CTR预估技术介绍

Fred

http://www.julyedu.com/

写在最前面

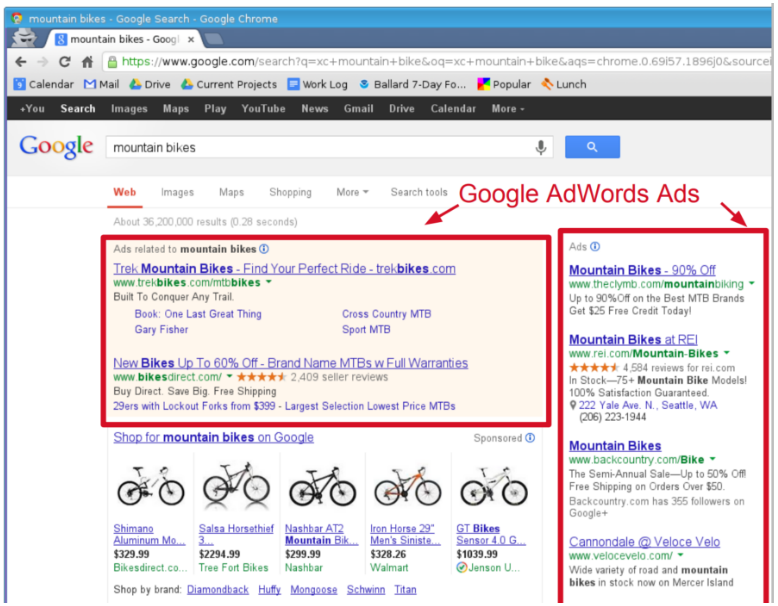
- 整体介绍和梳理
- 大量的参考资料
- 典型的工具和例子
- 没有特别细节的代码和原理讲解
- 算法架构没有详细介绍

目录

- 背景：什么是CTR预估
- CTR常见应用产品与场景
- 机器学习经典Formulation之一
- 常用算法及发展过程
- 常用工具
- 经典比赛

背景：什么是CTR预估

- CTR（Click-Through-Rate）为点击率，起源自互联网广告，有人将它称作镶嵌在互联网技术上的明珠[1]
- charge = pv * cpm, cpm = sum(ctr*bid)[3]
- 如右图[2]
- 为什么要预估CTR
    - 排序
    - 最大化后验点击率
    - 最大化Revenue



CTR常见应用产品与场景

- 最典型的场景：广告和推荐
    - 百度/Google的搜索广告
    - 阿里妈妈广告
    - 今日头条的信息流
- 其他
    - 任何0/1分类问题，比如：

机器学习经典Formulation之一

- 二分类问题
- label： y为0/1
- 特征： X
- 假设H： p(x) = H(X)
- Loss = -y log(p) - (1 - y) log(1 - p)
- 评估：
    - offline: AUC/MAPE
    - online: 业务指标

常用算法及发展过程

- LR
- FM (Google)
- FFM (Criteo)
- FTRL (Google)
- GBDT
- Wide & Deep(Google)
- GBDT+LR(Facebook)

LR：逻辑回归

- 假设H: p(Y=1|x) = sigmod(-w*x)
  - 为什么逻辑回归要用Sigmoid函数[4]
- Loss = -y log(p) - (1 - y) log(1 - p)
- 训练算法：
  - 凸优化：LBFGS/OWLQN/SGD, Batch Learning
- Overfitting
  - 正则化：L1/L2

LR

- 优点
  - 可解释性强
  - 大规模分布式实现容易
  - online更新
- 缺点
  - 模型表征能力有限

FM/FFM

- 假设H

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

$$y(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_{i,f_j}, \mathbf{v}_{j,f_i} \rangle x_i x_j$$

- 深入FFM原理与实践[5]
- 目的
  - FM: 在解决稀疏数据下的特征组合问题
  - FFM: 通过引入field的概念，FFM把相同性质的特征归于同一个field。

FTRL[6]

- Batch Learning
    - 系统无法进行增量学习——即必须使用所有可用数据进行训练。这需要大量时间和计算资源，所以通常情形下，都是离线完成的
- Online Learning
    - 可以循序渐进地给系统提供训练数据，逐步积累学习成果。这种提供数据的方式可以是单独地，也可以采用小批量（mini-batches）的小组数据来进行训练。每一步学习都很快速并且便宜，所以系统就可以根据飞速写入的最新数据进行学习

FTRL

- Online Learning能否得到全局最优解?
- Regret

$$R(T) = \sum_{t=1}^{T} f_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^{T} f_t(w).$$

- online learning https://courses.cs.washington.edu/courses/cse599s/14sp/

Algorithm 1 Per-Coordinate FTRL-Proximal with L_1 and L_2 Regularization for Logistic Regression

 # With per-coordinate learning rates of Eq. (2).
Input: parameters α, β, λ_1, λ_2
$(\forall i \in \{1, \dots, d\})$, initialize $z_i = 0$ and $n_i = 0$
for $t = 1$ **to** T **do**
  Receive feature vector $\mathbf{x}_t$ and let $I = \{i \mid x_i \neq 0\}$
  For $i \in I$ compute

$$w_{t,i} = \begin{cases} 0 & \text{if } |z_i| \leq \lambda_1 \\ -\left(\frac{\beta + \sqrt{n_i}}{\alpha} + \lambda_2\right)^{-1} (z_i - \operatorname{sgn}(z_i)\lambda_1) & \text{otherwise.} \end{cases}$$

  Predict $p_t = \sigma(\mathbf{x}_t \cdot \mathbf{w})$ using the $w_{t,i}$ computed above
  Observe label $y_t \in \{0, 1\}$
  for all $i \in I$ **do**
    $g_i = (p_t - y_t)x_i$ *#gradient of loss w.r.t.* w_i
    $\sigma_i = \frac{1}{\alpha}\left(\sqrt{n_i + g_i^2} - \sqrt{n_i}\right)$ *#equals* $\frac{1}{\eta_{t,i}} - \frac{1}{\eta_{t-1,i}}$
    $z_i \leftarrow z_i + g_i - \sigma_i w_{t,i}$
    $n_i \leftarrow n_i + g_i^2$
  end for
end for

GBDT

- 树模型+Ensemble
- 多个弱分类器
- GBDT与XGBoost的区别[7]
- 优点
  - 连续值统计特征
  - 模型比较鲁棒，但是参数不合理容易过拟合
  - 数据量较少时效果比LR/FM好
- 缺点
  - 树结构无法做online更新
  - 超大规模分布式实现比较困难

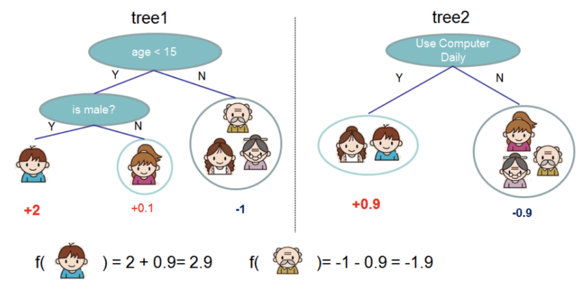
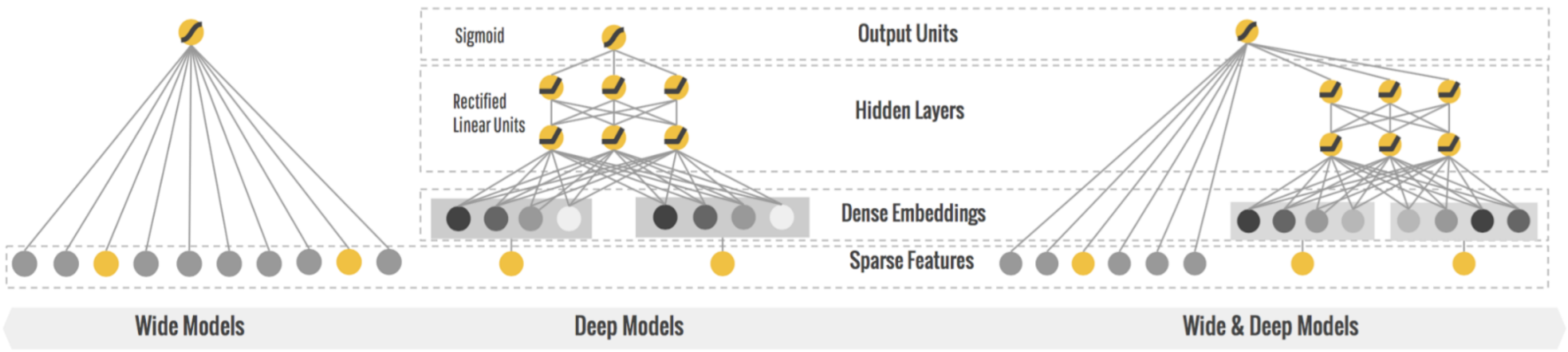


Figure 1: Tree Ensemble Model. The final prediction for a given example is the sum of predictions from each tree.

Wide & Deep

- 很长一段时间，LR统治工业界的CTR预估
- Wide & Deep Learning: Better Together with TensorFlow[8]



Wide & Deep

- Memorization
  - Wide
- Generalization
  - Deep
- Explore & Exploit
  - MAB

Table 1: Offline & online metrics of different models. Online Acquisition Gain is relative to the control.

Model	Offline AUC	Online Acquisition Gain
Wide (control)	0.726	0%
Deep	0.722	+2.9%
Wide & Deep	0.728	+3.9%

GBDT + LR

- 连续变量切分点如何选取
- 离散化为多少份合理
- 选择哪些特征交叉
- 多少阶交叉，二阶，三阶或更多

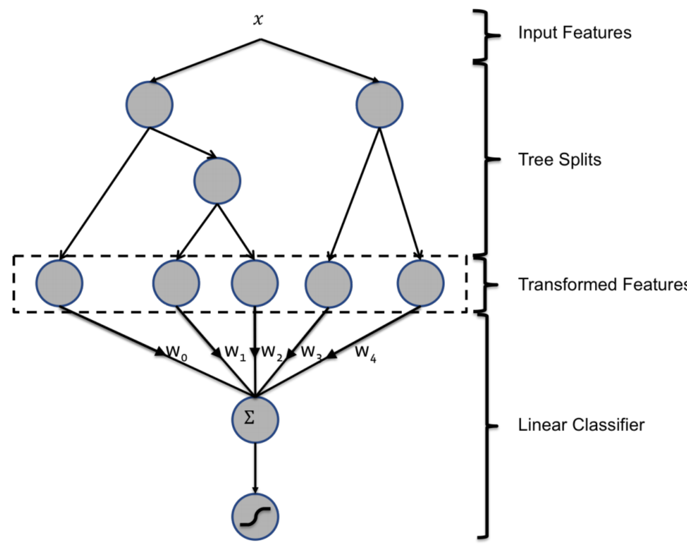


Figure 1: Hybrid model structure. Input features are transformed by means of boosted decision trees. The output of each individual tree is treated as a categorical input feature to a sparse linear classifier. Boosted decision trees prove to be very powerful feature transforms.

常用工具

- LR/FM/FFM
  - liblinear https://www.csie.ntu.edu.tw/~cjlin/liblinear/
  - Spark MLlib
    - https://spark.apache.org/docs/2.3.0/mllib-optimization.html
    - LBFGS/SGD
  - https://github.com/ycjuan/libffm
  - https://github.com/dmlc/difacto

常用工具

- GBDT
    - XGBoost, https://xgboost.readthedocs.io/en/latest/
    - LightGBM,https://lightgbm.readthedocs.io/en/latest/

常用工具

- Wide & Deep
    - TensorFlow教程[10]
    - Google Colab https://colab.research.google.com/
- DeepCTR https://github.com/shenweichen/DeepCTR
- 由此可见这个问题多火

Models List

Model	Paper
Factorization-supported Neural Network	[ECIR 2016]Deep Learning over Multi-field Categorical Data: A Case Study on User Response Prediction
Product-based Neural Network	[ICDM 2016]Product-based neural networks for user response prediction
Wide & Deep	[DLRS 2016]Wide & Deep Learning for Recommender Systems
DeepFM	[IJCAI 2017]DeepFM: A Factorization-Machine based Neural Network for CTR Prediction
Piece-wise Linear Model	[arxiv 2017]Learning Piece-wise Linear Models from Large Scale Data for Ad Click Prediction
Deep & Cross Network	[ADKDD 2017]Deep & Cross Network for Ad Click Predictions
Attentional Factorization Machine	[IJCAI 2017]Attentional Factorization Machines: Learning the Weight of Feature Interactions via Attention Networks
Neural Factorization Machine	[SIGIR 2017]Neural Factorization Machines for Sparse Predictive Analytics
Deep Interest Network	[KDD 2018]Deep Interest Network for Click-Through Rate Prediction
Deep Interest Evolution Network	[arxiv 2018]Deep Interest Evolution Network for Click-Through Rate Prediction
xDeepFM	[KDD 2018]xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems

经典比赛

- Kaggle Click-Through Rate Prediction
  https://www.kaggle.com/c/avazu-ctr-prediction
- FFM winner
  - https://github.com/ycjuan/kaggle-avazu
  - 特征工程
- 非常好的练习题
  - 先复现冠军的解法
  - 尝试GBDT，Wide & Deep的算法

Ref

[1] 镶嵌在互联网技术上的明珠：漫谈深度学习时代点击率预估技术进展,https://zhuanlan.zhihu.com/p/54822778

[2] Ad Click Prediction: a View from the Trenches,https://courses.cs.washington.edu/courses/cse599s/14sp/kdd_2013_talk.pdf

[3] 机器知道你会点广告：写给普通人的CTR预估科普, https://baijiahao.baidu.com/s?id=1610035181636775323&wfr=spider&for=pc

[4] 为什么逻辑回归要用sigmoid 函数？ ,https://ask.julyedu.com/question/85100

[5] 深入FFM原理与实践, https://tech.meituan.com/2016/03/03/deep-understanding-of-ffm-principles-and-practices.html

[6] Mcmahan H B, Holt G, Sculley D, et al. Ad click prediction: a view from the trenches[C], KDD 2013.

[7] 机器学习算法中 GBDT 和 XGBOOST 的区别有哪些？,https://www.zhihu.com/question/41354392

[8] Wide & Deep Learning: Better Together with TensorFlow, https://ai.googleblog.com/2016/06/wide-deep-learning-better-together-with.html

[9] Practical Lessons from Predicting Clicks on Ads at Facebook, https://research.fb.com/publications/practical-lessons-from-predicting-clicks-on-ads-at-facebook/

[10] Predicting Income with the Census Income Dataset, https://github.com/tensorflow/models/tree/master/official/wide_deep