点击率预估方法介绍

钱烽
qf6101 at gmail.com

概述

- 点击率预估问题及意义
- 点击率预估建模及求解
- 模型技术
  - Factorization Machines
- 特征技术
  - 历史特征
  - GBDT特征

Part I

点击率预估问题及意义

点击率预估问题

基本概念

- CTR： Click-Through Rate
- pCTR： Click-Through Rate Prediction

pCTR的输入

- $a_1, a_2, ..., a_n$ 候选广告集合及其属性
- u 当前请求的用户属性
- c 当前请求的上下文属性

pCTR的输出

- $\{a_{i1}, s_{i1}\}, \{a_{i2}, s_{i2}\}, ..., \{a_{in}, s_{in}\}$ 排序后的广告及其得分

点击率预估问题（续）

应用扩展

- 广告系统：a 表示广告
- 推荐系统：a 表示内容、商品、服务等

策略扩展

- 侧重于a的策略：重定向广告和推荐
- 侧重于u的策略：个性化推荐
- 侧重于c的策略：原生广告

点击率预估的意义

- 商业意义
  – 投放什么广告直接影响平台收益
  – **eCPM = pCTR * CPC**
- 技术意义
  – 在广告、推荐、搜索系统中，采用机器学习方法对广告或内容被点击的概率进行预测和排序（广告乘以单价后再排序），以提高曝光转化率，位于检索模块之后、推送模块之前

Part II

点击率预估的建模及求解

点击率预估的建模

hypotheses

- $P(x) = h_z(x) = \frac{1}{1+e^{-z}}$ where z is linear regression model

models

- Logistic Regression： $z = w^t x$
- Factorization Machine： $z = w_0 + w^t x + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j$

parameters

- $\theta \in \{w_0, w, v\}$

点击率预估的求解

loss = negative log likelihood

- $y \in \{-1, +1\}$： $\ell(\theta) = \log(1 + e^{-yz})$
- $y \in \{0, 1\}$： $\ell(\theta) = -\left(y \log(h_\theta(x)) + (1 - y) \log(1 - h_\theta(x))\right)$

gradient

- $y \in \{-1, +1\}$： $\frac{\partial \ell(\theta)}{\partial \theta} = -y\left(1 - \frac{1}{1+\exp(-yz)}\right) \cdot \frac{\partial z}{\partial \theta}$
- $y \in \{0, 1\}$： $\frac{\partial \ell(\theta)}{\partial \theta} = (h_\theta(x) - y) \cdot \frac{\partial z}{\partial \theta}$

optimization

- Stochastic Gradient Descent
- L-BFGS

点击率预估的模型选择

为什么使用线性模型（LR、FM）？

- 具有很好的可解释性，可以输出概率，适合Ranking
- 适合高维稀疏特征，训练和预测都很快
- 容易大规模并行求解，适合分布式计算

模型参数选择

- Linear Search instead of Grid Search
- Wolf Line Search to dynamically choose learning rate
- 人工经验：观察，调整；再观察，再调整……

不平衡数据的处理

采样方法

- 对正样本做重采样（SMOTE算法），对负样本做下采样
- 正负样本比例控制在1:2以内

梯度惩罚

- 根据样本数比例，对负样本的梯度或损失值做惩罚

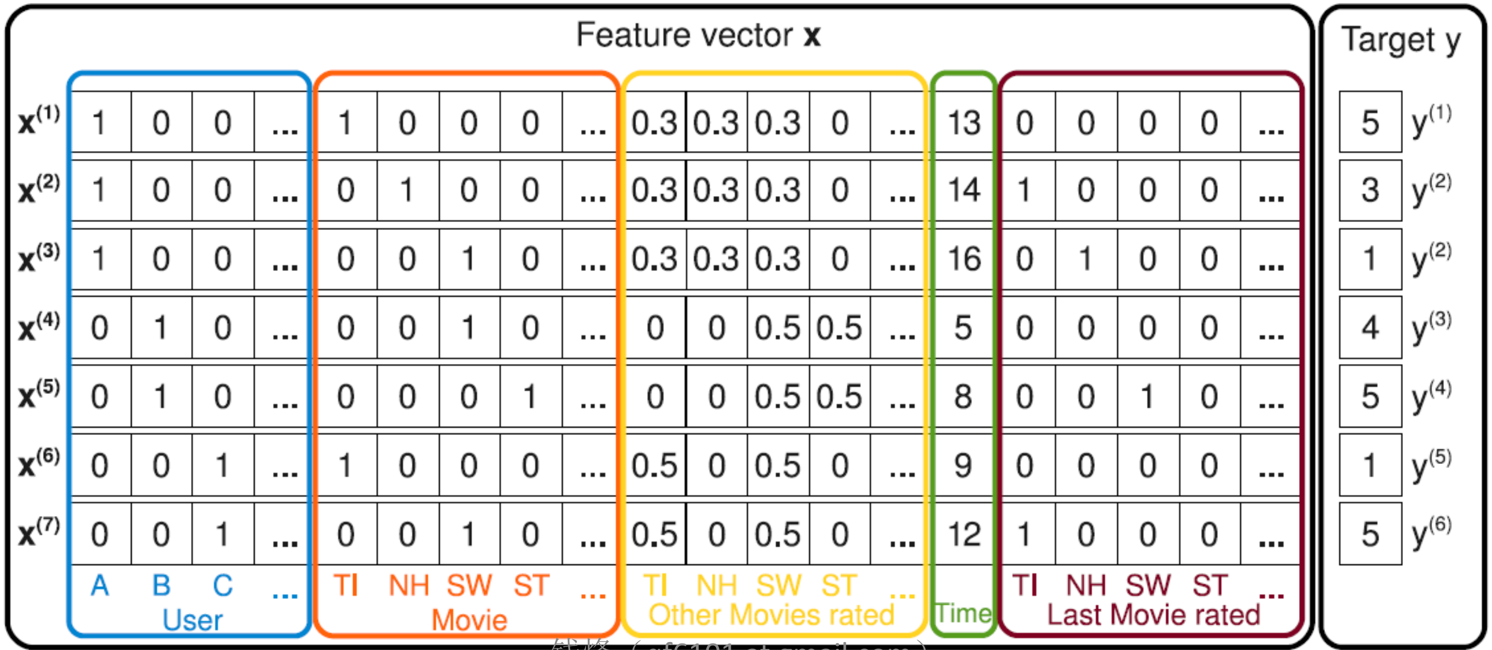
- 数据采样引入的新问题
  - Calibration = avg. pCTR / BG CTR
  - 模型校准

Part III

模型技术
——FACTORIZATION MACHINES

FM的动机

- 目标数据类型
  - Categorical, Set-Categorical, Real Valued
  - Sparse Representation (DT and SVM may fail)



	User A	User B	User C	User ...	Movie TI	Movie NH	Movie SW	Movie ST	Movie ...	Other Movies rated TI	Other Movies rated NH	Other Movies rated SW	Other Movies rated ST	Other Movies rated ...	Time	Last Movie rated TI	Last Movie rated NH	Last Movie rated SW	Last Movie rated ST	Last Movie rated ...	Target y	
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	$y^{(2)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	$y^{(3)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	$y^{(4)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	$y^{(5)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	$y^{(6)}$

FM的动机（续）

- 目标任务
  - Classification, Regression, Ranking, etc.
  - Weaken importance of feature engineering
- 目标模型
  - A general model that subsumes a wide variety of factorization models (e.g., polynomial kernel SVM, SVD++, PITF and PFMC).
- 目标算法
  - Linear time complexity

传统方法

- Linear Regression (LR)
  - Let $\mathbf{x} \in \mathbb{R}^p$ be an input vector with p predictor variables.
  - Model equation:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i \, x_i$$

  - Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p$$

  $\mathcal{O}(p)$ model parameters.

传统方法（续）

- Polynomial Regression
  - Let $\mathbf{x} \in \mathbb{R}^p$ be an input vector with p predictor variables.
  - Model equation (degree 2):

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j \geq i}^{p} w_{i,j}\, x_i\, x_j$$

  - Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad \mathbf{W} \in \mathbb{R}^{p \times p}$$

  $\mathcal{O}(p^2)$ model parameters.

传统方法（续）

- Weaknesses
  - Linear regression has no user-item interaction.
    - $\Rightarrow$ Linear regression is not expressive enough.
  - Polynomial regression includes pairwise interactions but cannot estimate them from the data.
    - $n \ll p^2$: number of cases is much smaller than number of model parameters.
    - Max.-likelihood estimator for a pairwise effect is:

$$w_{i,j} = \begin{cases} y - w_0 - w_i - w_u, & \text{if } (i,j,y) \in S. \\ \text{not defined}, & \text{else} \end{cases}$$

    - Polynomial regression cannot generalize to *any* unobserved pairwise effect.

Factorization Machines

- Modeling

  - Let $\mathbf{x} \in \mathbb{R}^p$ be an input vector with p predictor variables.
  - Model equation (degree 3):

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

$$+ \sum_{i=1}^{p} \sum_{j>i}^{p} \sum_{l>j}^{p} \sum_{f=1}^{k} v_{i,f}^{(3)} v_{j,f}^{(3)} v_{l,f}^{(3)} x_i x_j x_l$$

  - Model parameters:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad \mathbf{V} \in \mathbb{R}^{p \times k}, \quad \mathbf{V}^{(3)} \in \mathbb{R}^{p \times k}$$

Factorization Machines（续）

- Advantages
  - FMs work with real valued input.
  - FMs include variable interactions like polynomial regression.
  - Model parameters for interactions are factorized.
  - Number of model parameters is $\mathcal{O}(k\,p)$ (instead of $\mathcal{O}(p^2)$ for poly. regr.).

Factorization Machines (续)

- Efficient Computation

The model equation of an FM can be computed in $\mathcal{O}(p\,k)$.

Proof:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{p} w_i\,x_i + \sum_{i=1}^{p}\sum_{j>i}^{p}\langle \mathbf{v}_i, \mathbf{v}_j\rangle\, x_i\,x_j$$

$$= w_0 + \sum_{i=1}^{p} w_i\,x_i + \frac{1}{2}\sum_{f=1}^{k}\left[\left(\sum_{i=1}^{p} x_i\,v_{i,f}\right)^2 - \sum_{i=1}^{p}(x_i\,v_{i,f})^2\right]$$

- In the sums over i, only non-zero x_i elements have to be summed up $\Rightarrow \mathcal{O}(N_z(\mathbf{x})\,k)$.
- (The complexity of polynomial regression is $\mathcal{O}(N_z(\mathbf{x})^2)$.)

Factorization Machines（续）

- Multilinearity
  - Opportunities to efficient learning and engineering

FMs are multilinear:

$$\forall \theta \in \Theta = \{w_0, \mathbf{w}, \mathbf{V}\}: \qquad \hat{y}(\mathbf{x}, \theta) = h_{(\theta)}(\mathbf{x})\,\theta + g_{(\theta)}(\mathbf{x})$$

where $g_{(\theta)}$ and $h_{(\theta)}$ do not depend on the value of θ.

E.g. for second order effects ($\theta = v_{l,f}$):

$$\hat{y}(\mathbf{x}, v_{l,f}) := \overbrace{w_0 + \sum_{i=1}^{p} w_i\, x_i + \sum_{i=1}^{p} \sum_{j=i+1}^{p} \sum_{\substack{f'=1 \\ (f' \neq f) \vee (l \notin \{i,j\})}}^{k} v_{i,f'}\, v_{j,f'}\, x_i\, x_j}^{g_{(v_{l,f})}(\mathbf{x})}$$

$$+ \underbrace{v_{l,f}\; x_l \sum_{i=1, i \neq l} v_{i,f}\, x_i}_{h_{(v_{l,f})}(\mathbf{x})}$$

Learning FM with SGD

Stochastic Gradient Descent

$$\frac{\partial}{\partial\theta}\hat{y}(\mathbf{x}) = \begin{cases} 1, & \text{if } \theta \text{ is } w_0 \\ x_i, & \text{if } \theta \text{ is } w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta \text{ is } v_{i,f} \end{cases}$$

- For each training case $(\mathbf{x}, y) \in S$, SGD updates the FM model parameter θ using:

$$\theta' = \theta - \alpha \left((\hat{y}(\mathbf{x}) - y) h_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)} \theta \right)$$

- α is the learning rate / step size.
- $\lambda_{(\theta)}$ is the regularization value of the parameter θ.
- SGD can easily be applied to other loss functions.

Remaining Problems

- 写论文就像医生治病：头痛医头脚痛医脚
  - 前半部分描述病情 (Bing Ru Gao Huang)
  - 后半部分开药方 (Yao Dao Bing Chu)
- SGD needs to tune the learning rate
  - ALS algorithm for Sum of Squared Loss [SIGIR 2011]
- Needs to tune the regularization parameters
  - Adaptive Regularization [WSDM 2012]
  - Bayesian Inference [NIPS-WS 2011]
- Relational data introduces repetitive computation
  - Block Structure to scale algorithms [VLDB 2013]

All adopt multilinearity

Leaning FMs with ALS

- Analytical solution for least squares

$$\frac{\partial}{\partial\theta}\text{RLS-OPT} = \sum_{(\mathbf{x},y)\in S} 2\ (\hat{y}(\mathbf{x}) - y)\ h_{(\theta)}(\mathbf{x}) + 2\,\lambda_{(\theta)}\,\theta$$

$$\sum_{(\mathbf{x},y)\in S} 2\ (\hat{y}(\mathbf{x}) - y)\ h_{(\theta)}(\mathbf{x}) + 2\,\lambda_{(\theta)}\,\theta = 0$$

$$\Leftrightarrow \sum_{(\mathbf{x},y)\in S} \big(g_{(\theta)}(\mathbf{x}) - y\big)\ h_{(\theta)}(\mathbf{x}) + \sum_{(\mathbf{x},y)\in S} \theta\, h^2_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)}\,\theta = 0$$

$$\Leftrightarrow \theta = -\frac{\sum_{(\mathbf{x},y)\in S} \big(g_{(\theta)}(\mathbf{x}) - y\big)\, h_{(\theta)}(\mathbf{x})}{\sum_{(\mathbf{x},y)\in S} h^2_{(\theta)}(\mathbf{x}) + \lambda_{(\theta)}}$$

- Using caches of intermediate results, the runtime for updating all model parameters is $O(k\,N_z(X))$.
- The advantage of ALS compared to SGD is that no learning rate has to be specified.
- ALS can be extended to classification [Rendle, 2012].

Leaning FMs with ALS (续)

- Recall the multilinearity

$$\hat{y}(\mathbf{x}|\theta) = g_{(\theta)}(\mathbf{x}) + \theta\, h_{(\theta)}(\mathbf{x})$$

- Reformulate for each parameter

$$\hat{y}(\mathbf{x}|w_0) = w_0 \underbrace{1}_{h_{(w_0)}(\mathbf{x})} + \underbrace{\sum_{i=1}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j}\, x_i\, x_j}_{g_{(w_0)}(\mathbf{x})}$$

$$\hat{y}(\mathbf{x}|w_l) = w_l \underbrace{x_l}_{h_{(w_l)}(\mathbf{x})} + \underbrace{w_0 + \sum_{i=1, i\neq l}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j}\, x_i\, x_j}_{g_{(w_l)}(\mathbf{x})}$$

$$\hat{y}(\mathbf{x}|v_{l,f}) := v_{l,f} \overbrace{x_l \sum_{i=1, i\neq l} v_{i,f}\, x_i}^{h_{(v_{l,f})}(\mathbf{x})}$$

$$\underbrace{+ w_0 + \sum_{i=1}^{n} w_i\, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{\substack{f'=1 \\ (f'\neq f)\vee(l\notin\{i,j\})}}^{k} v_{i,f'}\, v_{j,f'}\, x_i\, x_j}_{g_{(v_{l,f})}(\mathbf{x})}$$

Leaning FMs with ALS (续)

- Precomputing skills
  - error-terms

$$e(\mathbf{x}, y|\Theta) := \hat{y}(x|\Theta) - y$$

$$g_{(\theta)}(\mathbf{x}) - y = e(\mathbf{x}, y|\Theta) - \theta\, h_{(\theta)}(\mathbf{x})$$

$$e(x, y|\Theta^*) = e(x, y|\Theta) + (\theta^* - \theta)\, h_{(\theta)}(\mathbf{x})$$

  - h-terms

$$h_{(v_{l,f})}(\mathbf{x}) = x_l \sum_{i=1}^{n} v_{i,f}\, x_i - x_l^2\, v_{l,f}$$

$$= x_l\, q(\mathbf{x}, f|\Theta) - x_l^2\, v_{l,f}$$

$$q(\mathbf{x}, f|\Theta) := \sum_{i=1}^{n} v_{i,f}\, x_i$$

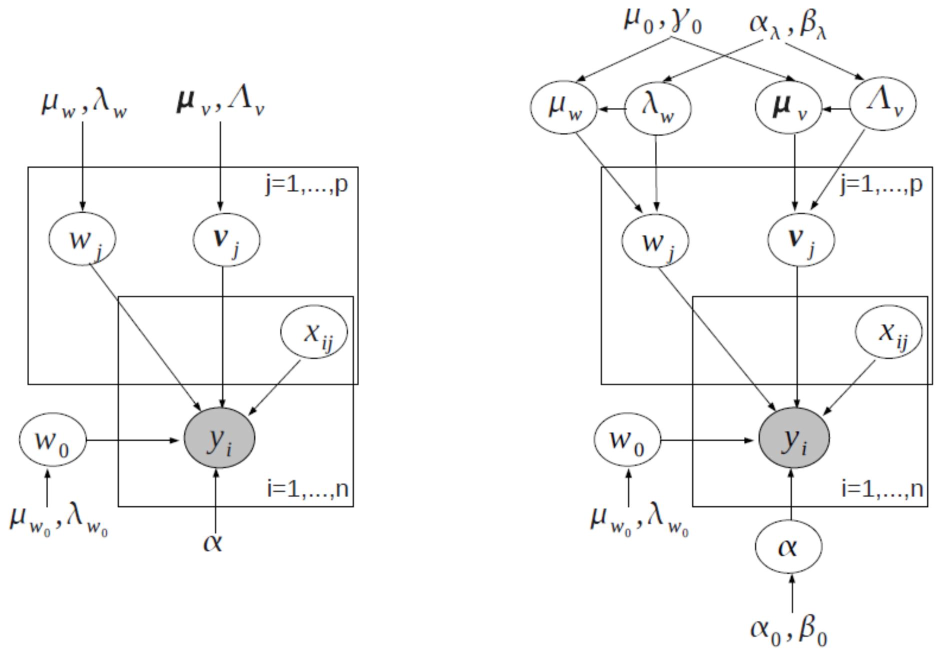
$$q(\mathbf{x}, f|\Theta^*) = q(\mathbf{x}, f|\Theta) + (v_{l,f}^* - v_{l,f})\, x_l$$

  - 无须计算g(x)

procedure LEARNALS(S)
  $w_0 \leftarrow 0$ ▷ Initialize the model parameters
  $\mathbf{w} \leftarrow (0, \ldots, 0)$
  $\mathbf{V} \sim \mathcal{N}(0, \sigma)$
  for $(\mathbf{x}, y) \in S$ **do** ▷ Precompute e and q
    $e(\mathbf{x}, y|\Theta) \leftarrow \hat{y}(\mathbf{x}, y) - y$
    for $f \in \{1, \ldots, k\}$ **do**
      $q(\mathbf{x}, f|\Theta) \leftarrow \sum_{i=1}^{n} v_{i,f}\, x_i$
    end for
  end for
  repeat ▷ Main optimization loop
    $w_0^* \leftarrow -\frac{\sum_{(\mathbf{x},y)\in S}(e(\mathbf{x},y|\Theta) - w_0)}{|S| + \lambda_{(w_0)}}$ ▷ global bias
    $e(\mathbf{x}, y|\Theta) \leftarrow e(\mathbf{x}, y|\Theta) + (w_0^* - w_0)$
    $w_0 \leftarrow w_0^*$
    for $l \in \{1, \ldots, n\}$ **do** ▷ 1-way interactions
      $w_l^* \leftarrow -\frac{\sum_{(\mathbf{x},y)\in S}(e(\mathbf{x},y|\Theta) - w_l\, x_l)x_l}{\sum_{(\mathbf{x},y)\in S} x_l^2 + \lambda_{(w_l)}}$
      $e(\mathbf{x}, y|\Theta) \leftarrow e(\mathbf{x}, y|\Theta) + (w_l^* - w_l)\, x_l$
      $w_l \leftarrow w_l^*$
    end for
    for $f \in \{1, \ldots, k\}$ **do** ▷ 2-way interactions
      for $l \in \{1, \ldots, n\}$ **do**
        $v_{l,f}^* \leftarrow -\frac{\sum_{(\mathbf{x},y)\in S}(e(\mathbf{x},y|\Theta) - v_{l,f}\, h_{(v_{l,f})}(\mathbf{x}))h_{(v_{l,f})}(\mathbf{x})}{\sum_{(\mathbf{x},y)\in S} h_{(v_{l,f})}^2(\mathbf{x}) + \lambda_{(v_{l,f})}}$
        $e(\mathbf{x}, y|\Theta) \leftarrow e(\mathbf{x}, y|\Theta) + (v_{l,f}^* - v_{l,f})h_{(v_{l,f})}(\mathbf{x})$
        $q(\mathbf{x}, f|\Theta) \leftarrow q(\mathbf{x}, f|\Theta) + (v_{l,f}^* - v_{l,f})h_{(v_{l,f})}(\mathbf{x})$
        $v_{l,f} \leftarrow v_{l,f}^*$
      end for
    end for
  until stopping criterion is met
  return $w_0, \mathbf{w}, \mathbf{V}$
end procedure

Learning with Bayesian Inference

- Applying Hierarchical prioris
- Learning with Gibbs Sampling



$$w_0 \sim \mathcal{N}(\mu_{w_0}, 1/\lambda_{w_0}), \quad \forall j \in \{1, \ldots, p\}: \; w_j \sim \mathcal{N}(\mu_w, 1/\lambda_w), \quad \mathbf{v}_j \sim \mathcal{N}(\mu_v, \Lambda_v^{-1})$$

$$\mu_w \sim \mathcal{N}(\mu_0, \gamma_0 \lambda_w), \quad \lambda_w \sim \Gamma(\alpha_\lambda, \beta_\lambda), \quad \mu_{v,f} \sim \mathcal{N}(\mu_0, \gamma_0 \lambda_{v,f}), \quad \lambda_{v,f} \sim \Gamma(\alpha_\lambda, \beta_\lambda)$$

Learning with Adaptive Regularization

- Intuitive Idea
  - alternatively solving parameters and regularization parameters

$$\text{OptReg}(S, \lambda) := \underset{\Theta}{\operatorname{argmin}} \left(\sum_{(\mathbf{x},y)\in S} l(\hat{y}(\mathbf{x}|\Theta), y) + \sum_{\theta\in\Theta} \lambda_\theta \theta^2 \right)$$ (On training data set)

$$\lambda^* := \underset{\lambda\in\mathbb{R}_+^c}{\operatorname{argmin}} \sum_{(\mathbf{x},y)\in S_V} l\left(\hat{y}(\mathbf{x}|\text{OptReg}(S_T, \lambda)), y\right)$$ (On validation data set)

- But… it's non-trivial
  - The formula Independent of regularization parameter
  - The gradient vanishes

$$\lambda^*|\Theta^t := \underset{\lambda\in\mathbb{R}_+^c}{\operatorname{argmin}} \sum_{(\mathbf{x},y)\in S_V} l\left(\hat{y}(\mathbf{x}|\Theta^t)), y\right)$$

$$\frac{\partial}{\partial\lambda} L(S_V, \Theta^t) = \frac{\partial}{\partial\lambda} \sum_{(\mathbf{x},y)\in S_V} l(\hat{y}(\mathbf{x}|\Theta^t), y) = 0$$

Learning with Adaptive Regularization (续)

- 从相邻的两次迭代观察

$$\theta^{t+1} = \theta^t - \alpha \left(\frac{\partial}{\partial \theta^t} l(\hat{y}(\mathbf{x}|\Theta^t), y) + 2\,\lambda\,\theta^t \right)$$

$$\hat{y}(\mathbf{x}|\Theta^{t+1}) = w_0^{t+1} + \sum_{l=1}^{p} w_l^{t+1}\, x_l + \sum_{l_1=1}^{p} \sum_{l_2>l_1}^{p} \langle \mathbf{v}_{l_1}^{t+1}, \mathbf{v}_{l_2}^{t+1} \rangle\, x_{l_1}\, x_{l_2}$$

- 合并上述两式

$$\begin{aligned}
\hat{y}(\mathbf{x}|\Theta^{t+1}) = {} & w_0^t - \alpha \left(\frac{\partial l(\hat{y}(\mathbf{x}|\Theta^t), y)}{\partial w_0^t} + 2\,\lambda_0\, w_0^t \right) \\
& + \sum_{l=1}^{p} x_l \left(w_l^t - \alpha \left(\frac{\partial l(\hat{y}(\mathbf{x}|\Theta^t), y)}{\partial w_l^t} + 2\,\lambda_w\, w_l^t \right) \right) \\
& + \sum_{l_1=1}^{p} \sum_{l_2>l_1}^{p} \sum_{f=1}^{k} \left[x_{l_1} \left(v_{l_1,f}^t - \alpha \left(\frac{\partial l(\hat{y}(\mathbf{x}|\Theta^t), y)}{\partial v_{l_1,f}^t} + 2\,\lambda_f\, v_{l_1,f}^t \right) \right) \right. \\
& \qquad \left. x_{l_2} \left(v_{l_2,f}^t - \alpha \left(\frac{\partial l(\hat{y}(\mathbf{x}|\Theta^t), y)}{\partial v_{l_2,f}^t} + 2\,\lambda_f\, v_{l_2,f}^t \right) \right) \right]
\end{aligned}$$

- 重写正则参数的优化目标

$$\lambda^* | \Theta^t := \operatorname*{argmin}_{\lambda \in \mathbb{R}_+^c} \sum_{(\mathbf{x},y) \in S_V} l\left(\hat{y}(\mathbf{x}|\Theta^{t+1}), y \right)$$

Learning with Adaptive Regularization (续)

- Learning with SGD

$$\lambda^{t+1} = \lambda^{t} - \alpha \frac{\partial}{\partial \lambda} l\left(\hat{y}(\mathbf{x}|\Theta^{t+1}), y\right)$$

$$\frac{\partial}{\partial \lambda_0} \hat{y}(\mathbf{x}|\Theta^{t+1}) = -2\,\alpha\, w_0^t,$$

$$\frac{\partial}{\partial \lambda_w} \hat{y}(\mathbf{x}|\Theta^{t+1}) = -2\,\alpha \sum_{i=1}^{p} w_i^t x_i$$

$$\frac{\partial}{\partial \lambda_f} \hat{y}(\mathbf{x}|\Theta^{t+1}) = -2\alpha \left[\sum_{i=1} x_i v_{i,f}^{t+1} \sum_{j=1} x_j v_{j,f}^{t} - \sum_{j=1} x_j^2 v_{j,f}^{t+1} v_{j,f}^{t} \right]$$

Learning with Adaptive Regularization (续)

- Approximation
  - 每次更新完模型参数都去更新下正则参数吗？

$$\tilde{\theta}^{t+1} := \theta^t - \alpha \left(\partial_\theta + 2\,\lambda\,\theta^t\right) \approx \theta^{t+1}$$

This approximation $\tilde{\theta}^{t+1}$ is used for the λ-steps.

procedure SolveOptAdaptiveReg(S_T, S_V)
  $w_0 \leftarrow 0$
  $\mathbf{w} \leftarrow (0, \ldots, 0)$
  $\mathbf{V} \sim \mathcal{N}(0, \sigma)$
  $\lambda \leftarrow (0, \ldots, 0)$
  $\partial \leftarrow (0, \ldots, 0)$
  repeat
    for $(\mathbf{x}, y) \in S_T$ **do**
      $\partial_0 \leftarrow \frac{\partial}{\partial w_0} l(\hat{y}(\mathbf{x}|\Theta), y)$
      $w_0 \leftarrow w_0 - \alpha\,(2\,\lambda_0\,w_0 + \partial_0)$
      for $i \in \{1, \ldots, p\} \wedge x_i \neq 0$ **do**
        $\partial_i \leftarrow \frac{\partial}{\partial w_i} l(\hat{y}(\mathbf{x}|\Theta), y)$
        $w_i \leftarrow w_i - \alpha\,(2\,\lambda_w\,w_i + \partial_i)$
        for $f \in \{1, \ldots, k\}$ **do**
          $\partial_{i,f} \leftarrow \frac{\partial}{\partial v_{i,f}} l(\hat{y}(\mathbf{x}|\Theta), y)$
          $v_{i,f} \leftarrow v_{i,f} - \alpha\,(2\,\lambda_f\,v_{i,f} + \partial_{i,f})$
        end for
      end for
      $(\mathbf{x}', y') \sim S_V$ ▷ draw a case from valid. set
      $\lambda_0 \leftarrow \max\left(0, \lambda_0 - \alpha \frac{\partial}{\partial \lambda_0} l\left(\hat{y}(\mathbf{x}'|\tilde{\Theta}), y'\right)\right)$
      $\lambda_w \leftarrow \max\left(0, \lambda_w - \alpha \frac{\partial}{\partial \lambda_w} l\left(\hat{y}(\mathbf{x}'|\tilde{\Theta}), y'\right)\right)$
      for $f \in \{1, \ldots, k\}$ **do**
        $\lambda_f \leftarrow \max\left(0, \lambda_f - \alpha \frac{\partial}{\partial \lambda_f} l\left(\hat{y}(\mathbf{x}'|\tilde{\Theta}), y'\right)\right)$
      end for
    end for
  until stopping criterion is met
  return $\Theta := (w_0, \mathbf{w}, \mathbf{V})$
end procedure

Block Structure for Relational Data

- Improve repetitive computations and storages

(a) Predictive function

score : UserID × MovieID × Date × UserGender × UserAge × MovieGenres × UserFriends × ItemsWatched → $\mathbb{R}$

(b) Training Data for Predictive Function

UserID	*MovieID*	*Date*	*Gender*	*Age*	*Genres*	*Friends*	*ItemsWatched*	*Score*
Alice	TI	2012-09-01	F	30	{A,R}	{E,C}	{TI,NH,SW}	5
Alice	NH	2012-09-12	F	30	{C,R}	{E,C}	{TI,NH,SW}	3
Alice	SW	2012-09-15	F	30	{S,A}	{E,C}	{TI,NH,SW}	1
Bob	SW	2012-09-02	M	25	{S,A}	{C,D}	{SW,ST}	4
Bob	ST	2012-10-07	M	25	{S}	{C,D}	{SW,ST}	5
Charlie	TI	2012-09-05	M	28	{A,R}	{A,B,D}	{TI,SW}	1
Charlie	SW	2012-09-05	M	28	{S,A}	{A,B,D}	{TI,SW}	5
...	...	...	...	...	...	...	...	...

(c) Training Data in Numeric Format (*Design Matrix*)

UserID	*MovieID*	*Date*	*Gender*	*Age*	*Genres*	*Friends*	*ItemsWatched*	*Score*
1 0 0	1 0 0 0	1	1 0	30	.5 .5 0 0	0 0 .5 0 .5	.3 .3 .3 0	5
1 0 0	0 1 0 0	12	1 0	30	0 .5 .5 0	0 0 .5 0 .5	.3 .3 .3 0	3
1 0 0	0 0 1 0	15	1 0	30	.5 0 0 .5	0 0 .5 0 .5	.3 .3 .3 0	1
0 1 0	0 0 1 0	2	0 1	25	.5 0 0 .5	0 0 .5 .5 0	0 0 .5 .5	4
0 1 0	0 0 0 1	37	0 1	25	0 0 0 1	0 0 .5 .5 0	0 0 .5 .5	5
0 0 1	1 0 0 0	5	0 1	28	.5 .5 0 0	.3 .3 0 .3 0	.5 0 .5 0	1
0 0 1	0 0 1 0	5	0 1	28	.5 0 0 .5	.3 .3 0 .3 0	.5 0 .5 0	5
...	...	...	...	...	...	...	...	...
A B C	*TI NH SW ST*		*F M*		*A R C S*	*A B C D E*	*TI NH SW ST*	

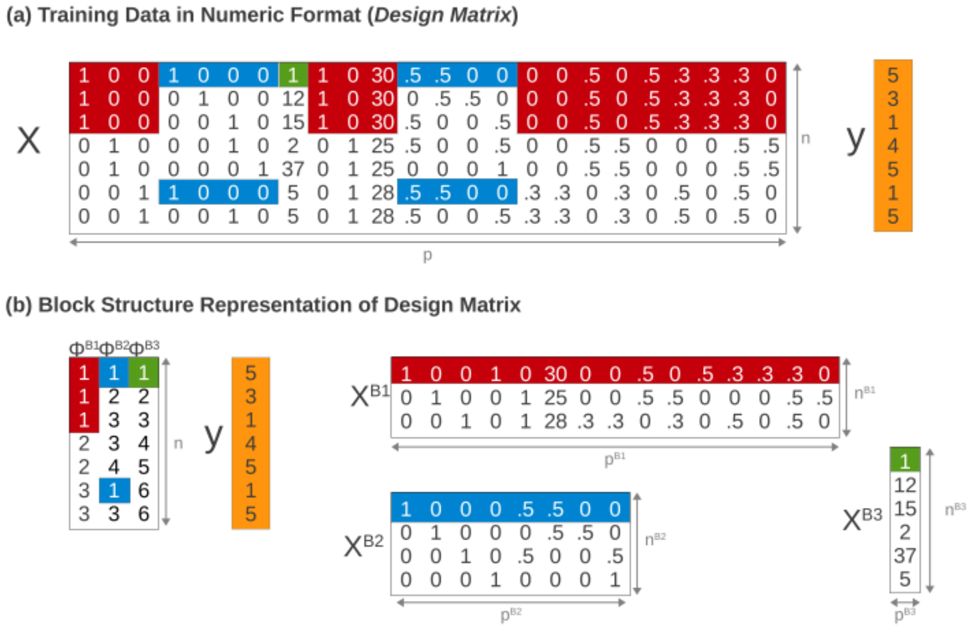
corresponding levels of categorical variables

Block Structure for Relational Data (续)

- Straight approach
  - Compress attributes

Definition 1 (Block Structure (BS)). *Let $\mathcal{B} = \{B_1, B_2, \ldots\}$ be a set of blocks, where each block $B = (X^B, \phi^B)$ consists of a design matrix $X^B \in \mathbb{R}^{n^B \times p^B}$ and a mapping $\phi^B : \{1, \ldots, n\} \to \{1, \ldots, n^B\}$ from rows in the original design matrix X to rows within X^B. $\mathcal{B}$ is a block structure representation of X iff for all rows i:*

$$\mathbf{x}_i \equiv (x^{B_1}_{\phi^{B_1}(i),1}, x^{B_1}_{\phi^{B_1}(i),2}, \ldots, x^{B_2}_{\phi^{B_2}(i),1}, x^{B_2}_{\phi^{B_2}(i),2}, \ldots) \quad (1)$$



Block Structure for Relational Data (续)

- Coordinate descent on 0th and 1st order parameters

$$\hat{y}(\mathbf{x}_i) = w_0 + \sum_{j=1}^{p} w_j \, x_{i,j}$$

$$w_l \leftarrow \frac{w_l \sum_{i=1}^{n} x_{i,l}^2 + \sum_{i=1}^{n} x_{i,l} \, e_i}{\sum_{i=1}^{n} x_{i,l}^2 + \lambda_l}$$

- Need to compute when learning parameters

$$\sum_{i=1}^{n} x_{i,l}^2, \qquad \sum_{i=1}^{n} x_{i,l} \, e_i.$$

- After applying Bayesian prioris

$$w_l \sim \mathcal{N}\left(\frac{\alpha \, w_l \sum_{i=1}^{n} x_{i,l}^2 + \alpha \sum_{i=1}^{n} x_{i,l} \, e_i + \mu_l \, \lambda_l}{\alpha \sum_{i=1}^{n} x_{i,l}^2 + \lambda_l}, \quad \frac{1}{\alpha \sum_{i=1}^{n} x_{i,l}^2 + \lambda_l} \right)$$

Block Structure for Relational Data (续)

- Scaling to Block Structures
  - Prediction

$$\hat{y}(\mathbf{x}_i) = w_0 + \sum_{B \in \mathcal{B}} \sum_{j=1}^{p^B} w_j^B \, x_{\phi(i),j}^B = w_0 + \sum_{B \in \mathcal{B}} q_{\phi(i)}^B$$

$$q_i^B = \sum_{j=1} w_j^B \, x_{i,j}^B, \quad \forall i \in \{1, \ldots, n^B\}$$

  - Learning

$$\sum_{i=1}^{n} x_{i,l}^2 = \sum_{i=1}^{n^B} \sum_{j=1}^{n} \delta(\phi^B(j) = i) x_{j,l}^2 = \sum_{i=1}^{n^B} (x_{i,l}^B)^2 \#_i^B$$

$$\#_i^B = \sum^{n} \delta(\phi^B(j) = i)$$

$$\sum_{i=1}^{n} x_{i,l} \, e_i = \sum_{i=1}^{n^B} x_{i,l}^B e_i^B, \quad e_i^B := \sum_{j=1}^{n} \delta(\phi^B(j) = i) \, e_j.$$

- Similar skills to the 2nd order parameters
  - But more complex

Part IV

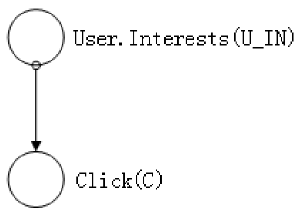
特征技术
——历史特征

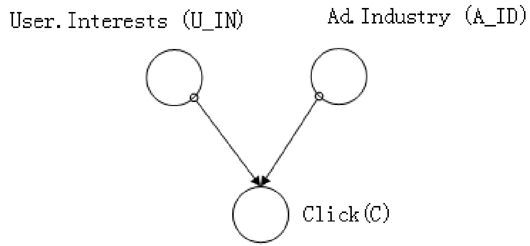
历史特征的定义

- 历史特征是一种动态特征
  - 用某一特征或特征组合的历史点击率作为新的特征
  - 动态：同一样本在不同时间的历史特征值是不同的
- 优点
  - 本身包含了对点击率预测的决策信息
  - 用来替换原有特征时可以有效降低特征维度

历史特征举例

- 以下图为例
  - 左边是基于用户兴趣到点击行为的转移概率（用户兴趣->点击），右边是基于用户兴趣和广告行业组合到点击行为的转移概率（用户兴趣×广告行业->点击）





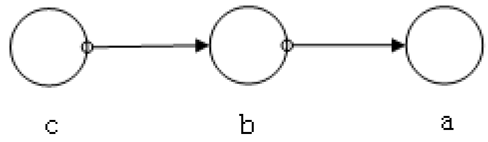
- 将转移概率作为新的特征
  - 式子中的第一项意味着在特征或特征组合条件下的点击率，可以通过统计最近历史日志获得；其他几项的信息直接包含在样本中

$$F_{1,U\_IN} = \sum_{U\_IN} P(C|U\_IN)P(U\_IN|U)$$

$$F_{1,U\_IN\times A\_ID} = \sum_{U\_IN\times A\_ID} P(C|U\_IN \times A\_ID)P(U\_IN|U)P(A\_ID|A)$$

附：最简单的概率图模型推导

单特征



上图可以如下推导

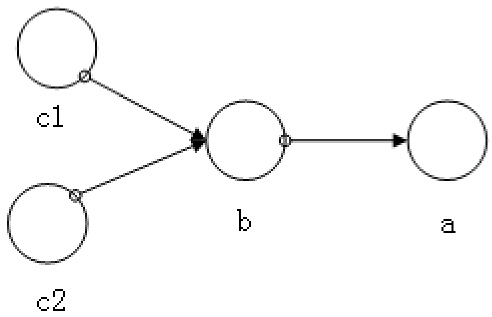
$$P(a|c)=\frac{P(ac)}{P(c)}$$

$$=\frac{\sum_b P(abc)}{P(c)}$$

$$=\frac{\sum_b P(a|bc)P(b|c)P(c)}{P(c)}$$

$$=\sum_b P(a|b)P(b|c)$$

特征组合



因为c1和c2条件独立，所以可以如下推导

$$P(a|c1,c2)=\sum_b P(a|b)P(b|c1,c2)$$

$$=\sum_b P(a|b)P(b|c1)P(b|c2)$$

历史特征的扩展

- 重写第一项
  - 分子表示在特征条件下的点击数，分母表示特征条件下的曝光数

$$P(C|U\_IN) = \frac{N_C}{N_V} \longrightarrow F_{1,U\_IN} = \sum_{U\_IN} \frac{N_C}{N_V} \cdot M$$

- 当只关注点击数时，可以延伸出下面的特征

$$F_{2,U\_IN} = \sum_{U\_IN} N_C \cdot M$$

- 当只关注曝光数时，可以延伸出下面的特征

$$F_{3,U\_IN} = \sum_{U\_IN} N_V \cdot M$$

- 当关注平均点击率时，可以延伸出下面的特征

$$F_{4,U\_IN} = \frac{\sum_{U\_IN} N_C \cdot M}{\sum_{U_IN} N_V \cdot M}$$

时间衰减

- 以当天为基准，对于之前的第t天
  - 统计该天的历史行为数据，可以对每个特征条件都计算出

$$N_C^{(t)} 、 N_V^{(t)}$$

- 继续做近似，进而计算出所有的 $F_*^{(t)}$

$$M^{(t)} \approx M^{(0)}$$

- 实际训练和预测用的特征由时间衰减计算得到

$$F_* = \sum_{t=0}^{\infty} \frac{F(t)}{2^t}$$

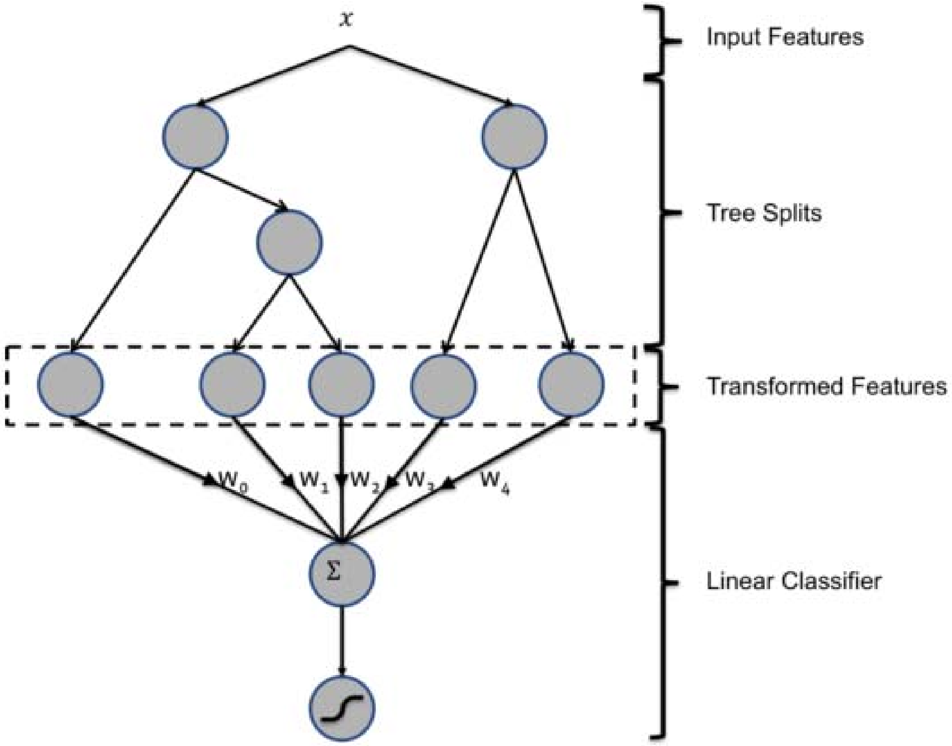
Part V

特征技术

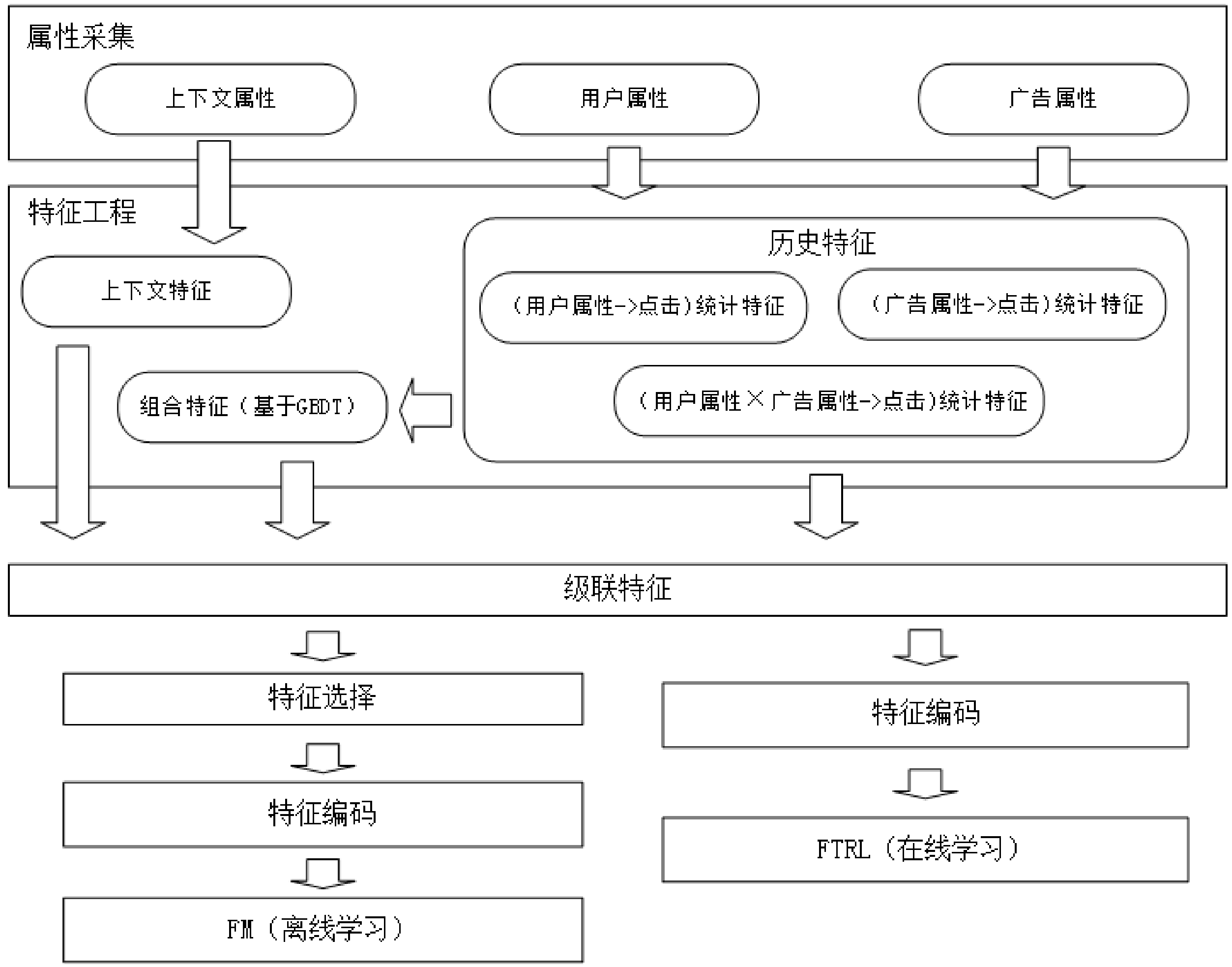
——GBDT特征

GBDT特征

- 两类特征组合方法
  - Tuple transformation
  - Non-linear transformation
- GBDT实现特征转换
  - 历史特征 → (e_{i1}, …,e_{in})
  - 例如：e_{i1}=[0,1,0], e_{i2}=[1,0]
  - e表示一条路径(一种规则)



pCTR的特征转换流程



谢谢！