

推荐算法中的特征工程

杨旭东

阿里巴巴算法专家

自我介绍

- 杨旭东
 - 阿里云-计算平台事业部-机器学习PAI
 - 前 阿里巴巴-搜索事业部-推荐算法团队
- 知乎专栏《算法工程师的进阶之路》作者
- 欢迎扫码关注~

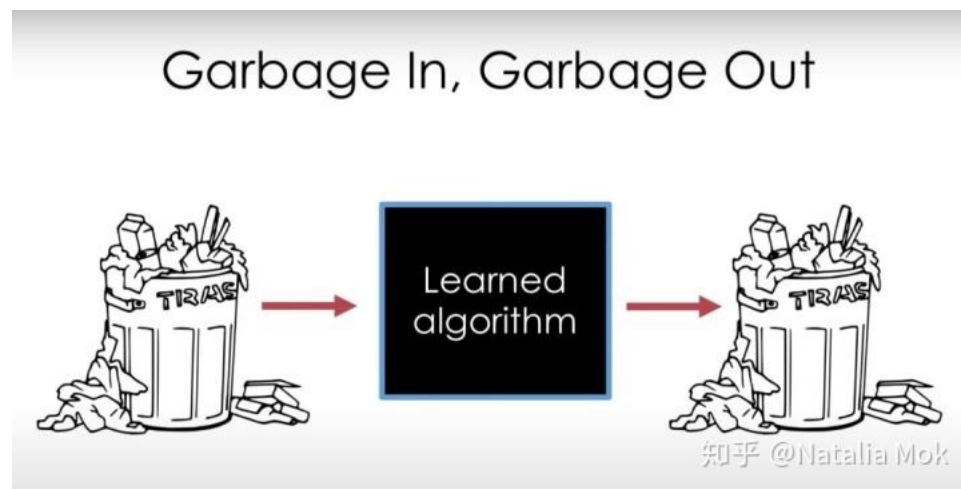


大纲

- 一．为什么要精做特征工程
- 二．何谓好的特征工程
- 三．常用的特征变换操作
- 四．搜推广场景下的特征工程

为什么要精做特征工程

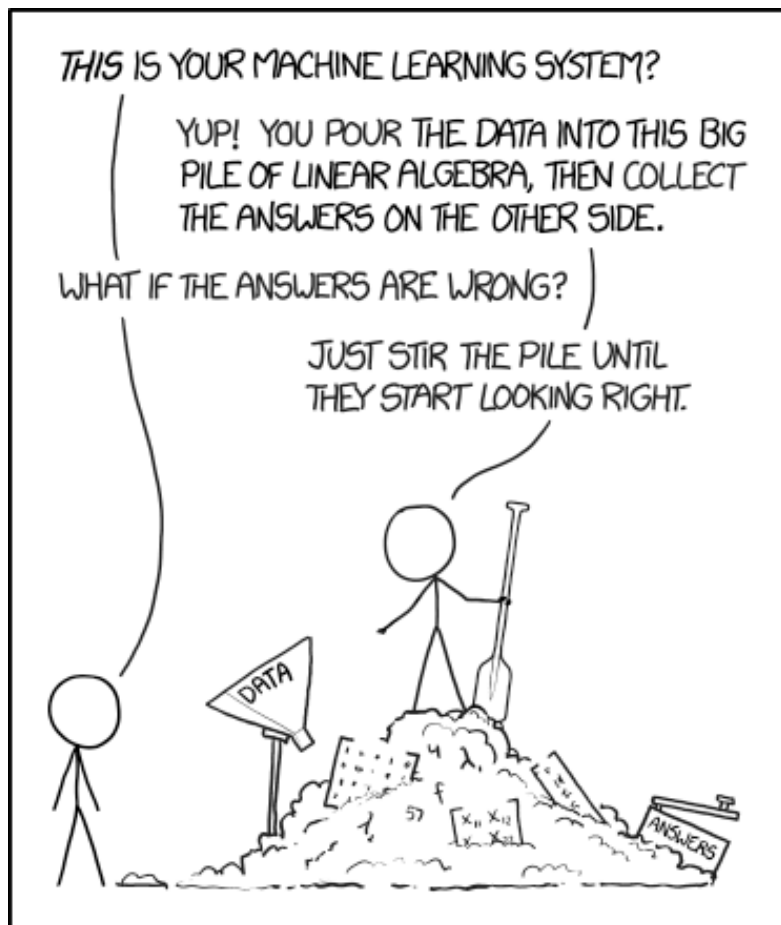
- 数据和特征决定了效果的上界，算法和模型只是逼近上界的手段



- 特征工程是编码领域专家经验的重要手段
- 好的特征工程能够显著提升模型性能
 - 高质量的特征能够大大简化模型复杂度

特征工程的常见误区

- 误区一：深度学习时代不需要特征工程



- 搜索、推荐、广告领域，数据主要以关系型结构组织
- 特征生成、变换操作的两大类型：
 - Row-based: e.g. feature interaction
 - Column-based: e.g. counting, tf-idf

	it	is	puppy	cat	pen	a	this
it is a puppy	1	1	1	0	0	1	0
it is a kitten	1	1	0	0	0	1	0
it is a cat	1	1	0	1	0	1	0
that is a dog and this is a pen	0	2	0	0	1	2	1
it is a matrix	1	1	0	0	0	1	0

$$w_{x,y} = \text{tf}_{x,y} \times \log\left(\frac{N}{\text{df}_x}\right)$$

TF-IDF

Term x within document y

$\text{tf}_{x,y}$ = frequency of x in y
 df_x = number of documents containing x
N = total number of documents

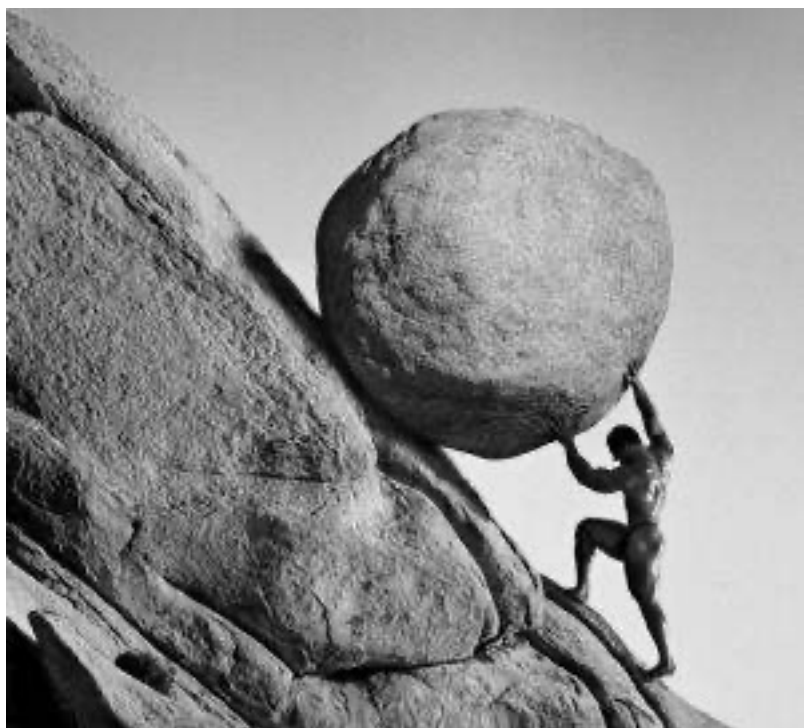
- 模型某种程度上可以学习row-based的特征变换；但无法学习column-based的特征变换
 - 一次只能接受一个小批次的数据

特征工程的常见误区

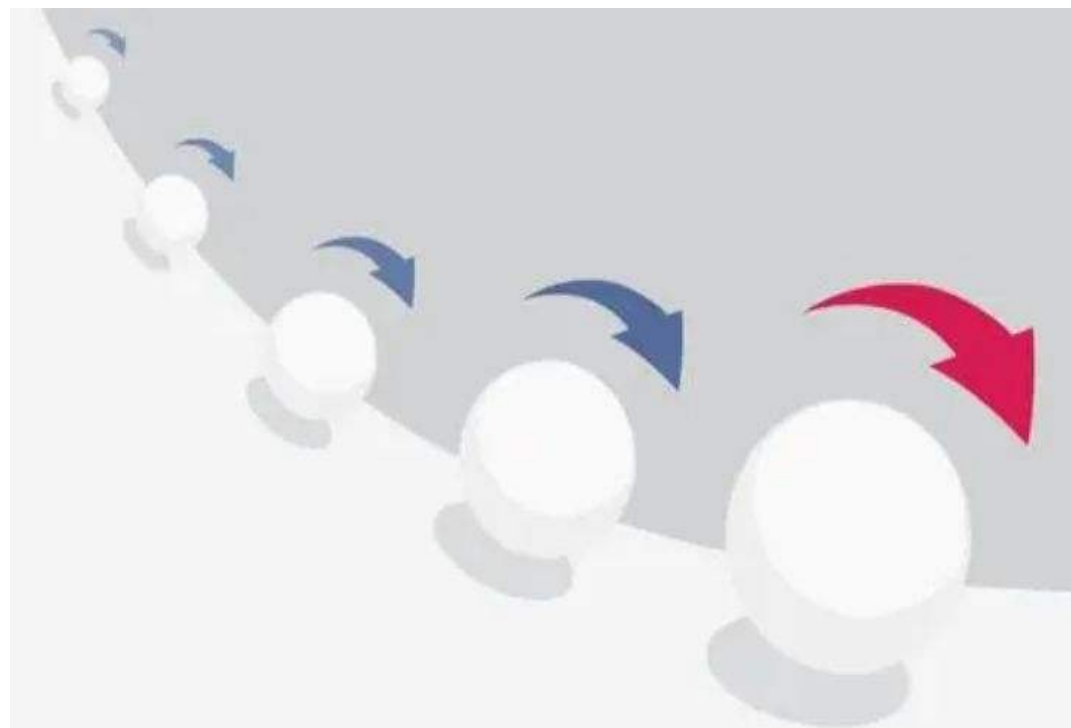
- 误区二：有了AutoFE工具就不再需要手工做特征工程
 - AutoFE的研究尚处于初级阶段
 - 主要依赖特征变换、生成、搜索与评估
 - 无法自动识别场景的特殊性
 - 瓶颈：评估特征子集的有效性
 - 特征工程非常依赖于数据科学家的业务知识、直觉和经验
 - 富有创造性和艺术性

特征工程的常见误区

- 误区三：特征工程没有技术含量



算法模型的学习



特征工程的经验

大纲

- 一．为什么要精做特征工程
- 二．何谓好的特征工程
- 三．常用的特征变换操作
- 四．搜推广场景下的特征工程

什么是好的特征工程

- 高质量特征
 - 有区分性 (Informative)
 - 特征之间相互独立 (Independent)
 - 简单易于理解 (Simple)
- 伸缩性 (Scalable) ：支持大数据量、高基数特征
- 高效率 (Efficient) ：支持高并发预测、低维
- 灵活性 (Flexible) ：对下游任务有一定的普适性
- 自适应 (Adaptive) ：对数据分布的变化有一定的鲁棒性

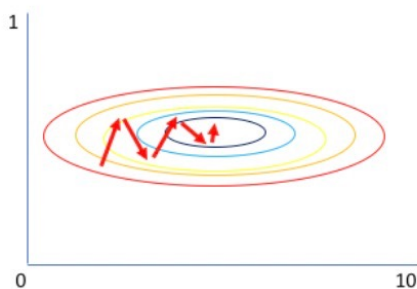
大纲

- 一．为什么要精做特征工程
- 二．何谓好的特征工程
- 三．常用的特征变换操作
- 四．搜推广场景下的特征工程

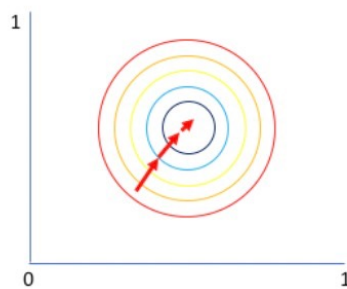
数值型特征的常用变换

• 特征缩放

Why normalize?



Gradient of larger parameter dominates the update



Both parameters can be updated in equal proportions

思考题：

1. 如何量化短视频的流行度（播放次数）？
2. 如何量化商品“贵”或“便宜”的程度？
3. 如何量化用户对新闻题材的偏好度？

1. Min-Max: $x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \in [0, 1]$

2. Scale to $[-1, 1]$: $x_{norm} = \frac{x - \max(x) + \min(x)}{\max(x) - \min(x)}$

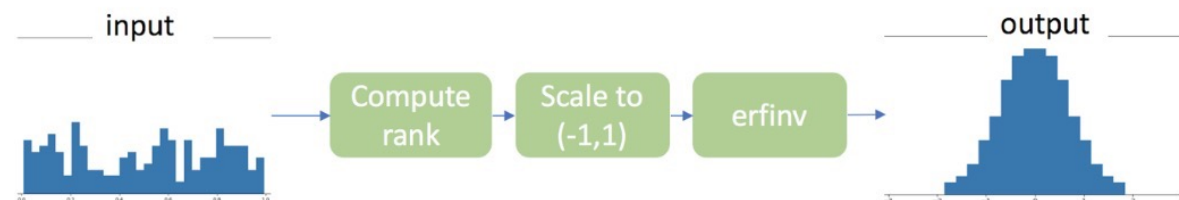
3. Z-score: $x_{norm} = \frac{x - \text{mean}(x)}{\text{std}(x)} \sim N(0, 1)$

4. Log-based: $x_{log} = \log(1 + x)$

$$x_{log-norm} = \frac{x_{log} - \text{mean}(x_{log})}{\text{std}(x_{log})}$$

5. L2 normalize: $x_{norm} = \frac{x}{\|x\|_2}$

6. Gauss Rank:



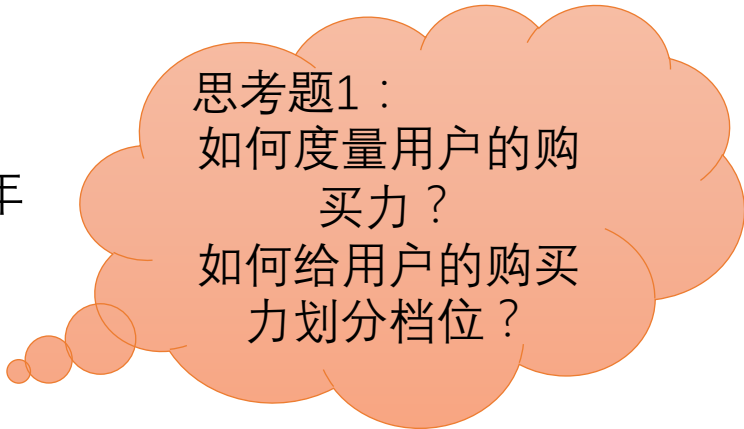
数值型特征的常用变换

- Robust scaling: $x_{scaled} = \frac{x - median(x)}{IQR}$


1	Original	Standardization	Max-Min Scaler	Robust Scaler
2	6.9314183	-0.2244971	0.0000003	0.8283487
3	2.6674115	-0.2244979	0.0000001	0.0690181
4	7.7248183	-0.2244970	0.0000003	0.9696367
5	5.7388433	-0.2244973	0.0000002	0.6159760
6	0.8965615	-0.2244982	0.0000000	-0.2463333
7	4.5147618	-0.2244975	0.0000002	0.3979926
8	2.9934144	-0.2244978	0.0000001	0.1270724
9	4.8708377	-0.2244975	0.0000002	0.4614023
10	4.2797819	-0.2244976	0.0000002	0.3561476
11	1.0085616	-0.2244982	0.0000000	-0.2263885
12	5.5166580	-0.2244974	0.0000002	0.5764094
13	1.1171326	-0.2244981	0.0000000	-0.2070542
14	0.4069897	-0.2244983	0.0000000	-0.3335159
15	5.0536949	-0.2244975	0.0000002	0.4939654
16	8.4068370	-0.2244969	0.0000003	1.0910900
17	8.9588050	-0.2244968	0.0000003	1.1893840
18	0.9543401	-0.2244982	0.0000000	-0.2360442
19	94750.5292279	-0.2079018	0.0037104	16872.6857158
20	2051.2433203	-0.2241390	0.0000803	364.8776314
21	25536631.9371928	4.2485000	1.0000000	4547540.7645023

数值型特征的常用变换

- Binning(分箱)
 - 连续特征离散化
 - E.g. 年龄段划分：儿童、青少年、中年、老年
 - Why
 - 非线性变换
 - 增强特征可解释性
 - 对异常值不敏感、防止过拟合
 - 统计、组合
 - 无监督分箱
 - 固定宽度分箱
 - 分位数分箱
 - 对数转换并取整
 - 有监督分箱
 - 卡方分箱
 - 决策树分箱



思考题1：
如何度量用户的购买力？
如何给用户的购买力划分档位？

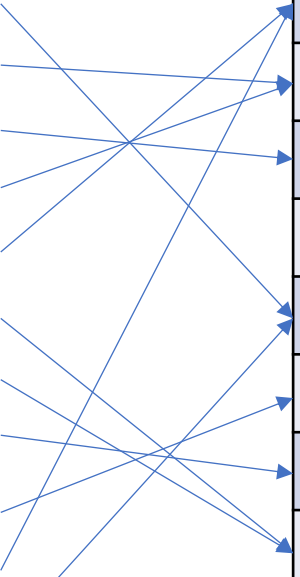


思考题2：
经纬度如何分箱？

GeoHash

特征Binning示例

User	Category	Count	bin	boundary
Alice	Beauty	209	0	
Alice	Fashion	34	1	10
Alice	Entertainment	90	2	50
Alice	Women	10	3	100
Alice	Technology	1	4	200
Bob	Military	811	5	400
Bob	Sport	999	6	800
Bob	Politics	570	7	1000
Bob	Science	210	8	
Joe	Society	7		
Joe	Game	124		



Binning

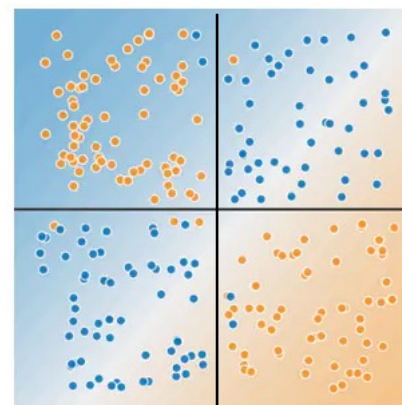
- ✗ Bad idea: 全局 binning
- ✓ Good idea: 按用户分组 binning (不同用户的行为频次可能差异较大)
- ✗ Bad idea: store boundaries for online binning (may not be updated in time, and need one group of boundaries per user)
- ✓ Good idea: store #bin for online predicting

统计特征的binning本质上是按照count排序后对rank做一个划分

类别型特征的常用变换

- 交叉组合
 - 单特征区分性不强时，可尝试组合不同特征

mean count			mean count			mean count			
f1			f2			f1	f2		
0	0.5	100	0	0.5	200	0	0	0.9	50
1	0.5	100	1	0.5	200		1	0.1	50
2	0.5	200				1	0	0.9	50
							1	0.1	50
						2	0	0.1	100
							1	0.9	100



$$x_3 = x_1 x_2$$

类别型特征的常用变换

- 分箱
 - 高基数特征相对于低基数特征处于支配地位（尤其在tree based模型中）
 - 容易引入噪音，导致模型过拟合
 - 一些值可能只会出现在训练集中，另一些可能只会出现在测试集中
- 如何装箱
 - 基于业务理解
 - Back Off
 - 决策树模型

类别型特征的常用变换

- Count Encoding
 - 统计类别特征的frequency
- Target Encoding
 - 按照类别特征分组计算 target 的概率
 - 概率值不置信时需要做平滑

$$TE_{target}([Categories]) = \frac{count([Categories]) * mean_{target}([Categories]) + w_{smoothing} * mean_{target}(global)}{count([Categories]) + w_{smoothing}}$$

- Odds Ratio

$$\theta = \frac{p_1/(1-p_1)}{p_2/(1-p_2)}$$

$$\frac{(5/125)/(120/125)}{(995/19875)/(18880/19875)} = 0.7906$$

User	Category	Click	Non-Click	Total
Alice	Bag	5	120	125
Alice	Not Bag	995	18880	19875
Total	-	1000	19000	20000

类别型特征的常用变换

- WOE (Weight Of Evidence)

- $WOE = \ln\left(\frac{Event\%}{NonEvent\%}\right)$

Variable Name	Min. Value	Max. Value	Count	# Event	# Non Event	Event%	Non event%	WOE
Age	10	20	1200	150	1050	28.3%	19.0%	0.3992
Age	21	30	900	120	780	22.6%	14.1%	0.4733
Age	31	40	1090	110	980	20.8%	17.7%	0.1580
Age	41	50	1460	100	1360	18.9%	24.6%	-0.2650
Age	50	inf	1410	50	1360	9.4%	24.6%	-0.9582
Total			6060	530	5530			

时序特征

- 历史事件分时段统计
 - 统计过去1天、3天、7天、30天的总（平均）行为数
 - 统计过去1天、3天、7天、30天的行为转化率
- 差异
 - 环比、同比
- 行为序列
 - 需要模型配合

大纲

- 一．为什么要精做特征工程
- 二．何谓好的特征工程
- 三．常用的特征变换操作
- 四．搜推广场景下的特征工程

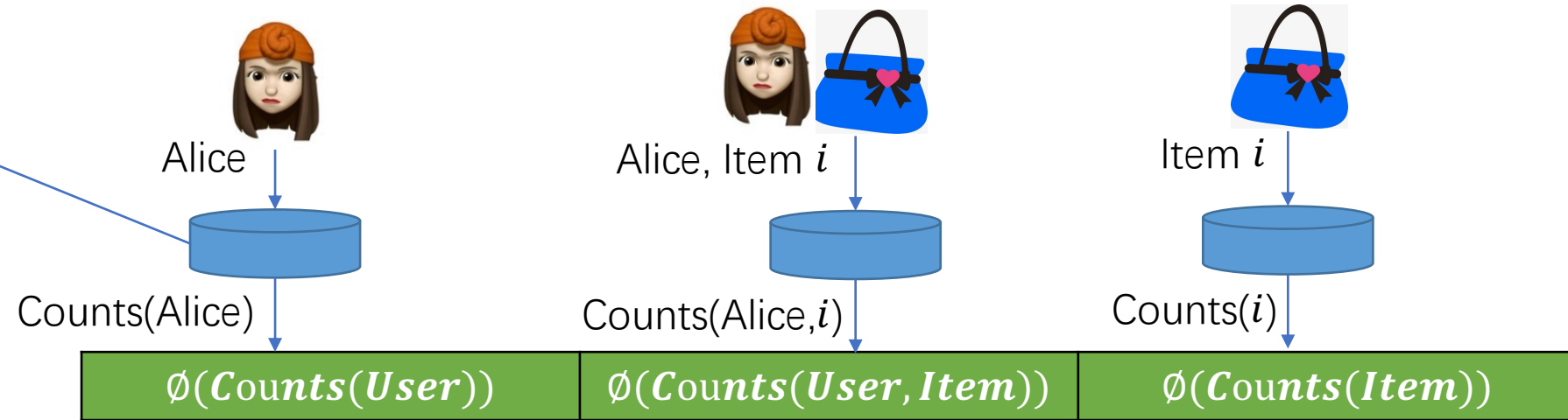
关系型数据下的数据挖掘



- 高基数(high-cardinality)属性表示为特征时的挑战
 - Scalable: to billions of attribute values
 - Efficient: ~10⁵⁺ predictions/sec/node
 - Flexible: for a variety of downstream learners
 - Adaptive: to distribution change

Learning with counts

User	N ⁺	N ⁻
Alice	7	134
Bob	17	235
Joe	2	274
.....
REST	7891	129437



- Features are **per-behavior-type, per-time-period, per-label counts** [+backoff]

$$\phi = [IsRest \quad trans(N^+) \quad trans(N^-) \quad target_encoding(N^+, N^-)]$$

- ✓ **Scalable** head in memory + tail in backoff
- ✓ **Efficient** low cost, low dimensionality
- ✓ **Flexible** low dimensionality works well with non-linear learners
- ✓ **Adaptive** new values easily added, back-off for infrequent values, temporal counts

Learning with counts: aggregation

- Aggregate $Count(y, bin(x))$ for different $bin(x)$

Item	N ⁺	N ⁻
101	217	934
102	170	635
103	52	474
.....

bin

Category, Price Level	N ⁺	N ⁻
Food,1	1127	90134
Food,3	517	2350
Furniture,5	92	5274
.....

- Bin function: **any projection**
 - 无监督：等距、等频、聚类
 - 有监督：卡方分箱、决策树分箱

- Backoff option: “tail bin”

User,Cat,PriceLvl	N ⁺	N ⁻
Alice,Hat,5	7	134
Bob,Food,8	9	101
Joe, Stationery, 3	2	99
.....
REST	7891	129437

Cross

User	N ⁺	N ⁻
Alice	7	134
Bob	17	235
Joe	12	274
.....
REST	7891	129437

bin

Age	N ⁺	N ⁻
<18	1017	63134
18~25	43917	909235
26~35	98944	2009974
.....
>60	7891	129437



Learning from counts: combiner training



User	N ⁺	N ⁻
Alice	7	134
Bob	17	235
.....
REST	7891	129437

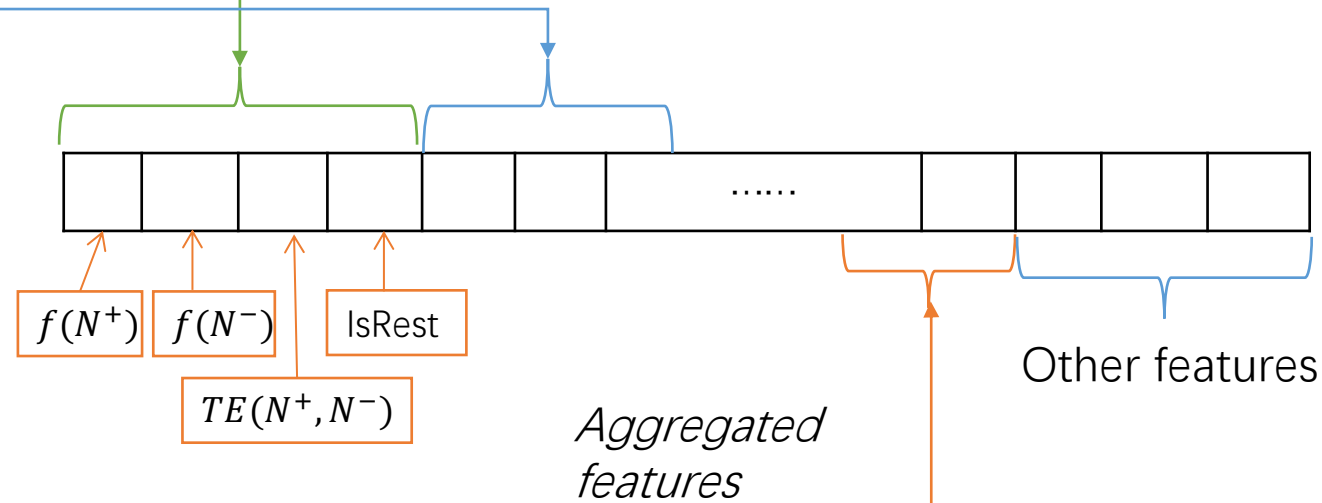


Item	N ⁺	N ⁻
101	217	934
102	170	635
103	52	474
.....

User,Cat,PriceLvl	N ⁺	N ⁻
Alice,Bag,5	7	134
Bob,Food,8	9	101
.....
REST	7891	29437

Train non-linear model on count-based features

- Counts, transforms, lookup properties
- Additional features can be injected



Counting

Train predictor

T_{now}

time

Where did it come from?

The second set of features, historical CTR features, are the most critical features used in our click models, and also provide a strong signal in relevance prediction. The past performance of a query-ad pair, when available, is a very accurate estimate of its future performance.

Hillard et al. 2011

In the CFB model, we keep historical counts of impressions and clicks for each page ad pair at multiple levels of granularity.

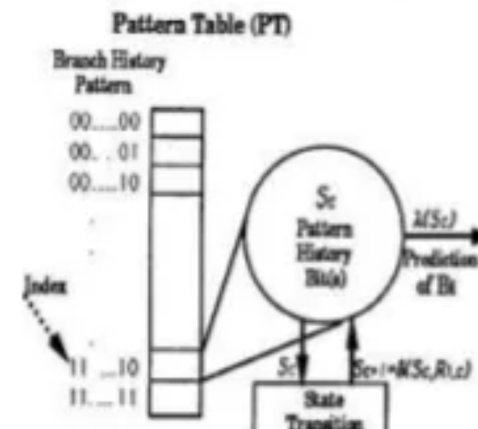
Li et al. 2010

We next aggregate event counts of a user (identified by cookie) over a configurable period of time and then merge counts with a same composite key (cookie, time) into a single entry.

Pavlov et al. 2009

we add a feature, the count of connections that have the same service and src_bytes as the current connection record in the past 140 seconds. When an attribute (with different values) is repeated several times in the rule, we add a corresponding *average* feature.

Lee et al. 1998



Yeh and Patt, 1991

查漏补缺

1. 列存实体 (entity)
2. 实体分箱 & 单维度统计/编码
3. 特征交叉 & 多维度统计/编码

Bin Counting

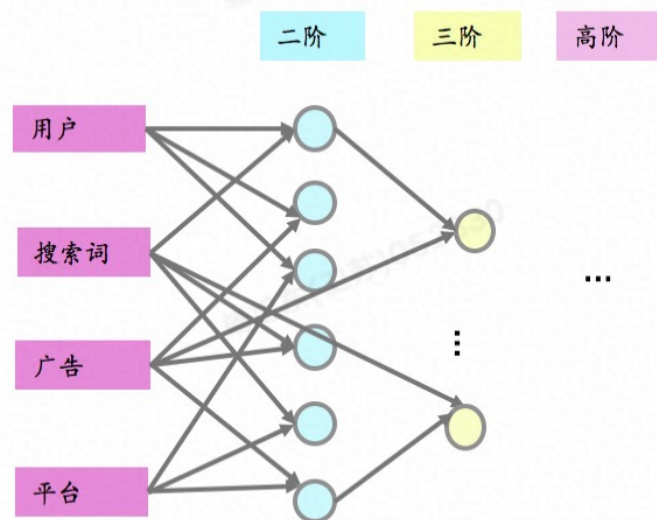
bin cross count

单特征矩阵

基础特征	泛化特征	统计特征	标签特征	行为特征	预训练	上下文

基本元素

交叉特征



感谢聆听！



Q&A