中山大學
SUN YAT-SEN UNIVERSITY

不平衡数据处理方法

任课老师：郭小波（副教授）

邮箱: mc03gxb@126.com

电话：15013293033

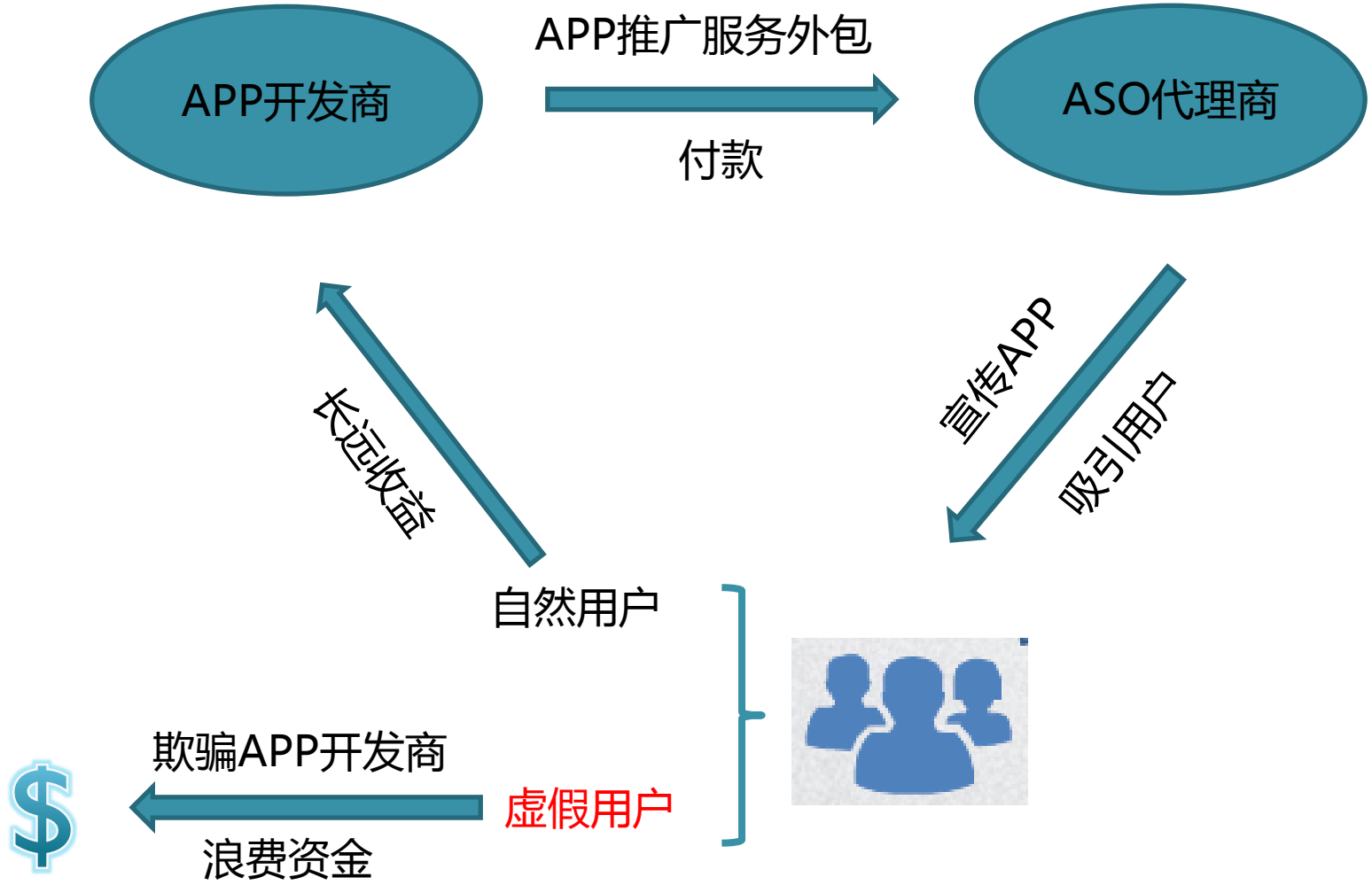
目录

案例 手机APP的虚假用户识别

- 在这个APP泛滥的年代，很多用户会认准下载量或者注册用户最多的APP。想在众多APP中脱颖而出，应用开发者面临着激烈的竞争，由此应运而生的，则是一个新兴的行业：应用市场优化(ASO)，其主要任务推广APP。但是承担ASO优化的代理商有时会通过使用虚假用户造成推广假象，从APP开发商获得高额利润。如何识别虚假用户呢？

中山大學 SUN YAT-SEN UNIVERSITY

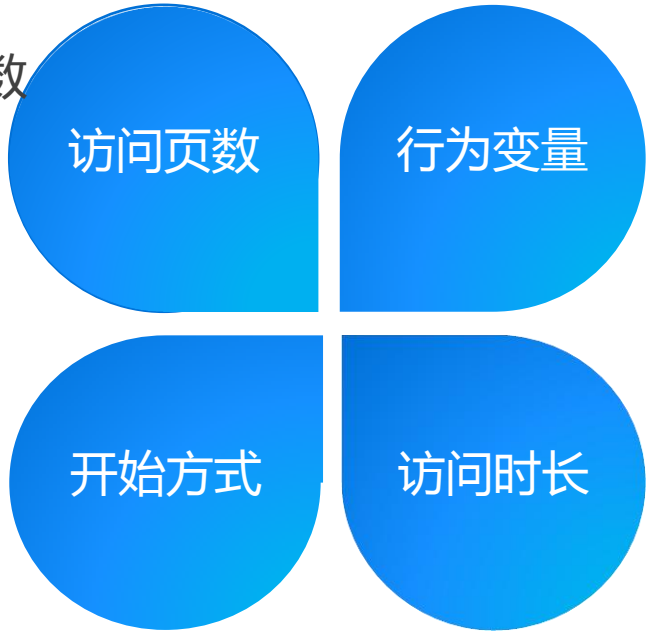
研究目的——识别虚假用户



中山大學 SUN YAT-SEN UNIVERSITY

- 本案例使用某APP市场咨询公司的用户数据——包括了刻画用户在APP上操作情况的用户行为变量、用户访问网页数、用户开始使用APP的方式以及用户访问的时长等，加入人为噪音，并对其进行分析。

每个用户访问网页的数目，取值范围1~20.



刻画了用户打开、关闭某个网页的操作，体现了用户在APP的操作方式。

每个用户打开应用的方式，1表示直接开始，0表示从后台开始。

每个用户访问网页的时长，取值越大，说明时间越久。

中山大學 SUN YAT-SEN UNIVERSITY

用户情况

开始方式	自然用户	虚假用户	合计
合计	50	5615	5665

首先，在自然用户占比0.09%的数据中，按以下流程建模并评估

划分训练测试集 → 逻辑回归 → 交叉验证 → 模型评价

此时，在自然用户占比0.09%的数据中，结果如下：

Model 1	test_AUC	test_ACC
Logistic model	0.744	0.993

例：分类后的混淆矩阵

	Predicted	
Actual	1	0
1	1871	1
0	13	4

ACC=（1871+4）/1889=0.993，虽然ACC很高，但是17例0样本里只有4例被正确分类，FPR=13/17=76%。

测试集中AUC= 0.744，较低。

传统的算法通常是偏向数量占优的类，因为传统的算法常可归为优化某一特定的损失函问题：

$$\theta^* = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i, \theta)),$$

而该损失函数常与准确率密切相关。因为学习算法在寻找最优的模型去最小化与y的距离，本质上也是在寻找最优的模型，使得准确率最大化。因此，学习算法会偏向把样本判为样本量大的类别以获得更高的准确率。

- 下面从损失函数的角度，分析数据不平衡给逻辑回归带来的问题。

令$f(x_i,\beta)=\frac{\exp(x_i^T\beta)}{1+\exp(x_i^T\beta)}$，逻辑回归要最小化下面目标函数：

$$L(\beta)=-[\sum_{i=1}^{n} y_i \log(f(x_i,\beta))+(1-y_i)\log(1-f(x_i,\beta))]$$

$$=\sum_{i:y_i=1} -\log(f(x_i,\beta))+\sum_{i:y_i=0} -\log(1-f(x_i,\beta)),$$

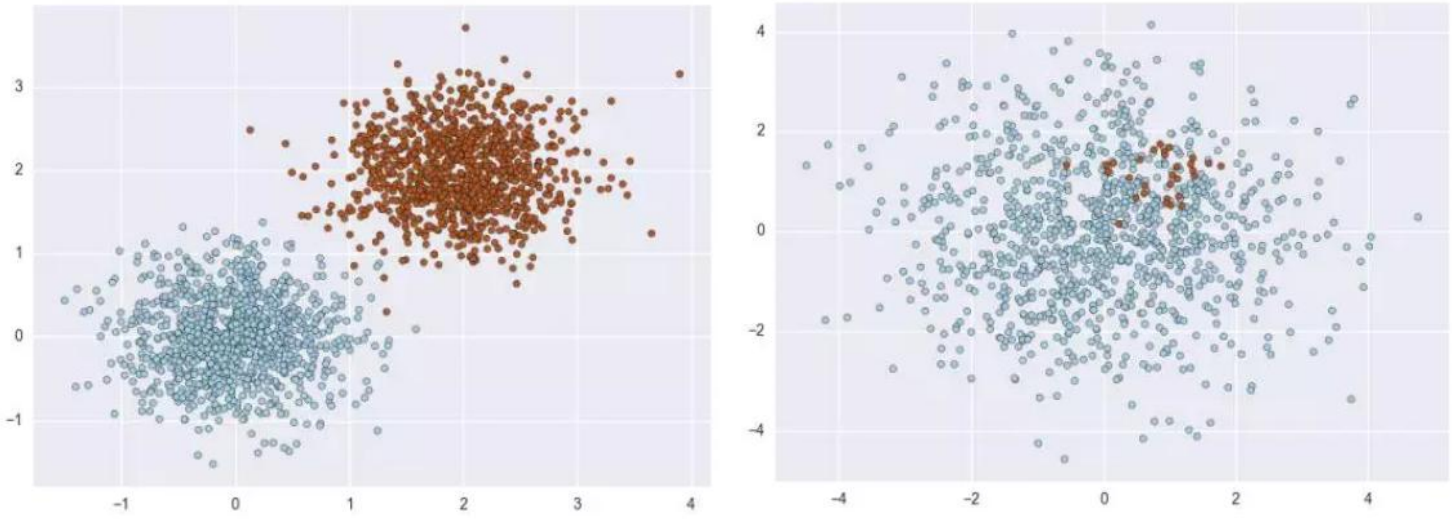
对应的损失函数为：

$$L(y_i,\ f(x_i,\beta))=\begin{cases} -\log(f(x_i,\beta)) \text{，如果} y_i=1, \\ -\log(1-f(x_i,\beta)), \text{如果} y_i=0。 \end{cases}$$

从损失函数，我们可以看出：目标函数本质上是在优化准确率。当数据存在不平衡时，例如$y_i=1$占大多数，那么法会主要去优化$\sum_{i:y_i=1} -\log(f(x_i,\beta))$这一部分。

此时，$f(x_i,\beta)$越接近1，损失越小。

- 在教学中，数据通常是净化过的，这样老师才能够把注意力集中在教授特定算法或技巧上，而不被其它问题干扰。一般情况下，我们遇到的样本类似下方右图。然而，当我们开始面对真实的、未加工过的数据时，会发现，这些数据要嘈杂且不平衡得多。真实数据的散点图看起来更像是下方右图。



图：其中的点代表样本、点的不同颜色（或形状）代表类

最主要的问题是这些类是不平衡的：蓝点的数量远超红点。

对于不平衡类的研究通常认为「不平衡」意味着少数类只占 10% 到 20% 。而在现实中，数据库甚至能够比上面的例子更加不平衡。

比如：

1. 每年，约 2% 的信用卡账户是伪造的
2. 美国的 HIV 感染率约为 0.4%
3. 每年，硬盘驱动器故障的发生率约为 1%
4. 在线广告的转化率在 10^-3 到 10^-6 的范围区间内
5. 工厂的产品缺陷率一般在 0.1% 左右

以上的许多领域都是不平衡的，在这种情况中，机器学习分类器要从庞大的负面（不相关）样本中，寻找少量的正面（相关的、值得注意的）样本。

- **数据层面的方法**

  - 随机欠采样（Random Under-Sampling）
  - 随机过采样（Random Over-Sampling）
  - 信息性过采样：合成少数类过采样技术（SMOTE）

- **算法集成技术（Algorithmic Ensemble Techniques）**

  - 基于 Boosting 的方法

    a)自适应 boosting--Ada Boost

    b)Gradient Boosting

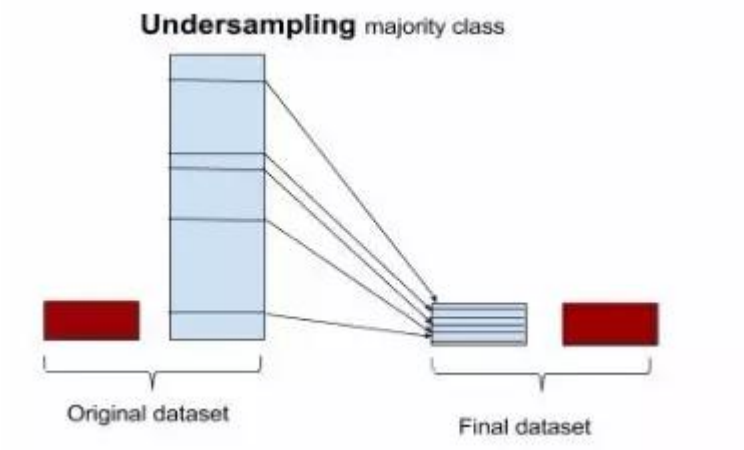
    c)XGBoost（Extreme Gradient Boosting）

- **数据层面的方法+算法集成技术**

  上述两种层面的结合使用，如SMOTE+ Gradient Boosting

□ 随机欠采样（Random Under-Sampling）

通过随机地消除占多数的类的样本来平衡类分布，直到多数类和少数类的实例实现平衡。



实例操作

不重复地从虚假用户中取 50例样本，并将其与自然用户相结合，组成新的平衡数据集。

- 结果

Model 2	test_AUC	test_ACC
Under-sampling + Logistic model	0.878	0.949

优点

- 提升运行时间
- 当训练数据集很大时，可以通过减少样本数量来解决存储问题

缺点

- 丢弃对构建规则分类器很重要的有价值的潜在信息
- 被随机欠采样选取的样本可能具有偏差。它不能准确代表大多数。从而在实际的测试数据集上得到不精确的结果

优点

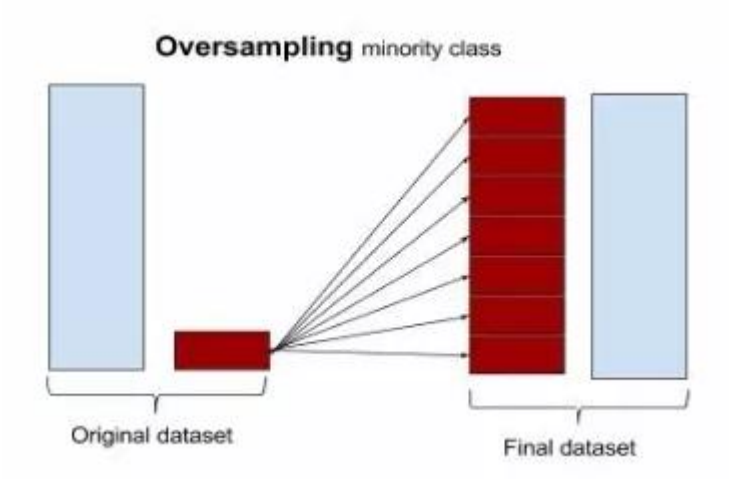
- 提升运行时间
- 当训练数据集很大时，可以通过减少样本数量来解决存储问题

缺点

- 丢弃对构建规则分类器很重要的有价值的潜在信息
- 被随机欠采样选取的样本可能具有偏差。它不能准确代表大多数。从而在实际的测试数据集上得到不精确的结果

□ 随机过采样（Random Over-Sampling）

通过随机复制少数类来增加其中的实例数量，从而可增加样本中少数类的代表性。



实例操作

复制每个自然用户20 次。

结果

Model 3	test_AUC	test_ACC
Over-sampling + Logistic model	0.875	0.936

优点

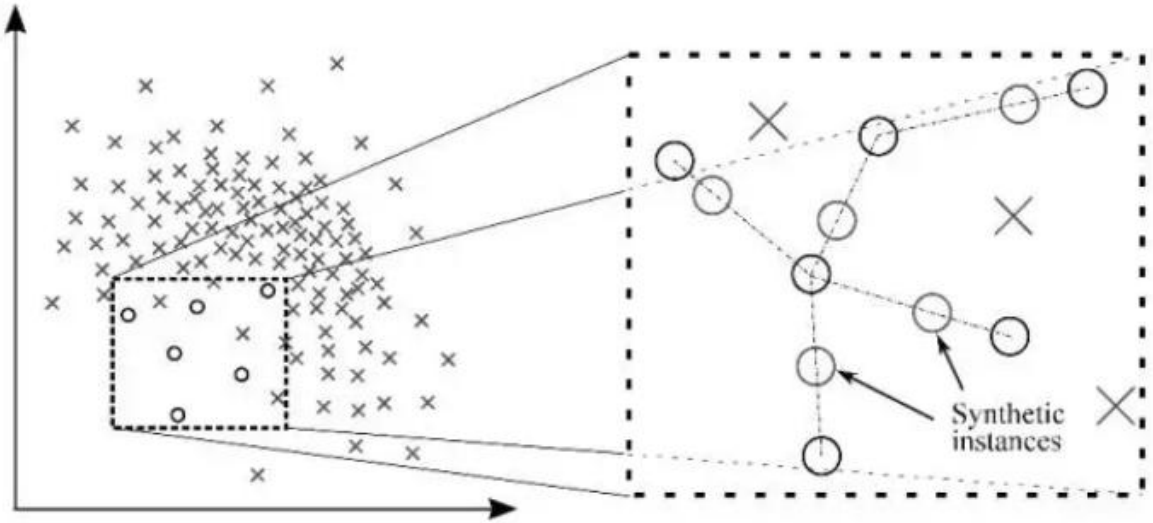
- 与欠采样不同，这种方法不会带来信息损失。

缺点

- 由于复制少数类事件，它加大了过拟合的可能性。

- **口信息性过采样：合成少数类过采样技术（SMOTE， synthetic minority over-sampling technique ）**

其思想是通过在已有的样本间插值来创造新的少数类样本。从少数类中把一个数据子集作为一个实例取走，接着创建相似的新合成的实例。这些合成的实例接着被添加进原来的数据集。新数据集被用作样本以训练分类模型。



结果

Model 4	test_AUC	test_ACC
SMOTE + Logistic model	0.900	0.951

优点

- 通过随机采样生成的合成样本而非实例的副本，可以缓解过拟合的问题。
- 不会损失有价值信息。

缺点

- 当生成合成性实例时，SMOTE 并不会把来自其他类的相邻实例考虑进来。这导致了类重叠的增加，并会引入额外的噪音。
- 因为它是在稀有的样本之间插值，所以它只能生成可用样本范围内的样本——永远不会生成例外的样本。形式上，SMOTE 只能填入已有少数类样本的凸包（Convex Hull）中，但不能创造在少数类样本域之外的新样本。

- 以上方法均是从数据层面解决不平衡带来的问题，下面使用的Gradient Boosting方法是从算法层面解决这一问题。

结果

Model 5	test_AUC	test_ACC
Gradient Boosting model	0.920	0.973

为了获得更好的结果，可以使用 Gradeint boosting 的同时也使用 SMOTE 合成采样技术。

Model 6	test_AUC	test_ACC
SMOTE+ Gradient Boosting model	0.954	0.995

模型对比

All Models	test_AUC	test_ACC
Logistic model	0.744	0.993
Under-sampling + Logistic model	0.878	0.949
Over-sampling + Logistic model	0.875	0.936
SMOTE + Logistic model	0.900	0.951
Gradient Boosting model	0.920	0.973
SMOTE+ Gradient Boosting model	0.954	0.995

- 遇到不平衡数据集时，没有改善预测模型准确性的一站式解决方案。可能需要尝试多个办法来搞清楚最适合数据集的采样技术。在绝大多数情况下，如 SMOTE 的合成技术会比传统过采样或欠采样的办法要好。

- 为了获得更好的结果，可以在使用诸如 Gradeint boosting 和 XGBoost 的同时也使用 SMOTE 等合成采样技术。