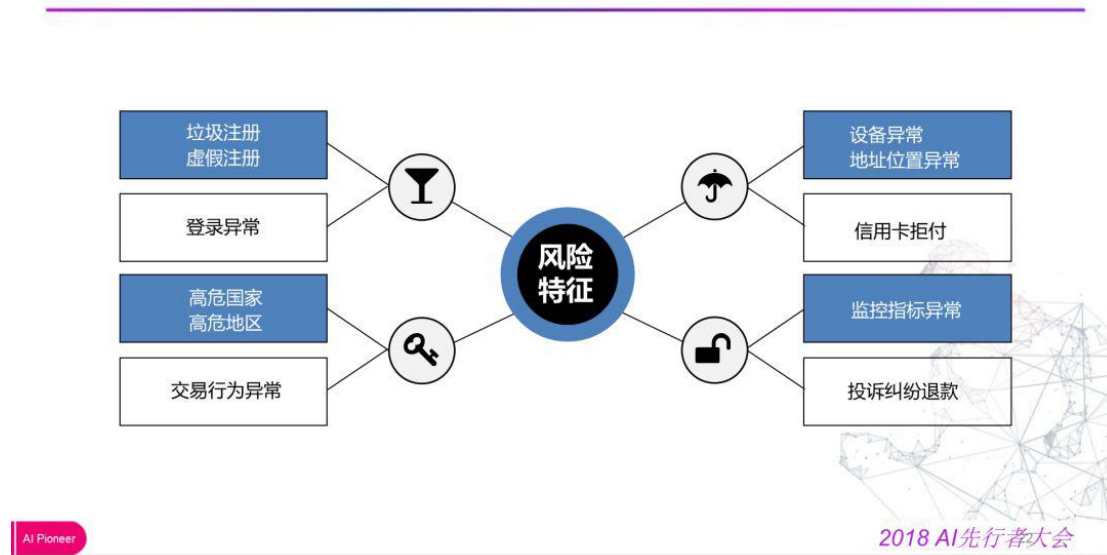
(3) 用户操作行为：不是针对个人，而是群控设备的人和机器的差别；

(4) 画像分析，黑产用户画像，黑产流窜于那些平台，是否有跨行业可能性，同一行业相互平台是否流窜。IP 画像，这个很难做因为可变，IP 可能是公共流量出口。



(5) 关联分析：团伙关联，有价值资产关联，交易双方关联，事件关联，相似特征聚类形成图谱进行分析。

欺诈典型风险特征



欺诈风险特征可能做得更细，上图是支付中可能会遇到的风险，也有更深的。如垃圾注册或者虚假注册，地区可能是高危国家或地区的，团伙作案空间上是有流窜的，相互是有关联，就是黑产地域性。还有就是交易行为异常，这是最重要的，这种很难逃避设备监控和 IP 监控。

同盾坚持的风控理念是跨行业联防联控，针对网络团伙欺诈流窜现状，跨行业联防联控，挖掘行业横向数据维度信息。

原则 3：人机协同的立体反欺诈体系，让策略人员轻松应对



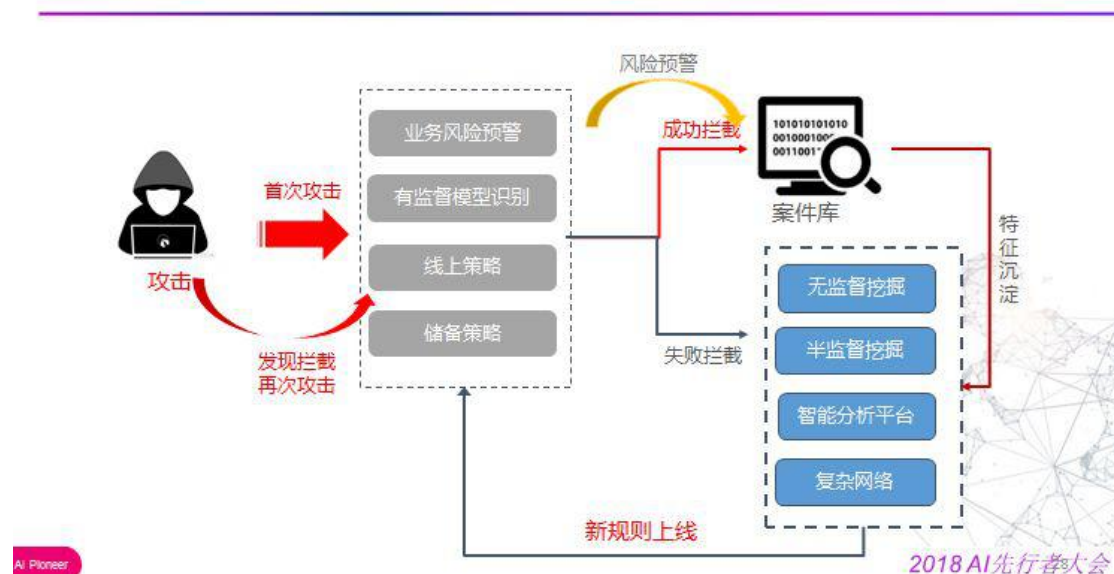
- 神经反应：脊椎神经反应，脑回路神经反应
- 监督算法：精确识别，训练周期、训练更新、回路时间长，依赖标签数据，专业人员稀缺
- 业务规则，专家分析决策经验，策略模板库
- 黑白名单，反馈机制与生效
- 混淆策略应用，让防控效果延长



神经反应：脊椎神经反应（立即反应）针对黑白名单，反馈及时、拦截快速；脑回路神经反应，这种是智能系统，因为系统从算法到上线周期很长，数据来源获取长，训练时间长，还有决策、上线。还有缺乏专业人员，目前周期较长。

通过业务规则，专家分析决策经验能够及时拦截事件，因此是立体的防控体系。

构建多层次、纵深的立体反欺诈体系

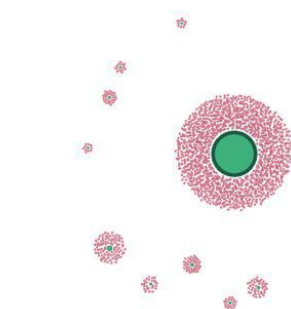


项目发展是规则决策到有监督再到无监督发展关系，当出现风险，该规则很快就能进行分析，有监督在有标签的时候就可以做，而无监督需要经验和业务体验的特征，这样做才有意义。欺诈可能有已知欺诈，这种通过业务规则就可以判断，对于未知欺诈模式需要有个异常监测，团伙需要有个网络的分析，挖掘出所有信息。发生攻击时，防护是靠业务预警系统还有线上



策略、黑白名单储备策略，离线时利用漏过的特征，通过失败和成功拦截构建无监督和半监督拦截来补充线上策略。

相似特征聚类：垃圾注册账号



■ 同盾积累跨行业大量的黑产资源库，包括设备指纹、IP、电话号码等。图中分析和挖掘注册账户之间的关联，能有效发现规模垃圾注册。

AI Pioneer

[illegible]

■ 半监督机器学习算法挖掘异常事件和行为模式，配合人工审核和有监督学习算法，构建立体防控体系。

2018 AI先行者大会

五、最后总结：

在与黑产作对抗时，推演是非常重要的。结合业务场景发现蛛丝马迹，挖掘有效特征；分析纵向事件序列、黑产画像、档案，设计多种智能算法框架；通过联防联控体系，挖掘横向信息；构建多层级、多级别的立体智能风控体系。



自动化特征工程和自动建模在风控场景的应用



分享嘉宾：**蒋宏** 融 360 风控业务模型部负责人

编辑整理：**梁晓**

内容来源：DataFun AI Talk 《**自动化特征工程和自动建模在风控场景的应用**》

出品社区：**DataFun**

注：欢迎转载，转载请注明出处。

一、背景和问题

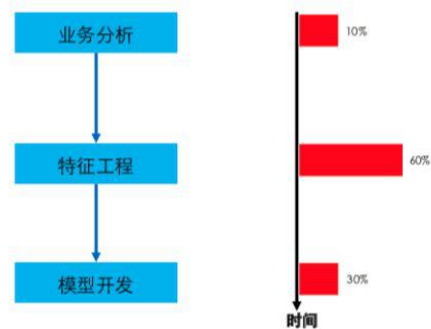
目前，模型开发的流程越来越规范化，通常可以分为业务分析、样本准备、特征工程、模型构建、模型评估及监控这几个步骤。其中，特征工程和模型构建在建模的整个流程中依然非常耗时，并且非常依赖于模型开发者对业务的理解及数据处理的能力。

在目前实际业务场景下，面临的最大的一个问题是，如何快速地构建起一个质量相对不错的模型，以适应业务的快速发展。传统的风控建模周期较长，通常要 20 天左右的时间才能达



到上线的要求。其中，特征工程的耗时在整个建模过程中会占到大约 60%的时间，且这个过程挑战就是人工提取特征的复杂性和高时间耗费性。自动化特征工程能够对必要而繁琐的工作进行自动化处理，从而优化机器学习模型的构建和部署，这样模型的开发者就可以更多地关注其他重要步骤。另一方面，模型开发过程的耗时在整个建模过程中会占到大约 30%的时间。这个过程的难点并不在于给出一个模型，而是在于同时比较多种模型甚至多种模型组合后，选出效果最佳的模型做主决策模型，其余的模型可以作为备用模型或者比较对象。这个过程如果用人工去实现，也会消耗非常多的时间，并且效果不一定最优。

● 传统风控建模周期长，其中特征工程和建模过程耗时尤为突出



基于这样的背景，融 360 推出的自动化特征工程和自动建模的方案，把建模过程中耗时最长的特征工程部分和模型开发部分抽象出了一套自动化工具。不仅集成了自动特征工程，还对建模过程中变量的筛选、建模调优、部署上线和模型监控进行了模块化的封装。该工具有如下优点：

- ① 大幅提升建模效率和标准化程度；



- ② 节约人力成本和时间成本；
- ③ 快速达到业务方要求；
- ④ 有利于尝试更多的模型方案和模型快速更新迭代。

通过自动化特征工程和自动建模方案后，模型从开发到上线的时间大幅减少，可以缩短至 5 天左右，不仅简化了开发流程，提升了开发效率，准确率方面也保持较高水准。



通过自动化特征工程和自动建模方案，可大幅提升建模效率和标准化程度



二、自动化特征工程

首先，我们先来看一下手动特征工程是如何完成特征工程这个过程。人工构建特征工程主要依赖领域知识来构建特征，一次只能产生一个特征，这是一个繁琐，费时又易出错的过程。此外，每次进行特征工程的代码是针对特定业务和特定场景的。当我们面对新的需求、新的数据源时，我们需要重构特征，重新进行开发，耗时长效率低。



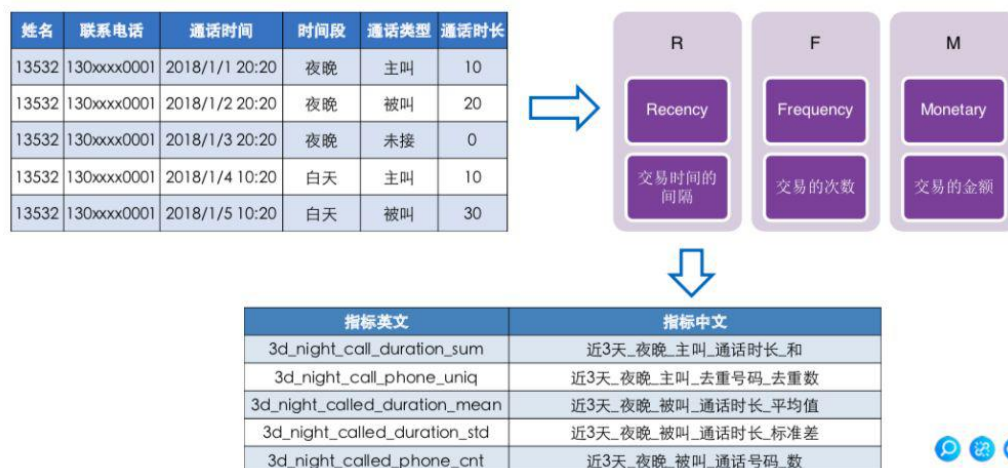
因此，特征工程若要能够实现自动化和标准化，其数据在结构上必须要具有一定的相似性和共通性。根据美国数据库营销研究所 Arthur Hughes 的研究，客户数据库中有三个神奇的要素，这三个要素构成了数据分析最好的指标：

- ① 最近一次消费(Recency)；
- ② 消费频率(Frequency)；
- ③ 消费金额(Monetary)。

在众多的客户关系管理(CRM)的分析模式中，RFM 模型是被广泛提到的。RFM 模型是衡量客户价值和客户创利能力的重要工具和手段。该模型通过一个客户的近期购买行为、购买的总体频率以及花了多少钱三项指标来描述该客户的价值状况。

人工通过RFM模型可以构造出大部分的特征

融360
rong360.com



在融 360 的业务场景下，更多的数据是交易级别的数据。那么，根据 RFM 模型构建出来的特征通常包含基本的统计变量，趋势型的变量等。例如，通话的时间段可以分为白天或者夜



晚，通话类型可以分为主叫、被叫或者未接，交易金额总和、次数等等，这些都属于统计型变量。而类似于近 3 个月比近 6 个月的通话次数减少了多少，这是趋势型变量。

基础数据具有相似性的结构为自动特征提供了可能性



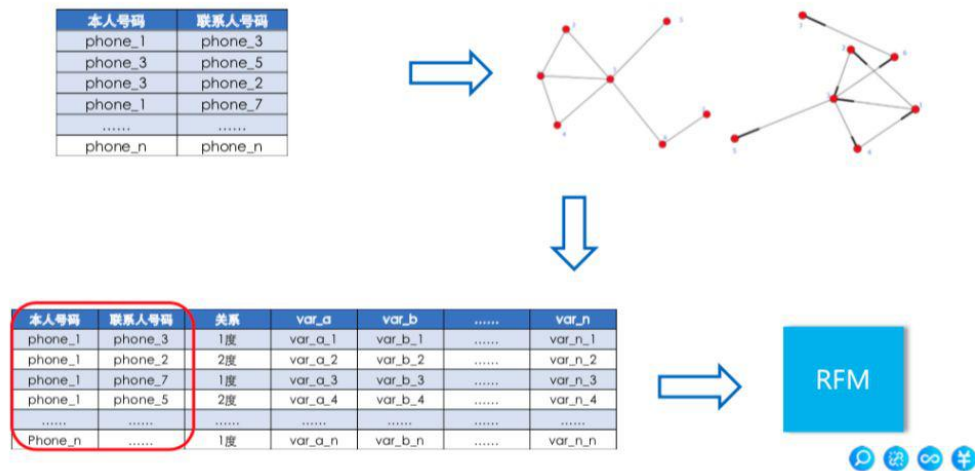
- 处理后的宽表数据在结构上具有一定的相似性，每个数据中都包含**时间型数据**、**分类型数据**、**数值型数据**和**标示型数据**。



不仅是结构化数据，在关系网络中也可以通过 RFM 模型构建特征。如下图所示是一个简单的关系网络。应用 RFM 模型，可以构建出例如某用户有多少 1 度联系人，1 度联系人中是否有过借贷行为，平均借款金额，以及其中有多少人有逾期记录等特征。

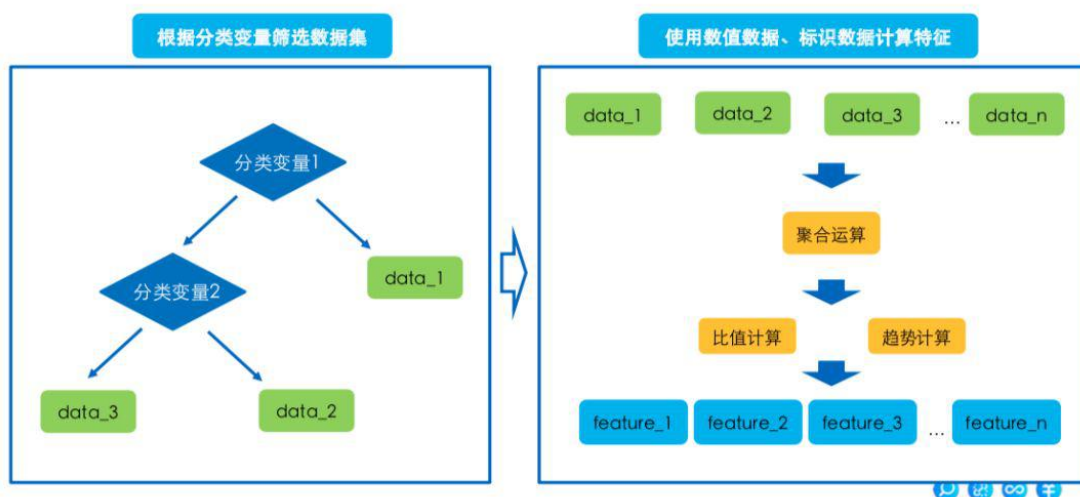


● 即使是关系网络也可以同过转换利用RFM模型构造特征



综合以上分析过程，可以将自动化特征工程的过程抽象成如下图所示的框架。首先，根据不同的数据类型，采取不同的处理方式。如果是分类变量，可以切分成不同的数据模块，用统计型或者标识型的方法去处理，完成第一层的聚合运算，生成基本的统计特征。那么在第二层中，会对基本的统计特征采取比值计算和趋势计算的一些方法，衍生出新的变量。

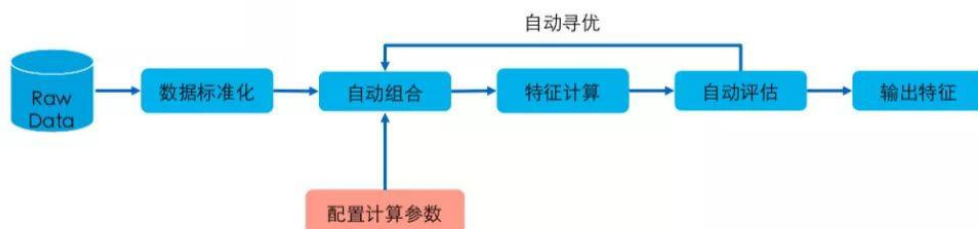
● 抽象公共步骤，构建自动化特征工程核心逻辑



回顾特征工程的整体过程，有了自动化特征生成工具之后，模型开发者关注的重点不再是构建能够通过简单运算得到基本的统计特征，而是能够有更多的时间关注数据本身。而需要人工处理的部分仅是从原始数据到把数据标准化的过程，后续的特征自动组合、计算，以及评估到输出都实现了自动化的过程。

完整的自动特征工程方案

融360
rong360.com



下图所示是某项目中，自动化特征工具和人工构建特征在同一数据集上的比较。可以看到，自动化特征相对人工特征效果明显提升且耗时更短。当然，开发自动化特征工具的目标并不是为了替代人工，而是希望自动化特征工具能够快速并且准确地提供一版基础特征，从而把人力更多地投入到对原始数据的理解以及对业务分析中，或者构建出更多衍生的特征。



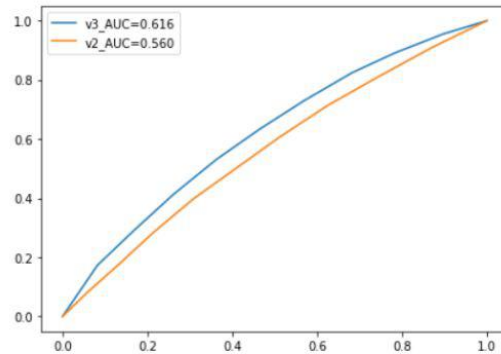


自动特征相对人工特征效果明显提升且耗时短



取用户运营商原始数据，分别用**自动特征（V3）**和**人工特征（V2）**建模对比，在相同的OOT样本上AUC对比：

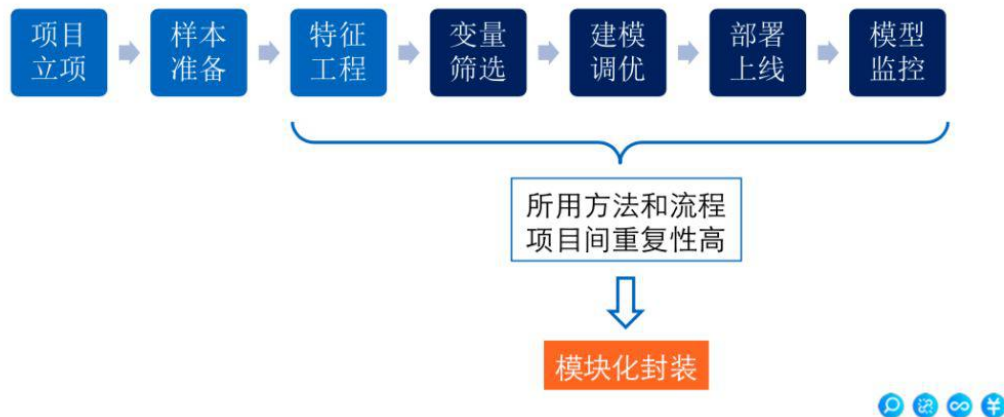
- V3_AUC相对V2_AUC 提升10%
- 特征挖掘耗时减少超过80%



三、自动建模

尽管不同的数据样本和不同的模型选择，得到的结果会有较大差异。但各项指标综合来看，目前不管是国内外的数据竞赛，或者是回归到公司业务本身，xgboost 和 lightGBM 依然是适用性最好，表现最优的算法之一。而在信贷场景下，LR 因其较好的解释性，也被广泛使用。因此，自动化建模工具整合了常用的算法，针对建模过程中通用性较高的部分进行了模块化的封装。这样极大地提升了开发效率，并且可以将更多的时间用于尝试不同的模型。





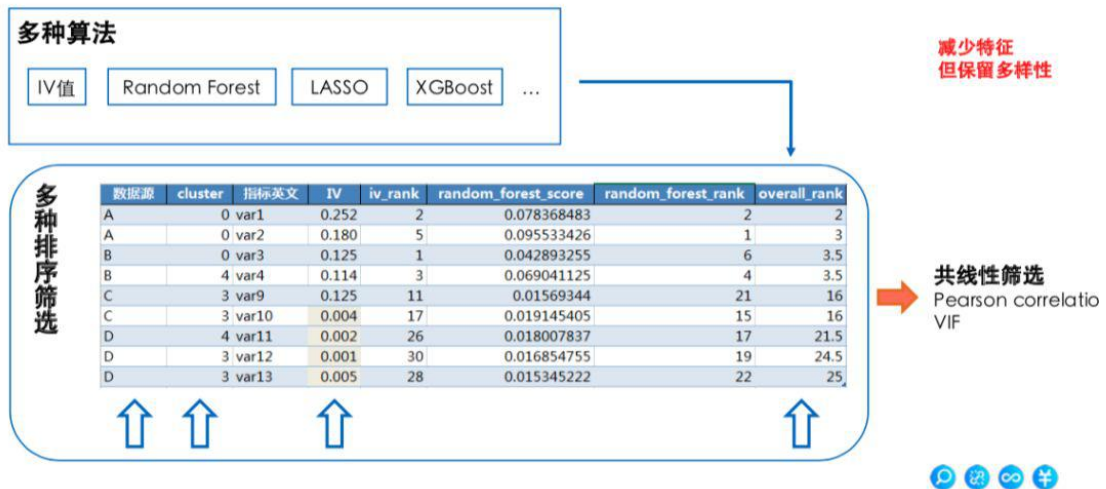
3.1 自动特征选择部分：

首先，自动通过 EDA(Exploratory Data Analysis)对特征进行初步筛选，例如筛选掉缺失率较高，方差恒定不变，在时间维度上不够稳定等特征。其次，自动综合多种常用算法和规则，进一步筛选出有效的特征。例如：

- ① IV 值筛选；
- ② 树模型输出重要性；
- ③ 共线性筛选。

最后，选择出综合排序 TopN，以及各个类别中排名靠前的特征。可以把几千维度的特征降低到几百个维度的范围内，并且在减少特征的同时，保留特征的多样性。





3.2 建模部分

3.2.1 逻辑回归模型(LR)

LR 在信贷场景下，通常用于构建评分卡模型。首先回顾一下逻辑回归实现评分卡的理论基础。

一个事件发生的几率(odds)，是指该事件发生与不发生的概率的比值。若一个用户违约的概率是 p ，则其正常的概率为 $1-p$ 。可以得到：

$$\text{odds} = \frac{p}{1-p}$$

此时，客户违约的概率 p 可以表示为：

$$p = \frac{\text{odds}}{1+\text{odds}}$$



而评分卡的表达式为：

$$\text{Score} = A - B * \log(\text{odds})$$

其中，A 和 B 为常数。通过给定（1）某特定 Odds 时的 Score 值（2）该特定 Odds 值翻倍时 Score 增加值，带入评分卡表达式即可求得 A、B。

而根据逻辑回归的原理可推导出该事件的对数几率如下：

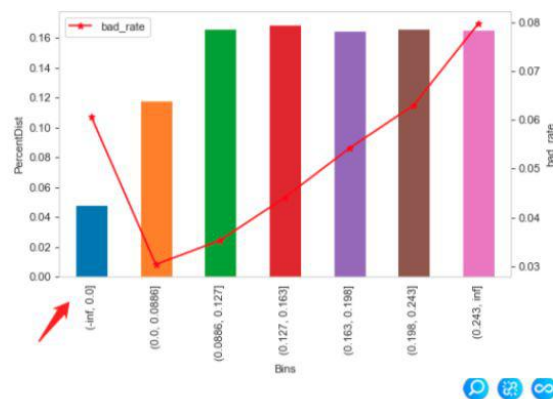
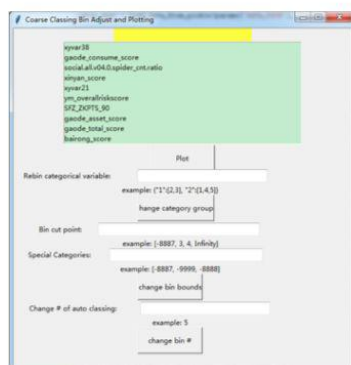
$$\log(\text{odds}) = \log\left(\frac{p}{1-p}\right) = \log\left(\frac{h_{\theta}(x)}{1-h_{\theta}(x)}\right) = \theta^T x$$

因此，构建评分卡的问题转化为利用逻辑回归模型求概率的问题，最后根据概率转化成评分卡的分数。

逻辑回归评分卡模型采用自动分箱（等频率单调性）转woe，同时提供人工调整方式

融360
Rong360.com

- 根据业务含义调整粗分箱，并根据时间维度确保粗分箱稳定
- p值显著



构建评分卡模型中最关键的步骤是 WOE 分箱，分箱的好坏直接影响最后模型的效果。传统的处理方式是，人工根据特征的业务含义，对特征进行一些粗分箱。抽象成自动化工具后，



统一对特征进行等频分箱后做 WOE 变换，并且进行单调性检测。同时，也保留了人工操作的接口，可以自由调整分箱的边界，达到更优的效果。

3.2.2 XGBOOST

xgboost 是目前应用最广泛的算法之一。不仅在多种竞赛场合取得非常好的成绩，在工业界也有广泛应用。在自动化建模的过程中，为了能让 xgboost 发挥出更好的效果，需要针对入模变量做额外的预处理。连续变量虽然可以直接入模，但为了减少过拟合，依然对连续变量进行了分箱处理。而分类变量则不能直接入模，可以通过决策树进行分箱的操作，或者通过各种编码方式对变量进行转换，例如标记编码，one-hot 编码等等。



除了对入模的特征实现预处理，自动化建模工具还集成了更全面的调参方法，包括：设置多组经验参数、GridSearch、RandomSearch 等，这个过程也是人工调参中耗时较长且重复



性较高的步骤之一。

四、自动化模型评估与监控

在完成模型开发后，最重要的一个环节就是模型评估与模型的监控。在各种数据竞赛中，评估一个模型的好坏，通常会用 AUC、KS 等指标。但在实际场景中，模型上线后需要关注的不仅仅是这些模型的评估指标。更加需要关注的是模型整体的排序性和稳定性。在不同的业务场景下，对模型的要求也不一样。例如，在欺诈检测的场景下，会希望模型在低分段有更好的区分度。反之，如果要做授信额度策略，则会期望模型在高分段有更好的区分度。

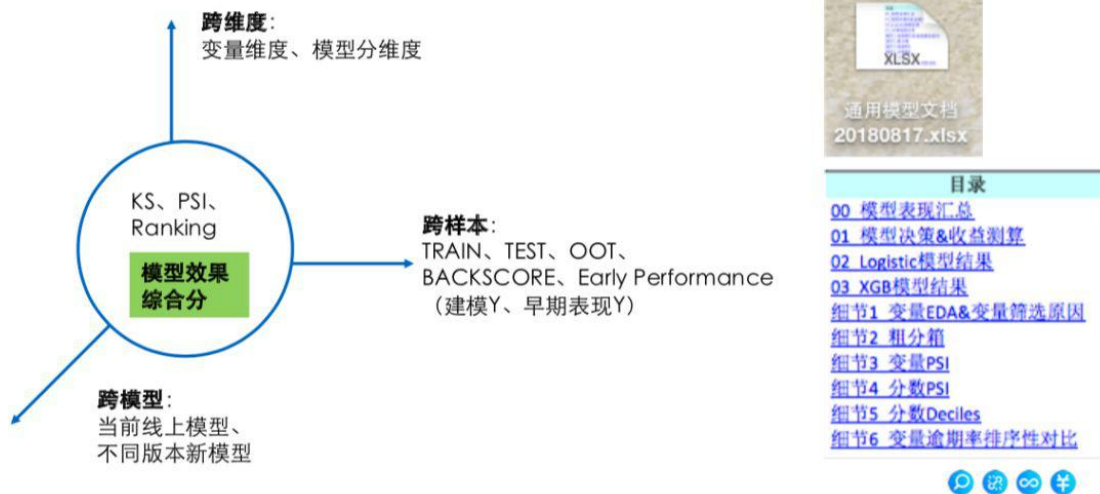
自动化模型评估在实现全面、细致、快速地评估模型效果方面，主要涵盖了以下三个方面：

- ① 跨维度：变量维度、模型分维度；
- ② 跨样本：Train、Test、OOT、
Early Performance 等；
- ③ 跨模型：当前线上模型、不同版本新模型。





自动全面、细致、快速的评估模型效果，给出多个版本模型效果排名



跨维度层面，包括对特征的监控以及特征对模型结果的影响。除了关注模型的基本监控指标 AUC、KS、PSI、Ranking 等，在变量维度，还会关注变量随着时间的变化，排序性是否下降，PSI 是否升高，以及特征是否对模型结果产生了负面影响等因素。

跨样本层面，由于在模型开发的过程中，经常会使用比较成熟的表现期的样本，例如 MOB6 进行建模，但在评估模型的时候，可以用 MOB1 的样本进行跨期测试，用来判断模型的早期效果。

跨模型层面，不同的模型有不同参数，会针对不同的模型进行比较。看当前线上的模型和新版本模型之前的效果差异。

最后，根据以上三个维度的评估，会进行加权求和，输出模型最终的综合评分以及模型相关内容的文档，包括模型表现的汇总、特征分箱的结果、变量的 PSI、变量与逾期率的关系



系等内容。

五、模型部署上线及线上监控

相较于传统的写 python 脚本完成从训练、预测到输出结果的模型上线方式，目前更常用的方法是输出模型文件，线上的部署环境支持读取和调用模型文件。但融 360 开发了针对模型部署的平台，省略了开发模型部署的脚本。模型自动输出配置文件，只需将配置文件放如新模型的文件夹中，就可以轻松完成上线部署，实现自动评分。并且配置文件具有简洁和良好的可读性等优点。

标准化输出 – 模型部署文件：简化部署、提高效率



Before:

- 单独写python脚本进行数据读入、模型预测、输出模型决策和风险评级

After:

- 模型自动输出配置文件。只需将配置文件放入新模型的文件夹中，就可完成上线模型部署
- 配置文件有良好的可读性

```
required_features = ['RG1', 'RG2']

score_rank_pair = [(RG1, RG2)]

loan_threshold_rg = {
    'threshold': 0.5,
    'rg_bound': 0.5
}

mall_threshold_rg = {
    'threshold': 0.5,
    'rg_bound': 0.5
}
```

xgb_bin_to_label.json
xgb_selected_rebin_spec.json
xgb_var_transform_method.json
xgb_xgbmodel.pkl



模型上线后的线上监控也是非常重要的环节之一。如果模型的监控仅依赖于各个模型的 owner，那么会产生非常多的问题，例如不能及时发现 AUC、PSI 等模型指标的变化，缺失对特征变化的监控，模型何时需要重新训练，更新迭代等等。针对这些问题，融 360 开



发了自动化模型监控工具，实现标准输入，标准输出。新模型配置监控不再需要重写代码，只需配置文件即可。

标准化输出 – 模型监控文件：快速实施监控、定期获取监控报告



Before:

- 模型Owner各自统计模型表现，对PSI, KS进行不定时监控

After:

- 新模型监控0代码开发，只需配置文件

指标英文	分箱	建模 N	建模占比
	(-inf, 0]	357	0.101796
	(532.0, 2.0]	351	0.100086
re	(572.0, 0.0]	346	0.09866
re	(601.0, 0.0]	360	0.102652
re	(631.0, 0.0]	343	0.097804
re	(656.0, 0.0]	364	0.103792
re	(681.0, 0.0]	345	0.098375
re	(708.0, 0.0]	344	0.09809
re	(736.0, 0.0]	348	0.09923
re	(779.0,]	349	0.099515

监控输出:

- 变量、总分的详细分布对比和PSI
- 变量、总分的排序性对比和KS
- 定时输出日报、周报、月报邮件

PSI监控配置文件:
- 建模/Benchmark
样本分布
- 分布边界

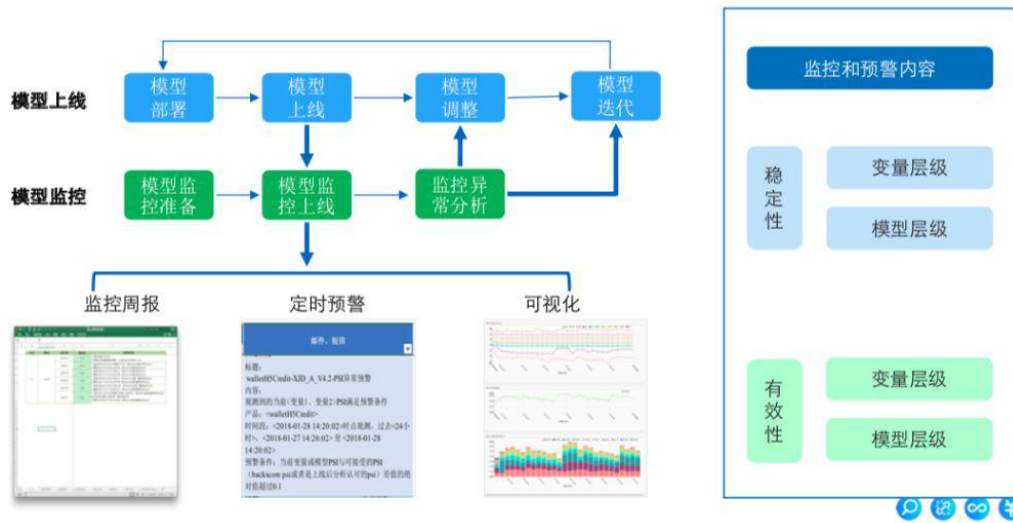


模型的每日监控会与模型上线时候的指标做对比。一方面，整体监控 KS、AUC、PSI、排序性等指标是否发生变化，另一方面，关注每个变量在训练时的分箱与上线后的分箱的差异，看特征的分布是否发生了偏移，实现自动预警。

当监控发生报警时，会提醒模型开发者哪些地方出现了问题，从而能够及时响应。例如，当变量发生偏移的时候，考虑是否是数据源发生了波动，业务客群是否发生变化，是否需要与产品团队、策略团队进行对接等问题。



模型和变量层级的监控和预警体系，覆盖模型生命周期



最后，模型的监控以三种形式输出：监控周报、定时预警以及模型效果可视化。



爱奇艺流量反作弊的“术”与“道”



爱奇艺流量反作弊的“术”与“道”



本文根据爱奇艺张晓明老师在 DataFun Talk 大数据风控系列活动中分享的《爱奇艺流量反作弊的“术”与“道”》编辑整理而成，在未改变原意的基础上稍作整理。



大纲

- 1 流量反作弊介绍
- 2 爱奇艺流量反作弊之“道”
- 3 爱奇艺流量反作弊之“术”
- 4 从系统应用的角度看反作弊体系
- 5 未来展望

今天主要从以下几个方面介绍，首先介绍下流量反作弊相关的介绍，然后是爱奇艺流量反作弊的“道”，爱奇艺流量反作弊涉及到的点很多，起“道”还是蛮贴切的，让大家对爱奇艺流量反作弊有个认知；接下来就是流量反作弊是怎么做的。第四个从系统应用的角度讲如何构建反作弊体系，最后就是未来的展望。

简介



什么是流量反作弊

制造非用户产生，或带有一定目的性的数据



流量作弊的形式

机器作弊：模仿投递日志，调用接口

人工作弊：微信群，QQ群，由专人指导
专业APP，指导操作并与用户分成



两种形式的特点

机器作弊的设定模式相对固定，短期效果明显，
人工欺诈模式不固定，效果与组织规模有关



首先说一下什么是流量反作弊，书面并没有官方定义，我定义为制造非用户产生，或带有一定目的性的数据。非用户产生的就是程序员写的一些代码重复执行投递逻辑或产生一些并非人为产生的数据，带有一定目的性指有些数据并不是机器产生的，很大部分都是人产生的。如一个人在 APP 上重复的操作一个事情，但所有的操作都是合法的（不断打开关闭视频刷视频），这也是一种反作弊。第二点说一下流量作弊的形式，一种是机器作弊：模仿投递日志,调用接口。有些数据并不是投递的，而是访问业务数据库，提供接口访问，这种也算为流量作弊；第二种是人工作弊：微信群，QQ 群，由专人指导，专业 APP，指导操作并与用户分成。这两种作弊的模式机器作弊的设定模式相对固定，短期效果明显，因为是机器循环操作，通过动态 ip，而且速度比较快。人工欺诈模式不固定，效果与组织规模有关。

产业现状



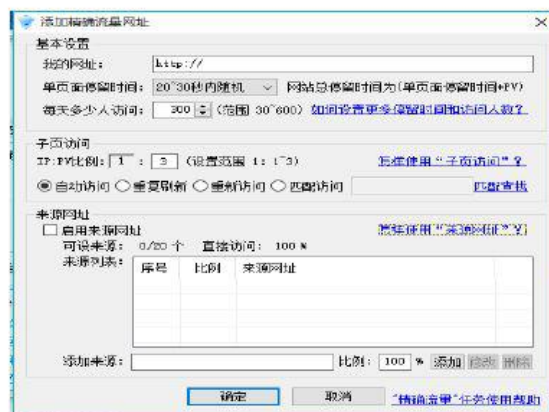
难辨识：刷量行为高度模仿真实操作，极具欺骗性



成本低：动态代理IP获取容易，还有P2P机制的刷量软件（流量精灵）



行动快：有组织有规模

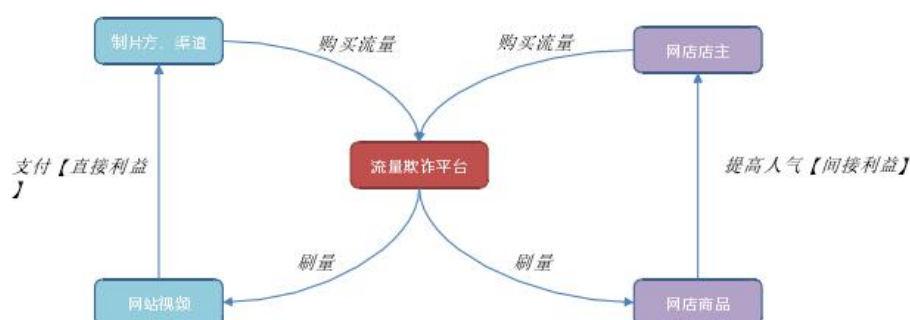


反作弊产业现状第一个就是难辨识，机器作弊越来越趋近现实，很具欺骗性。在平台和作弊方进行博弈的过程中就如同病毒一样不停成长，就很难防范。第二是成本低，动态代理 IP



获取容易，还有 P2P 机制的刷量软件（流量精灵），虽然我们平台能识别，但是如果是个小公司，一开始还是很难识别的。第三个就是行动快，非常的有组织有规模，不是某一个人某一台电脑下一个软件刷一下。

刷量模式



直播刷人气，网站刷广告等其他形式



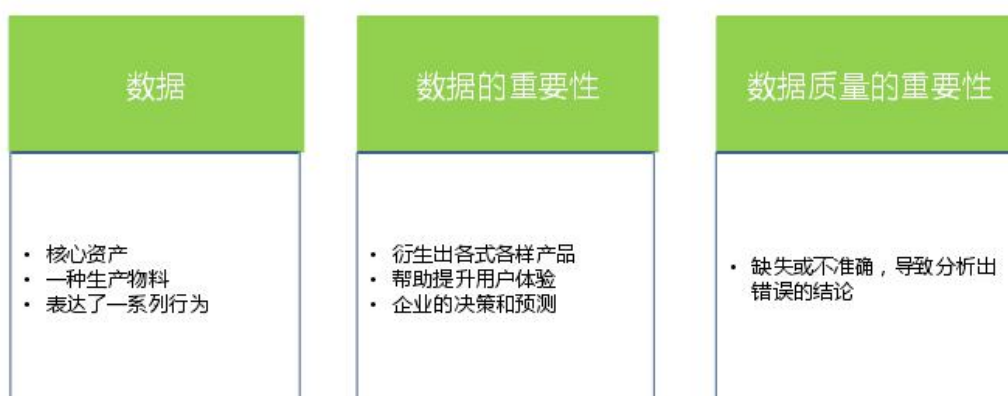
简单介绍一下其刷量模式，中间红色部分是流量欺诈平台，首先是制片方或渠道去流量欺诈平台购买流量，欺诈平台在网站视频刷流量，网站定期向渠道方支付流量分成。右边是网店向流量欺诈平台购买流量，平台刷网店商品，提高人气，这是两种比较直接的形式。还有就是直播刷人气，与网店模式差不多，还有就是网站刷广告等其他形式。网站刷广告形式是做了一个网站，让广告联盟向网站投广告，站长向欺诈平台买流量，带动广告量，广告联盟就会支付费用。

给行业带来的伤害直接就是经济损失，第二个就是正常用户对产品的判定标准失衡，如电商你都不知道产品是人评价还是机器评价。然后就是企业信誉受损，如票房十亿但是电影很烂，



长此以往信誉就会受损。企业数据分析不准确，做商业分析时各种指标对不上。企业成本上升，如爱奇艺本来十台机器就能维护正常运维，但是一旦遇到刷量情况就无法承受，大量数据导致服务卡顿，因此服务器采购成本高。

数据的重要性



数据是一个企业的核心资产，是一种生产物料，并且表达了一系列的行为。数据可以衍生出各式各样的产品，能够帮助用户提升用户体验，如可以基于数据做一些推荐产品。数据还能帮企业做一些决策、预测。如果数据质量不过关就会导致分析不准确。

从工程技术来看，流量反作弊是一个技术问题，机器生成的数据，算异常流量。从业务方面，只要分析师认为数据表象特征不具备规律性，算异常流量。总的来说两者结合都算异常流量。但是有一个特点，从工程师的角度，这个事一旦做完就完了，如数据加密过程，你在广告、评论都能应用。



流量反作弊与数据清洗之间的关系

数据清洗更关注数据的完整性

- 字段是否存在
- 枚举值是否正确
- 该版本的投递是否注册并审核通过

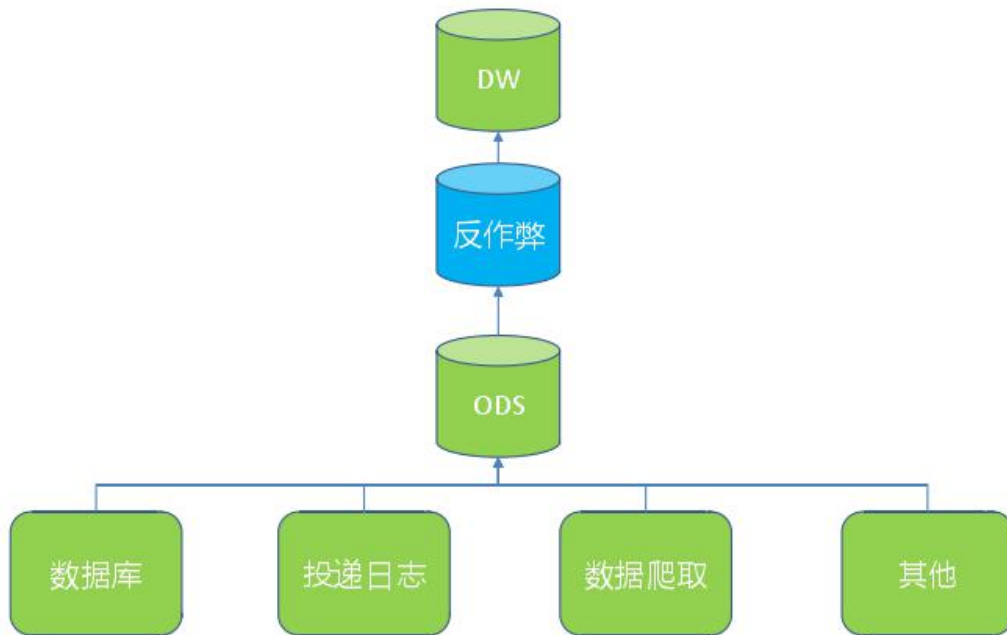
流量反作弊更关注数据的特征

- 指标是否正常
- 硬件信息是否正确
- 业务之间特征数据关联是否合理
- 是否满足预测模型

接下来讲一下流量反作弊与数据清洗之间的关系。数据清洗更加关注的是字段是否存在，枚举值是否正确，该版本的投递是否注册并审核通过。我们数据平台对数据投递是有一套规范，必须在投递平台注册并审核通过，才会认为投递是真实合法的。流量反作弊更关注数据的特征是：指标是否正常，硬件信息是否正确，业务之间特征数据关联是否合理，是否满足预测模型。



流量反作弊的位置



接下来说一下流量反作弊的位置，底下是各种数据源，然后将数据传到 ODS 上，ODS 直接对接反作弊的各种技术。反作弊做的越前越好，损伤越少，因为拿到的数据是一致的。这里有两种输出方式，一种就是以黑名单方式或者几种黑名单的方式；另一种就是直接给业务方表格。我们主要采用第二种方式，加入我们以黑名单形式，A 用户用来 A 黑名单，B 用户用来 B 黑名单，而 A 和 B 不一致。但是反作弊最好是能拿到所有数据，因此我们给出所有数据。我们以现有数据做反作弊，但是有些机密数据是拿不到的，机密数据是以黑名单的形式给用户。

流量反作弊的难点有：被动防守，事后分析，而且不断迭代发生。业务场景复杂，没有通用模型。第三个就是持续维护旧规则，不断增加新规则。因为作弊和反作弊是一个攻防过程，需要依据作弊技术不断更新规则。



流量反作弊的准备工作

-  掌握投递的日志或者数据库中的元数据含义
-  掌握主体业务的工作模式和场景
-  避免信息孤岛
-  了解作弊的目的什么
-  与业务部门良好的沟通，确认作弊的口径，并做好保密工作

做流量反作弊需要提前做一些准备工作。第一个就是掌握投递的日志或者数据库中的元数据含义，这个不可或缺，是所有工作的前提。第二个掌握主体业务的工作模式和场景，如视频相关，需要了解播放器相关的场景和工作模式，信息流是怎样的。第三个是避免信息孤岛，必须找到所有相关联的信息。第四个是了解作弊的目的什么，从目的入手比较好操作。第五个是与业务部门良好的沟通，确认作弊的口径，并做好保密工作。需要沟通确定作弊口径，如视频需要给业务方解释不以视频观看为目的的流量都算是作弊流量。



认定为技术问题的解决办法



认定为技术问题也是可以做的，一旦做完都可以用。具体方式有：做一些 IP 信誉机制，如果 IP 机制做好了，所有流量通过这个 IP 都是作弊流量。第二个安全画像，利用打分机制。第三个就是加密信息检测，运行起来所以机制都是一样的。然后是设备硬件信息检测，这是目前比较重要的机制，不管你是手机端的网页或是 APP 网页都能识别设备唯一 ID。

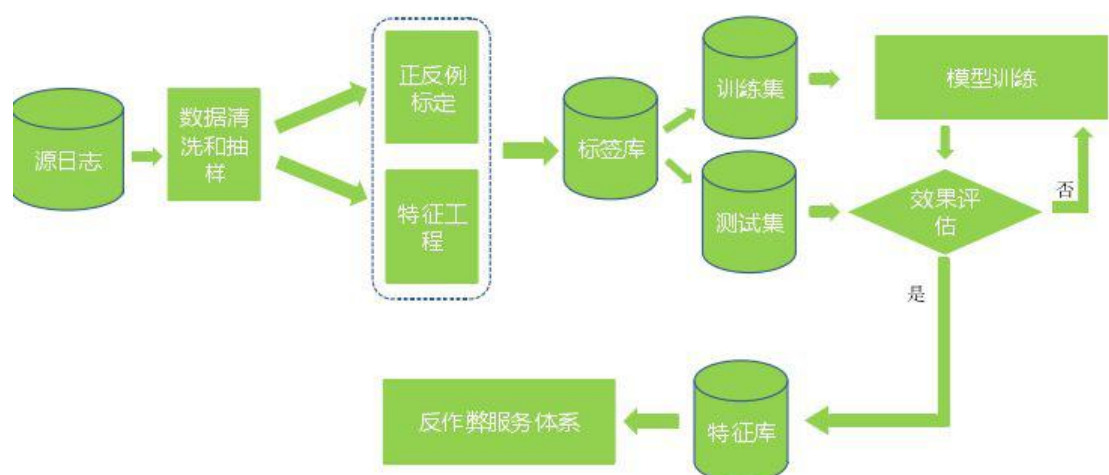
认定为业务问题的解决办法分为两个方面，一个基于规则统计，一种是基于机器学习。机器学习优点可以实现十分复杂的逻辑，但是需要关注模型的选择和效果，缺点解释性比较差。统计有一个很好地解释性，但是缺点要求统计模型的复杂程度限制在一定范围内。能找一些特征，但是选择特征有限。

认定为业务问题的解决思路先建立一个指标库，指标库要足够强大，第二个就是业务数据的上下文分析，业务发生时日志文件记录上面发生了什么还有下面要发生什么。第三个是行为



特征的分析，如用户先访问 A 再访问 B，再访问 C，但是它直接从 A 到 C 就不正常了。第四个是基于时间序列的分析，数据访问要符合一定的时间序列规则，如 A-B-C 不能 C-B-A。

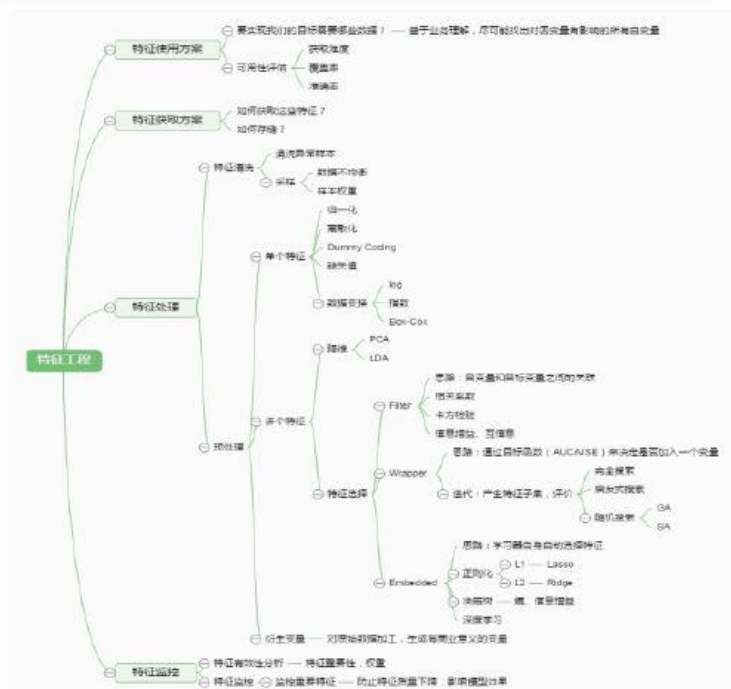
基于机器学习的反作弊



基于机器学习的反作弊思路是将数据从源日志中取出，进行数据清洗和抽样。然后做正反样例标定以及特征工程，然后将其放入标签库。分为测试集和训练集，然后进行模型训练，利用测试集进行模型评价，通过后构建特征库，进入反作弊服务体系。



认定为技术问题的解决办法



机器学习最重要的就是特征工程, 特征工程决定数据的天花板。如果特征工程好, 模型一般也可以很好, 但是如果特征工程不好模型再好一定不能做好。上图介绍了特征工程如何做、使用方案、专业评估, 然后如何获取特征, 这些特征如何称呼, 还有特征预处理、分级以及做一些降维。然后就是模型训练, 主要使用 LR、RF、GBDT、XGBoost。

效果评估就是敏感数据更关注 TP, 因此精确率必须要好, 否则就无法使用。还有一般数据, 如 F1, ROC, AUC。这些指标低点并不影响后续分析。业务不断维护旧规则创建新规则, 对旧规则而言对作弊数据打标签, 创建特征标签验证, 基于作弊数据和标签进行分析, 最后进行验证。比如今天做了一些反作弊, 有两条规则都对数据进行验证, 交集越来越大, 最后一个包围另一个, 那么被包围的规则就没有用了。

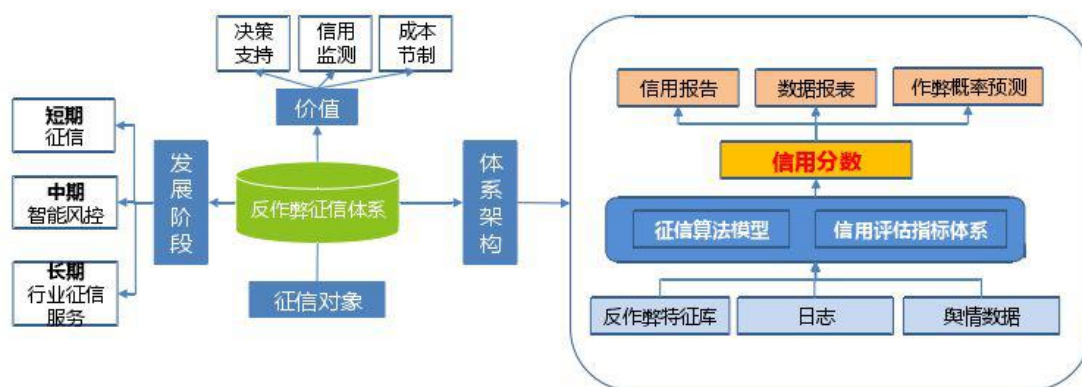


流量反作弊的监控



还有重要的一点是监控, 如果被业务方反应已经很被动, 需要在业务方发现之前解决。需要做一个监控提醒, 首先定义定义监控指标 (pv, uv, 评论数等), 业务主体基础数据的监控, 端整体数据的监控。然后分主题监控, 如细粒度的分主题相关数据的监控, 还有一点是监控的时效性, 需要实时监控, 准实时监控。

征信系统



从系统应用的角度看反作弊体系，反作弊离不开征信，首先确定征信对象，对于我们就是渠道征信。还有发展阶段，短期做什么，中期做什么，长期做什么。其价值就是做一些决策、信用监测以及成本节制。体系结构构件作弊与反作弊特征库，建立一些征信模型，建立信用评价体系然后做一些预测。

流量反作弊的服务应用第一个就是数据报表，具体有：内容流量&反作弊总体分析，单个内容流量&作弊情况查询，攻防效果分析，征信总体分析，单个征信对象信用报告查询智能分析。第二个智能分析，就是负责作弊识别-扣量-减付业务流程自动化，作弊概率预测，异常预警，策略分析。第三个就是接口服务，反作弊结果以接口形式服务于各个业务。

未来展望在技术方面引入深度学习方法，蚕蛹模型融合技术，产品方面加快信用体系建设，形成产业联盟。



DataFun视频回看年卡

内容简介

近3年累计大数据，算法实战案例300+

每年至少新增案例分享200+

专属交流群，配套PPT下载

可在公众号发布内推信息，助力招聘

其他权益

推荐算法

精排、粗排、召回、排序模型、推荐系统架构

广告算法

点击率预估、定向广告、广告冷启动、展示广告

搜索算法

Query理解、语义计算、搜索排序/召回

知识图谱

大规模知识图谱构建、电商知识图谱、金融知识图谱

用户画像

用户画像方法论、标签生产引擎、用户生命周期管理

大数据

数据架构、数据仓库、数据治理、数据应用、OLAP

~~原价199元/年~~
年度合集专属价**133元/年**





高级私人定制服务

职业 “问” 道

一流公司高级专家为你答疑解惑

想找心仪的工作？想升职加薪？

专家为你构建能力框架

想解决项目中的难题？想顺利推进工作？

专家为你制定落地方案

想让公司的营收进行指数级增长？

专家为你提供发展之道



扫码了解服务详情

