电子商务数据分析

第4章 电子商务欺诈与反欺诈

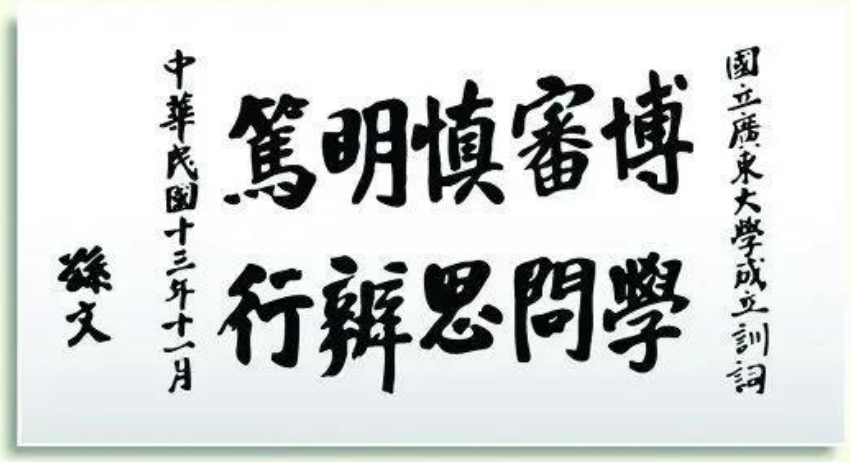
朱桂祥 (9120201070@nufe.edu.cn)
南京财经大学信息工程学院
江苏省电子商务重点实验室
电子商务信息处理国家级国际联合研究中心
电子商务交易技术国家地方联合工程实验室

https://github.com/zgx881205/Analysis-of-E-commerce-Data

南京财经大学 NANJING UNIVERSITY OF FINANCE & ECONOMICS

校训是什么?

中山大学校训:
博学 审问 慎思 明辨 笃行



http://tv.cctv.com/2014/08/13/VIDE1407888842216432.shtml

电子商务欺诈与反欺诈

4.1 电子商务欺诈

定义：
电子商务欺诈 是以粉饰，虚构或是扭曲商品信息的途径，
提高业绩，诱导消费者购买，提高商品排名等变相获取流量。

手段：
刷单，好评返现，修改评价。

电子商务欺诈与反欺诈

4.1 电子商务欺诈

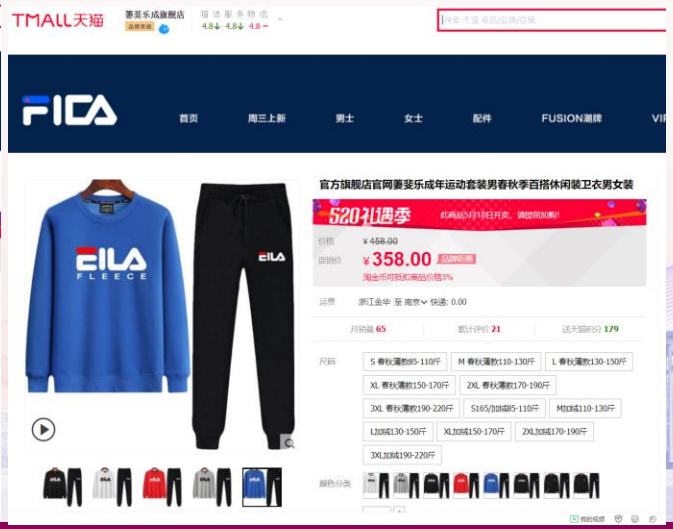
电子商务欺诈的危害：

危害：电子商务欺诈并非直接从消费者身上获得利益，[illegible]蔽性，严重影响了商家之间的公平竞争，干扰了消费者的判断，其危害性不容忽视。

(1) 对社会财产造成危害，使得资源不能有效配置。

(2) 危害了电子商务的信用，使人们对平台的信心受到打击。

(3) 隐蔽性多次进行，其实际结果比普通经济犯罪的危害更大。



电子商务欺诈与反欺诈

4.1 电子商务欺诈

刷单诈骗：

5月19日，家住井陉县城周边某村的范女士在一个微信群里看到一个兼职刷单的二维码。怀着好奇的心理，范女士打开二维码，下载了一款APP软件。按照APP上客服的指引，范女士开始做收藏店铺的“任务”，每次做完任务都会收到少量的“佣金”。

尝到甜头的范女士，接下来开始做一种名叫“集量变现”的任务。所谓“集量变现”就是向客服提供的银行账户转账，然后根据转账数额做不同的“任务”，赚取相应的“佣金”。第一次，范女士转账100元，获取“佣金”30元。第二次，她为了赚取更多的“佣金”转账10000元。可是，等她做完“任务”之后，却不能提现。经与客服联系，客服让她缴纳4000元“保证金”。缴纳“保证金”之后，客服说只能提现10000元，“保证金”暂时不能提取。10000元提现成功后，范女士对客服“深信不疑”。接下来，范女士在客服的提示下，连续做任务，连续不能提现，连续缴纳“保证金”。5月19日至22日四天时间，她先后缴纳“保证金”10次，总计20余万元。每次缴纳“保证金”时，她都十分相信客服的提示，幻想只要自己按照客服要求去做就能够提现成功。直到家人劝她到公安机关报警时，她还不认为自己被骗。

据负责接警的民警介绍，范女士4天10次缴纳“保证金”的主要原因是她不甘心“及时止损”，为了拿回自己已经缴纳的“保证金”，一次又一次对客服“心存幻想”。

[1]警惕！“兼职刷单”做任务 4天被骗20余万元.
http://www.he.xinhuanet.com/xinwen/2022-05/26/c_1128684784.htm

电子商务欺诈与反欺诈

4.1 电子商务欺诈

刷单产业：
有很多的网络店家雇人刷单，而所付出的成本，也的确是每单5-10元。但是这个刷单行业，也已经形成了产业化、规模化。

刷单属于伪造信用
伪造信用的目的，是为了让新来的顾客通过浏览交易记录和买家评论，判定这家店铺商品的好坏，大量的购买记录和铺天盖地的好评会让不熟悉这家店铺的新顾客第一时间认定这家卖的是好货，从而产生真实的购买。

电子商务欺诈与反欺诈

4.1 电子商务欺诈

好评返现：

2018年12月10日，中国网络诚信大会正式发布了《中国电子商务诚信发展报告》，报告建议“禁止电子商务平台采取‘好评返现’及其类似行为”。报告认为，“好评返现”实质上是商户通过货币返现对消费者进行交易贿赂，诱导消费者对产品和卖方的交易行为作出非客观评价。

淘宝的态度：有好评返现的图片进行宣传，商品一律下架。

电子商务欺诈与反欺诈

4.1 电子商务欺诈

电子商务欺诈形成的原因：

(1)信息不完备下的消费者决策。

商品的直接信息：包括商品相关的展示信息。

商品的间接信息：包括用户和第三方的评价。

(2) 信息不完备使得消费者

对间接信息过度依赖。

(3) 电商平台以间接信息为依据

进行流量分配。销量，好评度，客单价，买家在商品的停留时间，浏览和购买的

化率。

电子商务欺诈与反欺诈

4.2 电子商务反欺诈

目标：

(1) 对电子商务欺诈的一种识别服务。

(2)使用评论数据、用户行为数据、人口统计数据对电子商务中的恶意差评，刷单行为进行识别。

具体内容：

(1)电子商务推荐系统的恶意用户检测。

(2)电子商务网站恶意评论用户检测。

(3)社会化商务恶意用户检测。

电子商务欺诈与反欺诈

4.2 电子商务反欺诈

(1)托攻击：伪造用户，使得该虚假用户成为很多正常用户的近邻。
使得推荐系统频繁推荐自己的商品，而减少或是不推荐竞争对手的商品。
(2)托攻击的分类：
推攻击(恶意提高排名)
核攻击(恶意降低排名)
扰乱攻击(恶意歪曲排名)
(3)托攻击的检测算法
托攻击的本质是一个分类问题

分类问题和回归问题的本质区别是？

分类问题：分类问题的输出是离散型变量(如: 0，1，2)，是一种定性输出。(预测明天天气是阴、晴还是雨)

回归问题：回归问题的输出是连续型变量，是一种定量输出。(预测明天的温度是多少度)。

电子商务欺诈与反欺诈

4.2 电子商务反欺诈

(1)托攻击的检测算法：
基于监督学习的托攻击检测算法：
把数据分为训练集和测试集。
训练集中包含正常用户和托攻击用户，这两种用户都已经标注。
通过训练集，训练托攻击的算法模型。
基于常用的分类器算法(比如决策树，随机森林等)。

优点：训练集中已包括的用户能够较好的检测。
缺点：训练集中没有包含特征的用户不能正确的检测。

基于无监督学习的托攻击检测算法：

托攻击用户之间的皮尔逊相相似度很高(>0.9)
第一个无监督学习框架PCASelectUsers。
Lee等人提出的检测器：
先聚类，再根据Group RDMA(GRDMA)来判断是否为托。

电子商务欺诈与反欺诈

4.3 电子商务网站恶意评论

用户检测

职业差评师，顾名思义，就是靠给别人差评生活的人。这是一种由淘宝网催生的新兴职业，淘宝上有很多恶意买家做起职业差评师，专门以给网店差评为手段索要网店钱财，甚至还出现多人合作的“团伙作案”。而淘宝网开通拉黑买家的功能，目的就是为了解决职业差评师对卖家权益的侵害，防止给淘宝卖家造成巨大的伤害。

电子商务欺诈与反欺诈

4.3 电子商务恶意评论检测

案例分析[1]

上海某酒店店主告诉记者，上个月该酒店在大众点评上的页面出现了不少差评，随后便接到一名自称工作人员的电话，声称可以删除差评甚至提高评价星级。见面时，对方露出真相，承认是“专业”的点评炒作机构，称每月花1500元即可享受到他们提供的好评服务。遭拒后，此人伙同公司其他员工，开始注册小号，对商户进行差评攻击。最终，该名嫌疑人被警方以敲诈勒索罪起诉，其公司也由于该事件被查封并勒令停业。

[1] http://media.people.com.cn/n/2014/0624/c14677-25192478.html

电子商务欺诈与反欺诈

4.3 电子商务恶意评论检测

记者了解到，这种强卖“操纵点评”服务的事件，在沪上并非首次出现。2012年3月，新泾工商所就曾查处一起在第三方平台买卖虚假点评案：一家非法网络公关机构通过在点评网站上为合作商户撰写虚假好评谋取暴利，雇佣近10名“网络水军”，开设几百个马甲账号，每人每天负责“灌水”和炒作点评。

遗憾地是，大多数商户十分看重所谓用户的好评和获评的星级，在炒作机构勒索时选择了妥协。记者询问大众点评网上的20家商户，他们中有一半曾遭遇过勒索，多达6成的商户最终选择花钱了事，费用从几十元到上万元不等。原因一是存在侥幸心理，认为好评多可有助于店铺的口碑；二是担心这些职业差评师会报复。不过，在遇到勒索者抬高要价时，几乎所有商户都会选择拒绝。

对于第三方炒作机构和职业差评师的行为，北京市盛峰律师事务所主任律师于国富指出，这是一种典型的敲诈勒索行为。根据《刑法》，敲诈勒索公私财物，数额特别巨大或有其他特别严重情节的，处10年以上有期徒刑，并处罚金。

[1] http://media.people.com.cn/n/2014/0624/c14677-25192478.html

电子商务欺诈与反欺诈

4.3 电子商务恶意评论检测

用户检测 职业差评师的攻击目标[2]:

1、信誉不是很高的，一般在3钻（包括）的店铺。
2、卖家没有或者很少有中差评。
3、考察卖家卖的是什么产品符合的产品（具备的条件是1、价格不高 2、容易损坏）
4、地域性的选择、职业差评师一般会选择地方比较远、邮费比较高的地方进行购买。
5、产品的运输有局限性的产品（差评师会备注必须走xx快递，假如你没有走这个快递，差评师会以没有收到这个货物来给你差评

职业差评师的特点:

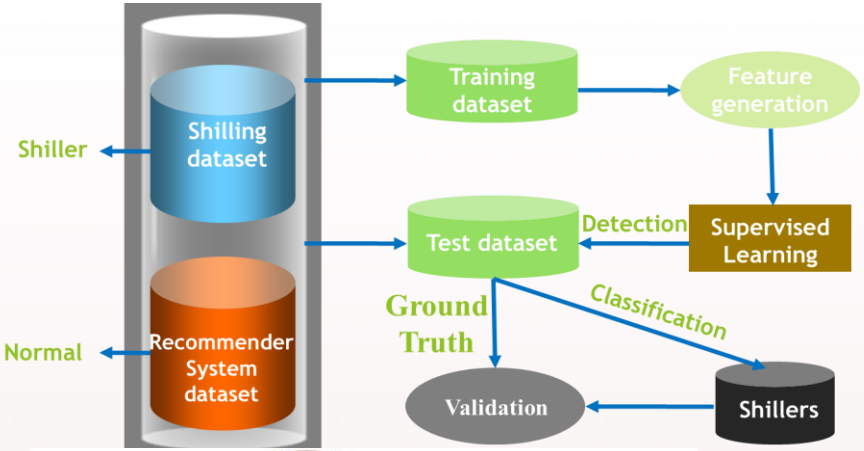
1、差评师的信誉不会很高的。
2、差评师一般不会计较价格方面的事情，但所买的东西一般容易再次卖出去。
3、差评师一般收货之后不会和你在即时通讯上联系，而通过差评后，由卖家打电话主动联系，不留下相关的证据。

[2] https://www.zhihu.com/question/20278316/answer/569976934

电子商务欺诈与反欺诈

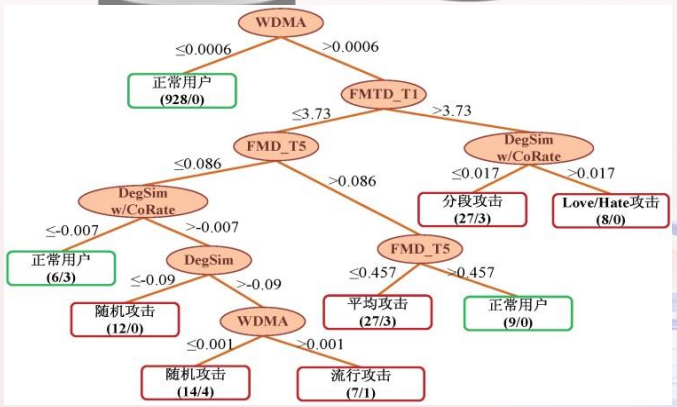
4.4 电子商务恶意用户检测

监督检测器将恶意用户检测看成二分类问题，并基于评分或评论特征来检测



朴素贝叶斯检测器

一个基于C4.5的推荐系统恶意用户检测示例



算法 4.2 基于朴素贝叶斯分类的恶意用户检测算法

输入：由少数正常用户和注入的恶意用户组成的训练数据集，以及测试数据集

输出：每个测试数据集中 u 的类别

1. 选取部分正常用户，并注入少部分恶意用户组成训练数据集
2. 计算训练和测试数据每个用户 u 的 n 个检测指标值，记为 $\{x_1, x_2, \ldots, x_n\}$
3. 利用 4.3.2 节中特征选择算法选择 m 个检测指标
4. 根据式(4.13)计算 $p(x_{ik}|c_j)$，μ_{jl} 和 σ_{jl} 分别为 C_j 类训练数据集第 l 个指标的均值和标准差：

$$p(x_{il}|c_j) = g(x_k, \mu_{jl}, \sigma_{jl}^2), \text{其中} g(x,\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (4.13)$$

5. 根据公式(4.14)计算拖攻击属于正常或恶意用户的值：

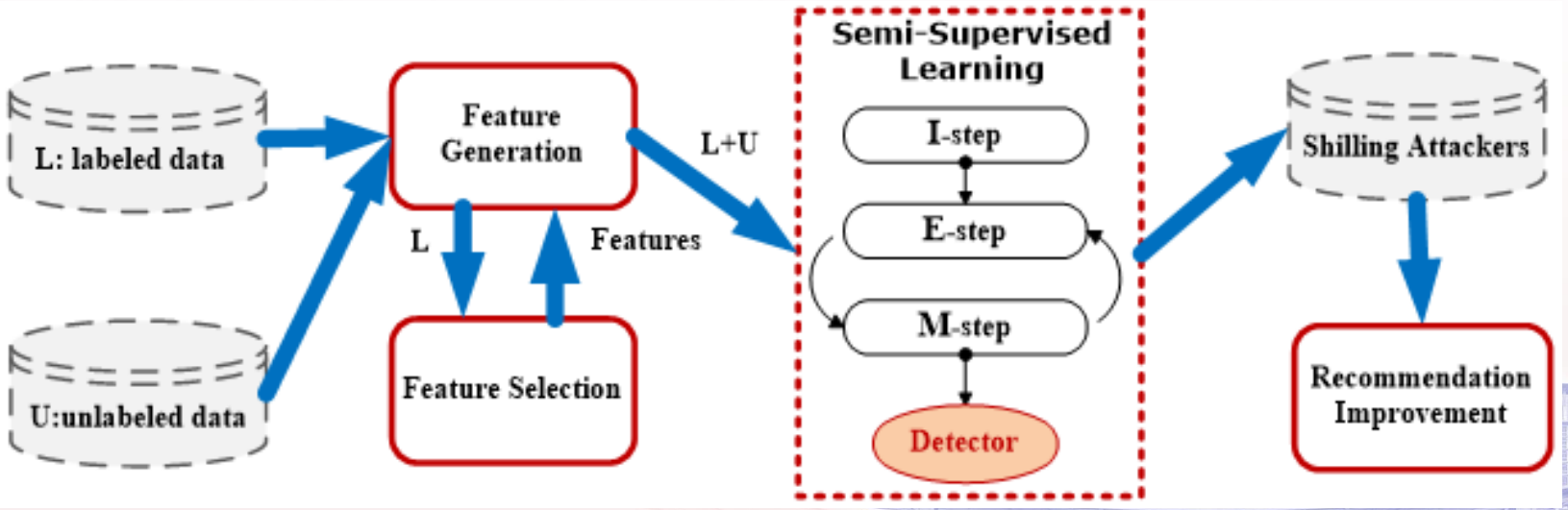
$$\Omega = \ln\frac{p(S|u)}{p(N|u)} = \ln\frac{P(S)}{P(N)} + \sum_{k=1}^{m}\ln\frac{p(x_k|S)}{p(x_k|S)} \quad (4.14)$$

电子商务欺诈与反欺诈

4.4 电子商务恶意用户检测

基于半监督学习的托攻击检测算法：
无标记数据往往容易获取，标记数据的获取往往费事费力。

怎样把有标记的数据和没有标记的数据结合起来，进行半监督的机器学习算法？

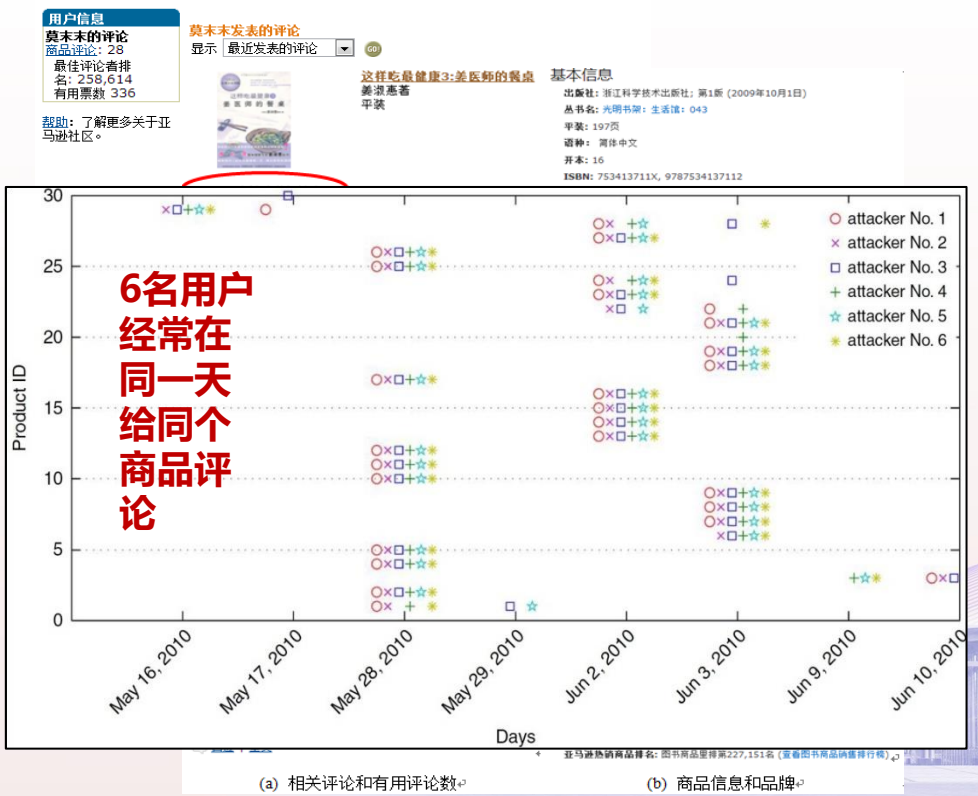


[1] Wu Z , Wu J , Jie C , et al. HySAD: a semi-supervised hybrid shilling attack detector for trustworthy product recommendation[C]// ACM SIGKDD international conference on knowledge discovery and dataMining. ACM, 2012.

电子商务欺诈与反欺诈

4.4 电子商务恶意用户检测

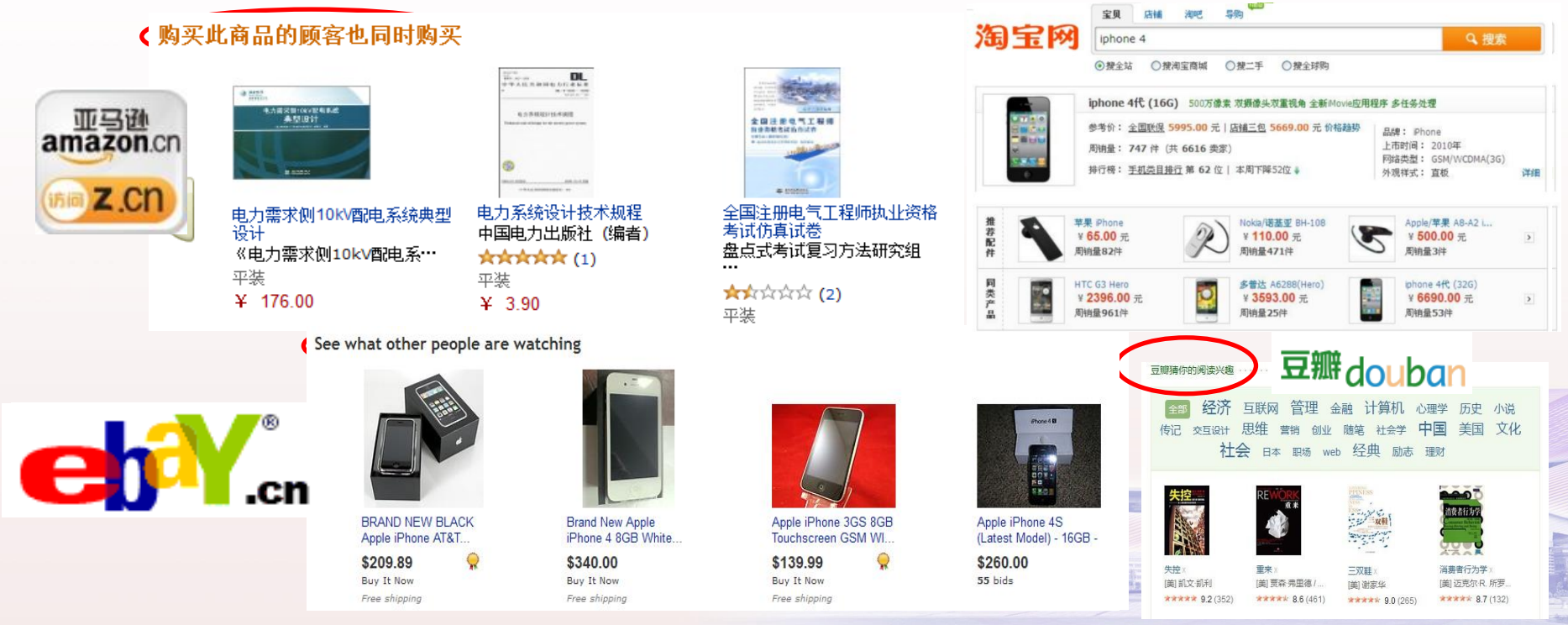
社会化商务恶意用户的目标通常为获得经济利益或造成网络影响，其行为模式必然与正常用户相比具有很大的差异性。



电子商务欺诈与反欺诈

4.4 电子商务恶意用户检测

试图从推荐系统角度研究恶意用户行为和检测方法



电子商务欺诈与反欺诈

4.4 电子商务恶意用户检测

-

The Process of User-CF

Scan all users, find similar interest users, use their scores to predict the scores of user u.

1. Establish a $m \times n$ matrix
2. Search neighbors using similarity algorithm
3. Recommendation generation

推

-

号，

商品评分	1	2	3	4	5	6	7	8	与Alice的相似度
Alice	5	3	3	4	2	1			\
User 1	3	4	2	3	4	5	1	3	-0.67
Use 2	4	3	1	2	4	2	4	1	0.23
User 3	4	2	1	3	4	1	5	5	0.52

Alice和User 3的相似度最高，而User 3对商品7、8打最高分，所建将商品7、8推荐给Alice.

电子商务欺诈与反欺诈

4.4 电子商务恶意用户检测

恶意用户攻击(托攻击, Shilling Attacks),托攻击者将伪造的用户模型注入推荐系统，这些伪造的用户模型通常能成为很多正常用户的近邻，从而扭曲系统的推荐结果

正常用户

目标项目

	Item1	Item2	Item3	Item4	Item5	Item6	Correlation with Alice
Alice	5	2	3	3		?	
User1	2		4		4	1	-1.00
User2	3	1	3		1	2	0.76
User3	4	2	3	1		1	0.72
User4	3	3	2	1	3	1	0.21
User5		3		1	2		-1.00
User6	4	3		3	3	2	0.94
User7		5		1	5	1	-1.00
Attack1	**5**		**3**		**2**	**5**	**1.00**
Attack2	**5**	**1**	**4**		**2**	**5**	**0.89**
Attack3	**5**	**2**	**2**	**2**		**5**	**0.93**

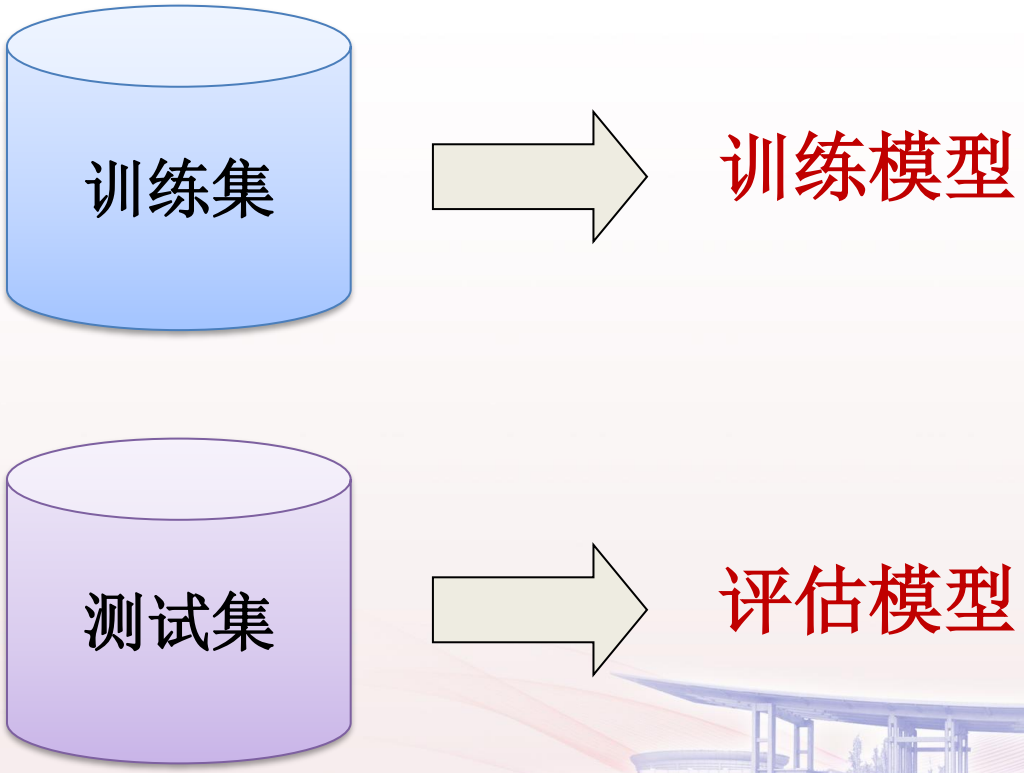
正常情况下，项目6未进入Alice的推荐列表

注入3个托攻击用户后，目标项目6推荐给了Alice

电子商务欺诈与反欺诈

4.4 电子商务恶意用户检测

训练集和测试集



电子商务欺诈与反欺诈

4.4 电子商务恶意用户检测

分类器的评价指标

- 准确率: 分类器正确分类的实例占测试集的百分比
  - 用Acc(M)表示，M表示模型，Acc为Accurcy的缩写
- 误差率： 1 - Acc(M)

测试集：

分类结果：

准确率 = 8 / 10 = 80%

误差率 = 2 / 10 = 20%

仅有准确率难以衡量一个分类器的优劣：设0类样本9990个、1类样本10个；如果分类器将所有样本分到0类，准确性为9990/10000=99.9%！

电子商务欺诈与反欺诈

4.3 电子商务恶意用户检测

分类器的评价指标

混淆矩阵(confusion matrix)

		Predicted Class	
		Class=Yes	Class=No
Actual Class	Class=Yes	TP	FN
	Class=No	FP	TN

给定两类，正例表示感兴趣的主类的实例，负例表示另一类的实例

P (Positive)和N (Negative) 表示模型的判断结果；

T (True)和F (False) 表示模型的判断结果是否正确；

1. 真正(TP, True Positives)：分类器正确标记的正例；
2. 假正(FP, False Positives)：分类器错误标记的负例；
3. 真负(TN, True Negatives)：分类器正确标记的负例；
4. 假负(FN, False Negatives)：分类器错误标记的正例；

4.3 电子商务恶意用户检测

5.分类器的评价指标

7000个恶意用户中，
有6954个被正确分类为“恶意用户”，
有46个被错误分类为“非恶意用户”

混淆矩阵

	Predicted Class		合计
Actual Class	Class=恶意用户	Class=非恶意用户	
Class=恶意用户	TP=6954	FN=46	7000
Class=非恶意用户	FP=412	TN=2588	3000
合计	7366	2634	10000

Yes 类的评价

$$\text{Precision (P)} = \frac{TP}{TP+FP} = 94.4\%$$

$$\text{Recall (R)} = \frac{TP}{TP+FN} = 99.3\%$$

$$\text{F-measure (F)} = \frac{2PR}{P+R} = 96.8\%$$

具有高准确率的分类器，
非对角线的项应接近于零

The harmonic mean of *R* and *P*!

电子商务欺诈与反欺诈

4.3 电子商务恶意用户检测

5.分类器的评价指标

自己编写函数:

```
def calculate_TP(y_true, y_pred):
    TP = 0
    for i, j in zip(y_true, y_pred):
        if i == j == 1:
            TP += 1
    return TP

def calculate_TN(y_true, y_pred):
    TN = 0
    for i, j in zip(y_true, y_pred):
        if i == j == 0:
            TN += 1
    return TN

def calculate_FP(y_true, y_pred):
    FP = 0
    for i, j in zip(y_true, y_pred):
        if i == 0 and j == 1:
            FP += 1
    return FP
```

```
def calculate_FN(y_true, y_pred):
    FN = 0
    for i, j in zip(y_true, y_pred):
        if i == 1 and j == 0:
            FN += 1
    return FN

def calculate_Precision (y_true, y_pred):
    TP=calculate_TP(y_true, y_pred)
    FP=calculate_FP(y_true, y_pred)
    Precision=TP/(TP+FP)
    return Precision

def calculate_Recall (y_true, y_pred):
    TP=calculate_TP(y_true, y_pred)
    FN=calculate_FN(y_true, y_pred)
    Recall=TP/(TP+FN)
    return Recall
```

电子商务欺诈与反欺诈

4.3 电子商务恶意用户检测

5.分类器的评价指标

```
if __name__ == '__main__':
    y_true = [1,1,0,1,0,1] #1代表预测为恶意用户，0代表预测为正常用户
    y_pred = [1,1,1,0,0,0]#1代表预测为恶意用户，0代表预测为正常用户
    Precision=calculate_Precision(y_true,y_pred)
    print('Precision:',Precision)
    Recall=calculate_Recall(y_true,y_pred)
    print('Recall:',Recall)
    F_measure=2*Precision*Recall/(Precision+Recall)
    print('F_measure:',F_measure)
```

```
Precision: 0.6666666666666666
Recall: 0.5
F_measure: 0.5714285714285715
```

电子商务欺诈与反欺诈

4.3 电子商务恶意用户检测

5.分类器的评价指标

调用sklearn中的confusion_matrix:

```
from sklearn.metrics import confusion_matrix
y_true = [1,1,0,1,0,1]#1代表预测为恶意用户，0代表预测为正常用户
y_pred = [1,1,1,0,0,0]#1代表预测为恶意用户，0代表预测为正常用户
ConfusionMatrix= confusion_matrix(y_true, y_pred, labels=[0, 1])
#print(ConfusionMatrix)
#print(ConfusionMatrix.ravel())
TN, FP, FN, TP =ConfusionMatrix.ravel()
print("TN, FP, FN, TP:",TN, FP, FN, TP)
Precision=TP/(TP+FP)
Recall=TP/(TP+FN)
F_measure=2*Precision*Recall/(Precision+Recall)
print('Precision:',Precision)
print('Recall: ',Recall)
print('F_measure:',F_measure)
```

```
TN, FP, FN, TP: 1 1 2 2
Precision: 0.6666666666666666
Recall:  0.5
F_measure: 0.5714285714285715
```

电子商务欺诈与反欺诈

4.3 电子商务恶意用户检测

精度和召回率举例：假设测试集中共100封邮件，其中10封为垃圾邮件，垃圾邮件识别系统在测试时，识别出了5封垃圾邮件，则识别出来的垃圾邮件的确全是垃圾邮件，因此垃圾邮件的识别精度为 100%，但10封垃圾邮件中只识别出5封，召回率只有50%。

识别出来的正例大部分都是真正例，假正的就少了

- ✓ 精度越高，分类器的假正类错误率就越低

大部分正例都被识别出来了，被误识别为负例的就少了

- ✓ 召回率高的分类器很少将正例误分为负例

4.4 电子商务恶意用户检测

三个臭皮匠顶个诸葛亮

组合(Ensemble)分类器

组合分类器：从训练样本得到一组分类器（分类模型可以不同），综合多个分类器的预测结果给出一个精度较高的预测结果

- ✓ 基础分类器之间应该相互独立
- ✓ 基础分类器应当优于随机猜测分类器(错误率<0.5)

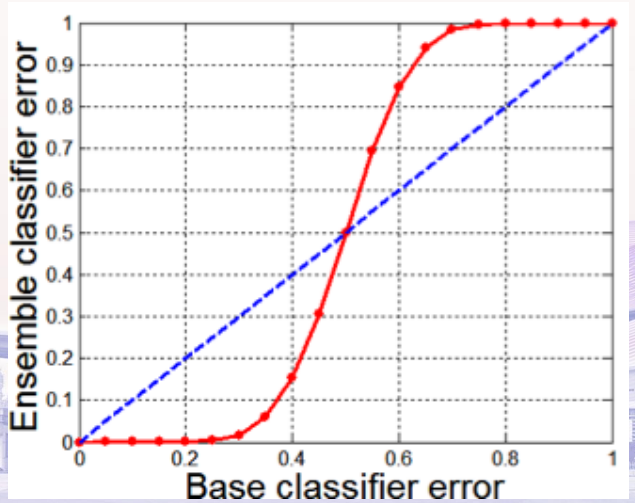
例子：假设有25个基础分类器，每个分类器的错误率都是0.35，且相互独立，组合分类器的错误率为：

$$P(X \geq 13) = \sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

Bagging：其原理是从现有数据中有放回抽取若干个样本构建分类器，重复若干次建立若干个分类器进行投票，通过投票决定最终的分类结构

RandomForest：对随机选取的子样本集分别建立m个CART (Classifier and Regression Tree)，然后投票决定最终的分类结果

GBDT (Gradient Boosting Decision Tree)：一种迭代的决策树算法，该算法由多棵决策树组成，所有树的结论累加起来做最终答案



谢谢观赏 下节课见