Kaggle and
Click-Through Rate Prediction

Dr. Todd W. Neller
Professor of Computer Science
TruWood
Sponsored •

Start 2019 with a win! Access our exclusive New Year Sale today! 🎄

Use the code NEWYEAR15 at checkout for an additional 15% OFF!

Shop Now -» www.mytruwood.com

vimeo

Tons and tons of video storage

vimeo.com

Your videos deserve more space. Keep rough cuts, final edits, and everything in between.

Flintstones Vitamins
flintstonesvitamins.com

Flintstones Complete Gummies are the #1 pediatrician recommended children’s vitamin brand,...
Click-Through Rate (CTR) Prediction

\[
CTR = \frac{\text{Number of click-throughs}}{\text{Number of impressions}} \times 100(\%)
\]

- Number of impressions = number of times an advertisement was served/offered
- Given: much data on past link offerings and whether or not users clicked on those links
- Predict: the probability that a current user will click on a given link
Example Data on Past Link Offerings

- **User data:**
  - User ID from site login, cookie
  - User IP address, IP address location

- **Link context data:**
  - Site ID, page ID, prior page(s)
  - Time, date

- **Link data:**
  - Link ID, keywords
  - Position offered on page
Example: Facebook Information

Access Your Information

Here is a list of your Facebook information that you can access at any time. We’ve categorized this information by type so you can easily find what you’re looking for. Our Data Policy has more information about how we collect and use your information, how it’s shared and how long we retain it. It also outlines your rights and how you can exercise them, and how we operate and transfer your information as part of our global services.

You can choose to download your information if you’d like a copy of it.

Your information

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Posts</td>
<td>Posts you've shared on Facebook and posts you've been tagged in</td>
</tr>
<tr>
<td>Photos and Videos</td>
<td>Photos and videos you've shared or been tagged in</td>
</tr>
<tr>
<td>Comments</td>
<td>Comments you've posted on your own posts, on other people's posts or in groups you belong to</td>
</tr>
<tr>
<td>Likes and Reactions</td>
<td>Posts, comments and Pages you've liked or reacted to</td>
</tr>
<tr>
<td>Friends</td>
<td>The people you are connected to on Facebook</td>
</tr>
<tr>
<td>Following and Followers</td>
<td>People, organizations or business you choose to see content from, and people who follow you</td>
</tr>
<tr>
<td>Messages</td>
<td>Messages you've exchanged with other people on Messenger</td>
</tr>
<tr>
<td>Groups</td>
<td>Groups you belong to, groups you manage and your posts and comments within the groups you belong to</td>
</tr>
<tr>
<td>Events</td>
<td>Your responses to events and a list of the events you’ve created</td>
</tr>
<tr>
<td>Profile Information</td>
<td>Your contact information, information you've written in your About You section in your profile</td>
</tr>
</tbody>
</table>
Why is CTR Prediction Important?

• Advertising Industry View:
  – Much of online advertising is billed using a pay-per-click model.
ATTENTION IS THE CURRENCY OF THE INTERNET

Traditional Economy → Scarce Resources
Information Economy → Unlimited Resources → Limited Time

Benefits Beyond Advertising

• **Herbert Simon**, 1971:
  – “In an information-rich world, the wealth of information means a dearth of something else: a scarcity of whatever it is that information consumes. What information consumes is rather obvious: the attention of its recipients.”

• Better CTR prediction → more relevance → better use of scarce time
Outline

• Click-Through Rate Prediction (CTRP) Introduction
• Kaggle
  – Learning community offerings incentives
  – CTRP Competitions
• Feature Engineering
  – Numbers, Categories, and Missing Values
• favored regression techniques for CTRP
  – Logistic Regression
  – Gradient Boosted Decision Trees (e.g. xgBoost)
  – Field-aware Factorization Machines (FFMs)
• Future Recommendations
What is Kaggle.com?

• Data Science and Machine Learning Community featuring
  – Competitions → $\$, peer learning, experience, portfolio
  – Datasets
  – Kernels
  – Discussions
  – Tutorials (“Courses”)
  – Etc.

• Status incentives
<table>
<thead>
<tr>
<th>Competition</th>
<th>Prize</th>
<th>Teams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two Sigma: Using News to Predict Stock Movements</td>
<td>$100,000</td>
<td>2,897 teams</td>
</tr>
<tr>
<td>Use news analytics to predict stock price performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Featured · Kernels Competition · 5 months to go · news agencies, time series, finance, money</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LANL Earthquake Prediction</td>
<td>$50,000</td>
<td>938 teams</td>
</tr>
<tr>
<td>Can you predict upcoming laboratory earthquakes?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research · 4 months to go · earth sciences, physics, signal processing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elo Merchant Category Recommendation</td>
<td>$50,000</td>
<td>3,487 teams</td>
</tr>
<tr>
<td>Help understand customer loyalty</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Featured · 20 days to go · tabular data, banking, regression</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Google Analytics Customer Revenue Prediction</td>
<td>$45,000</td>
<td>1,104 teams</td>
</tr>
<tr>
<td>Predict how much GStore customers will spend</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Featured · 9 days to go · regression, tabular data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gendered Pronoun Resolution</td>
<td>$25,000</td>
<td>30 teams</td>
</tr>
<tr>
<td>Pair pronouns to their correct entities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Research · 3 months to go · nlp, text data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PetFinder.my Adoption Prediction</td>
<td>$25,000</td>
<td>972 teams</td>
</tr>
<tr>
<td>How cute is that doggy in the shelter?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>Dataset Name</td>
<td>Description</td>
</tr>
<tr>
<td>---</td>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>150</td>
<td>Heart Disease UCI</td>
<td><a href="https://archive.ics.uci.edu/ml/datasets/Heart+Disease">https://archive.ics.uci.edu/ml/datasets/Heart+Disease</a></td>
</tr>
<tr>
<td>441</td>
<td>Graduate Admissions</td>
<td>Predicting admission from important parameters</td>
</tr>
<tr>
<td>48</td>
<td>Los Angeles Parking Citations</td>
<td>From Los Angeles Open Data</td>
</tr>
<tr>
<td>351</td>
<td>FIFA 19 complete player dataset</td>
<td>18k+ FIFA 19 players, ~90 attributes extracted from the latest FIFA database</td>
</tr>
<tr>
<td>907</td>
<td>Google Play Store Apps</td>
<td>Web scraped data of 10k Play Store apps for analysing the Android market.</td>
</tr>
</tbody>
</table>
Kernels

• Jupyter notebooks of mixed text and Python/R
  – Interleaved explanations and free runnable code
• E.g. [https://www.kaggle.com/mjbahmani/a-comprehensive-ml-workflow-with-python](https://www.kaggle.com/mjbahmani/a-comprehensive-ml-workflow-with-python)
Discussions

Display Advertising Challenge

Predict click-through rates on display ads
$16,000 · 718 teams · 4 years ago

Overview Data Kernels Discussion Leaderboard Rules Team My Submissions New Topic

81 topics Follow

Sort by Hotness

All Mine Upvoted

13 Document and code for the 3rd place finish
Guocong Song 4 years ago

last comment by
Vishal Gupta 3mo ago

59 3 Idiots’ Solution & LIBFFM
Yuchin Juan 4 years ago

last comment by
6mo ago

7 Data Release After Competition Ends
joycev 4 years ago

last comment by
Olivier Chapelle 3y ago

9 Congratulations to the winners!
Dylan Friedmann 4 years ago

last comment by
marbel 3y ago
Tutorials

Machine Learning
Machine learning is the hottest field in data science, and this track will get you started quickly.

Overview
- Free Course
- 4 hrs.
- 19 Lessons

Prerequisite Skills:
Python

Prepares you for these Learn Courses:
Deep Learning,
Machine Learning
Explainability

Instructor
Dan Becker
Data Scientist

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. How Models Work</td>
<td></td>
</tr>
<tr>
<td>The first step if you’re new to machine learning</td>
<td></td>
</tr>
<tr>
<td>2. Explore Your Data</td>
<td></td>
</tr>
<tr>
<td>Load data and set up your environment for your hands-on project</td>
<td></td>
</tr>
<tr>
<td>3. Exercise: Explore Your Data</td>
<td></td>
</tr>
<tr>
<td>4. Your First Machine Learning Model</td>
<td></td>
</tr>
<tr>
<td>Building your first model, Hurray!</td>
<td></td>
</tr>
<tr>
<td>5. Exercise: Your First Machine Learning Model</td>
<td></td>
</tr>
</tbody>
</table>
Status Incentives

Michael Jahrer

Graz, Austria
Joined 9 years ago · last seen in the past day
http://www.operasolutions.com/

Competitions Grandmaster
- Current Rank: 18 of 96,011
- Highest Rank: 6

Home Credit Default Risk
- 5 months ago · Top 1%
- 1st of 7198

Porto Seguro’s Safe Driver
- a year ago · Top 1%
- 1st of 5169

Cervical Cancer Screening
- 3 years ago · Top 3%
- 1st of 40

Kernels Contributor
- Unranked

XGB Feature Importance (P... 3 years ago
- 5 votes

naive XGB
- 2 years ago
- 4 votes

Simple LightGBM 4 ! 2 years ago
- 2 votes

Discussion Expert
- Current Rank: 32 of 82,748
- Highest Rank: 9

1st place with representation... a year ago
- 704 votes

Is 0.288 magical and optim... a year ago
- 61 votes

you were only supposed to ... 2 years ago
- 47 votes
Kaggle CTRP Competitions

Display Advertising Challenge
Predict click-through rates on display ads
Research · 4 years ago

Avito Context Ad Clicks
Predict if context ads will earn a user's click
Featured · 4 years ago · marketing, tabular data, click prediction

Outbrain Click Prediction
Can you predict which recommended content each user will click?
Featured · 2 years ago · internet, tabular data, click prediction

Click-Through Rate Prediction
Predict whether a mobile ad will be clicked
Featured · 4 years ago
Criteo Display Advertising Challenge

Display Advertising Challenge

criteolabs

Predict click-through rates on display ads

$16,000 · 718 teams · 4 years ago

Overview Data Discussion Leaderboard Rules

Overview

Description Evaluation Prizes About Criteo Timeline Winners

Display advertising is a billion dollar effort and one of the central uses of machine learning on the Internet. However, its data and methods are usually kept under lock and key. In this research competition, CriteoLabs is sharing a week’s worth of data for you to develop models predicting ad click-through rate (CTR). Given a user and the page he is visiting, what is the probability that he will click on a given ad?

The goal of this challenge is to benchmark the most accurate ML algorithms for CTR estimation. All winning models will be released under an open source license. As a participant, you are given a chance to access the traffic logs from Criteo that include various undisclosed features along with the click labels.

https://www.kaggle.com/c/criteo-display-ad-challenge
Criteo Display Advertising Challenge Data:

- Features (inputs):
  - 13 numeric: unknown meanings, mostly counts, power laws evident
  - 26 categorical: unknown meanings, hashed (encoding without decoding), few dominant, many unique

- Target (output): 0 / 1 (didn’t / did click through)
## Mysterious Data

<table>
<thead>
<tr>
<th>Label</th>
<th>L1</th>
<th>L2</th>
<th>...</th>
<th>L13</th>
<th>C1</th>
<th>C2</th>
<th>...</th>
<th>C26</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>20</td>
<td></td>
<td>2741</td>
<td>68fd1e64</td>
<td>80e26c9b</td>
<td></td>
<td>4cf72387</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>91</td>
<td></td>
<td>1157</td>
<td>3516f6e6</td>
<td>cfc86806</td>
<td></td>
<td>796a1a2e</td>
</tr>
<tr>
<td>0</td>
<td>12</td>
<td>73</td>
<td></td>
<td>1844</td>
<td>05db9164</td>
<td>38a947a1</td>
<td></td>
<td>5d93f8ab</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>9</td>
<td>62</td>
<td></td>
<td>1457</td>
<td>68fd1e64</td>
<td>cfc86806</td>
<td></td>
<td>cf59444f</td>
</tr>
</tbody>
</table>

#Train: \( \approx 45\text{M} \)

#Test: \( \approx 6\text{M} \)

### Mysterious Data

**Unknown Labels: meanings of numbers and categories not given**

<table>
<thead>
<tr>
<th>Label</th>
<th>I1</th>
<th>I2</th>
<th>⋯</th>
<th>I13</th>
<th>C1</th>
<th>C2</th>
<th>⋯</th>
<th>C26</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>20</td>
<td>⋯</td>
<td>2741</td>
<td>68fd1e64</td>
<td>80e26c9b</td>
<td>⋯</td>
<td>4cf72387</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>91</td>
<td>⋯</td>
<td>1157</td>
<td>3516f6e6</td>
<td>cfc86806</td>
<td>⋯</td>
<td>796a1a2e</td>
</tr>
<tr>
<td>0</td>
<td>12</td>
<td>73</td>
<td>⋯</td>
<td>1844</td>
<td>05db9164</td>
<td>38a947a1</td>
<td>⋯</td>
<td>5d93f8ab</td>
</tr>
<tr>
<td>⋮</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>9</td>
<td>62</td>
<td>⋯</td>
<td>1457</td>
<td>68fd1e64</td>
<td>cfc86806</td>
<td>⋯</td>
<td>cf59444f</td>
</tr>
</tbody>
</table>

**#Train:**

\[ \approx 45M \]

**#Test:**

\[ \approx 6M \]

Mysterious Data

<table>
<thead>
<tr>
<th>Label</th>
<th>I1</th>
<th>I2</th>
<th>...</th>
<th>I13</th>
<th>C1</th>
<th>C2</th>
<th>...</th>
<th>C26</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>20</td>
<td></td>
<td></td>
<td>68fd1e64</td>
<td>80e26c9b</td>
<td></td>
<td>4cf72387</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>91</td>
<td></td>
<td></td>
<td>3516f6e6</td>
<td>cfc86806</td>
<td></td>
<td>796a1a2e</td>
</tr>
<tr>
<td>0</td>
<td>12</td>
<td>73</td>
<td></td>
<td></td>
<td>05db9164</td>
<td>38a947a1</td>
<td></td>
<td>5d93f8ab</td>
</tr>
<tr>
<td>?</td>
<td>9</td>
<td>62</td>
<td></td>
<td></td>
<td>68fd1e64</td>
<td></td>
<td></td>
<td>cf59444f</td>
</tr>
</tbody>
</table>

Categorical data is *hashed*.

#Train:
- \( \approx 45M \)

#Test:
- \( \approx 6M \)

Hashing

- A **hash** function takes some data and maps it to a number.
- Example: URL (web address)
  - Representation: string of characters
  - Character representation: a number (Unicode value)
  - Start with value 0.
  - Repeat for each character:
    - Multiply value by 31
    - Add next character Unicode to value
  - Don’t worry about overflow – it’s just a consistent “mathematical blender”.

“Hi”

\[ H = 72, \ i = 105 \]
\[ 31 \ast 72 + 105 = 2337 \]
Hash Function Characteristics

• **Mapping**: same input $\rightarrow$ same output

• **Uniform**: outputs have similar probabilities
  – **Collision**: two different inputs $\rightarrow$ same output
  – Collisions are allowable (inevitable if $\text{#data} > \text{#hash values}$) but not desirable.

• **Non-invertible**: can’t get back input from output (e.g. cryptographic hashing, anonymization)
Missing Data

• The first 10 lines of the training data:

Missing numeric and categorical features:

0 1 1 5 0 1382
0 2 0 44 1 102
0 2 0 1 14 767
0 3 -1 4392
0 0 -1 12824
0 1 2 3168
1 1 4 2 0 0
0 44 4 8 19010
0 35 1 33737
0 7b5194c 0 3a171ecb 5c50484 0 0 8b83407 9727dd16
0 60f6221e 1 3a171ebc 43f13e8b 0 e 8b83407 731c3655
0 e587c466 3062eb 3a171ebc 3b1835c5
0 6b3a5ca6 3a171ebc 9117a34a
0 21c9516a 32c7478e b34f3128
0 242bb710 974f4 97c4b4 72c78f11
0 20062612 93bad2c0 1b256e61
0 5316a17f 32c7478e 9117a34a
0 0014c32a 32c7478e 3b1835c5
0 0e63fca0 32c7478e 0e8fe315
One approach to dealing with missing data is to \textit{impute} values, i.e. replace with reasonable values inferred from surrounding data.

In other words, create predictors for each value based on other known/unknown values.

\textbf{Cons:}

- Difficult to validate.
- In Criteo data, missing values are correlated.
- So ... we’re writing predictors to impute data we’re learning predictors from?
Missing Data: Embrace the “Unknown”

• Retain “unknown” as data that contains valuable information.

• Does the lack of CTR context data caused by incognito browsing mode provide information on what a person is more likely to click?

• Categorical data: For each category C# with missing data, create a new category value “C#:unknown”.
Missing Data: Embrace the “Unknown”

• Numeric data:
  – Create an additional feature that indicates whether the value for a feature is (un)known.
    • Additionally could impute mean, median, etc., for unknown value.
  – Convert to categorical and add “C#:unknown” category...
Numeric to Categorical: Binning

• Histogram-based
  – Uniform ranges: (+) simple (-) uneven distribution, poor for non-uniform data
  – Uniform ranges on transformation (e.g. log): (+) somewhat simple (-) transformation requires understanding of data distribution

• Quantiles
  – E.g. quartiles = 4-quantiles, quintiles = 5-quantiles
  – (+) simple, even distribution by definition, (-) preponderance of few values \( \rightarrow \) duplicate bins (eliminate)
Categorical to Numeric: One-Hot Encoding

• For each categorical input variable:
  – For each possible category value, create a new numeric input variable that can be assigned numeric value 1 ("belongs to this category") or 0 ("does not belong to this category").
  – For each input, replace the categorical value variable with these new numeric inputs.

Categorical to Numeric: Hashing

• When there are a large number of categories, one-hot encoding isn’t practical.
  – E.g. Criteo data category C3 in its small sample of CTR data had 10,131,226 distinct categorical values.
  – One approach (e.g. for power law data): one-hot encode few dominant values plus “rare” category.
  – Hashing trick:
    • Append category name and unusual character before category value and hash to an integer.
    • Create a one-hot-like category for each integer.
Hashing Trick Example

- From https://www.csie.ntu.edu.tw/~r01922136/kaggle-2014-criteo.pdf:

- Fundamental tradeoff: greater/lesser number hashed features results in ...
  - ... more/less expensive computation
  - ... less/more frequent hash collisions (i.e. unlike categories treated as like)
Logistic Regression Motivation

• Logistic regression is perhaps the simplest technique to beat the Criteo benchmark, scoring \(~42^{nd}\) percentile on leaderboard:
  – 100 lines of Python, 200MB RAM, 30 min. training
  – Also: logistic regression recommended for CTRP by researchers of [Criteo, Microsoft, LinkedIn, Google, and Facebook](https://www.kaggle.com/c/criteo-display-ad-challenge/discussion/10322) for practical, scalable implementation.
Example: Passing vs. Studying

Probability of passing an exam versus hours studying
Unknown Logistic Model

Probability of passing an exam versus hours studying
Misapplication of Linear Regression

Probability of passing an exam versus hours studying

Probability of passing exam

Hours studying
Logistic Regression Recovering Model

Probability of passing an exam versus hours studying

Probability of passing exam

Hours studying

0  1  2  3  4  5

0.0  0.2  0.4  0.6  0.8  1.0
Logistic Regression with Stochastic Gradient Descent

• Output: \( p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \)

• Initially: \( \beta_0 = \beta_1 = 0 \)

• Repeat:
  – For each input \( x \),
    • Adjust intercept \( \beta_0 \) by learning rate * error * \( p'(x) \)
    • Adjust coefficient \( \beta_1 \) by learning rate * error * \( p'(x) * x \)

• Note:
  – Error = \( y - p(x) \)
  – \( p'(x) = p(x) * (1 - p(x)) \) (the slope of \( p \) at \( x \))
  – This is neural network learning with a single logistic neuron with bias input of 1
Logistic Regression Takeaways

• The previous algorithm doesn’t require complex software. (12 lines raw Python code)
• Easy and effective for CTR prediction.
• Key to good performance: *skillful feature engineering of numeric features*
• Foreshadowing: Since logistic regression is a simple special case of neural network learning, I would expect deep learning tools to make future inroads here.
Maximizing Info with Decisions

• Number Guessing Game example:
  – “I’m thinking of a number from 1 to 100.”
  – What is the best strategy and why?

• Good play maximizes information according to some measure (e.g. entropy).
Decision Trees for Regression (Regression Trees)

• Numeric features (missing values permitted)
• At each node in the tree, a branch is decided on according to a features value (or lack thereof)

A regression tree estimating the probability of kyphosis (hunchback) after surgery, given the age of the patient and the vertebra at which surgery was started.

Source: https://en.wikipedia.org/wiki/Decision_tree_learning
The Power of Weak Classifiers

• Caveats:
  – Too deep: Single instance leaves $\rightarrow$ overfitting; similar to nearest neighbor (n=1)
  – Too shallow: Large hyperrectangular sets $\rightarrow$ underfitting; poor, blocky generalization

• Many weak classifiers working together can achieve good fit and generalization.
  – “Plans fail for lack of counsel, but with many advisers they succeed.” – Proverbs 15:22

• Ensemble methods: **boosting**, bagging, stacking
Gradient Boosting of Regression Trees

• Basic boosting idea:
  – Initially, make a 0 or constant prediction.
  – Repeat:
    • Compute prediction errors from the weighted sum of our weak-learner predictions.
    • Fit a new weak-learner to predict these errors and add its weighted error-prediction to our model.

• Alex Rogozhnikov’s beautiful demonstration: [https://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html](https://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html)
target function $f(x)$ and prediction of previous trees $D(x)$

residual $R(x)$ and prediction of next tree $d_n(x)$
target function $f(x)$ and prediction of previous trees $D(x)$

residual $R(x)$ and prediction of next tree $d_n(x)$

Tree depth: 3

Number of built trees: 1
target function $f(x)$ and prediction of previous trees $D(x)$

residual $R(x)$ and prediction of next tree $d_n(x)$

Tree depth: 3
Number of built trees: 2
target function $f(x)$ and prediction of previous trees $D(x)$

residual $R(x)$ and prediction of next tree $d_n(x)$

Tree depth: 3
Number of built trees: 10
XGBoost

• “Among the 29 challenge winning solutions published at Kaggle’s blog during 2015, 17 solutions [~59%] used XGBoost. Among these solutions, eight [~28%] solely used XGBoost to train the model, while most others combined XGBoost with neural nets in ensembles.” - Tianqi Chen, Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”
XGBoost Features

- **XGBoost** is a specific implementation of gradient boosted decision trees that:
  - Supports a command-line interface, C++, Python (scikit-learn), R (caret), Java/JVM languages + Hadoop platform
  - A range of computing environments with parallelization, distributed computing, etc.
  - Handles sparse, missing data
  - Is *fast* and *high-performance* across diverse problem domains
  - [https://xgboost.readthedocs.io](https://xgboost.readthedocs.io)
Field-aware Factorization Machines (FFMs)

• Top-performing technique in 3 of 4 Kaggle CTR prediction competitions plus RecSys 2015:
  – Criteo: https://www.kaggle.com/c/criteo-display-ad-challenge
  – Avazu: https://www.kaggle.com/c/avazu-ctr-prediction
  – Outbrain: https://www.kaggle.com/c/outbrain-click-prediction
What’s Different? Field-Aware Latent Factors

• Latent factor
  – learned weight; tuned variable
  – How much an input contributes to an output

• Many techniques learn “latent factors”:
  – Linear regression: one per feature + 1
    \[ Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \]
  – Logistic regression: one per feature + 1
    \[ p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}} \]
What’s Different? Field-Aware Latent Factors (cont.)

• Many techniques learn “latent factors”:
  – Degree-2 polynomial regression: one per pair of features

\[
\phi_{\text{Poly2}}(\mathbf{w}, \mathbf{x}) = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} w_{h(j_1,j_2)} x_{j_1} x_{j_2}
\]

  – Factorization machine (FM):
    • \( k \) per feature
    • “latent factor vector”, a.k.a. “latent vector”

\[
\phi_{\text{FM}}(\mathbf{w}, \mathbf{x}) = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} ( \mathbf{w}_{j_1} \cdot \mathbf{w}_{j_2} ) x_{j_1} x_{j_2}
\]
What’s Different? Field-Aware Latent Factors (cont.)

• Many techniques learn “latent factors”:
  – Field-aware Factorization machine (FFM):
    • $k$ per feature and field pair
  – Field:
    • Features are often one-hot encoded
    • Continuous block of binary features often represent different values for the same underlying “field”
    • E.g. Field: “OS”, features: “Windows”, “MacOS”, “Android”
    • libffm: FFM library (https://github.com/guestwalk/libffm)

$$\phi_{\text{FFM}}(w, x) = \sum_{j_1=1}^{n} \sum_{j_2=j_1+1}^{n} (w_{j_1,f_2} \cdot w_{j_2,f_1}) x_{j_1} x_{j_2}$$
Winning Team Process


  "nnz" means the number of non-zero elements of each impression; "feat" represents the size of feature space.
Is the Extra Engineering Worth it?

- Kaggle Criteo leaderboard based on **logarithmic loss** (a.k.a. logloss)
  - 0.69315 → 50% correct in binary classification (random guessing baseline)

- Simple logistic regression with hashing trick:
  - 0.46881 (private leaderboard) ~62.6% correct

- FFM with feature engineering using GBDT:
  - 0.44463 (private leaderboard) ~64.1% correct
Computational Cost

• ~1.5% increase in correct prediction, but greater computational complexity:
  – Logistic regression: $n$ factors to learn and relearn in dynamic context
  – FFM: $kn^2$ factors to learn and relearn
Model Ensemble for Click Prediction in Bing Search Ads

Xiaolong Ling
Microsoft
No. 5 Dan Ling Street
Beijing, China
xiaolong@live.com

Weiweng Deng
Microsoft
No. 5 Dan Ling Street
Beijing, China
ddeeng@microsoft.com

Chen Gu
Microsoft
No. 5 Dan Ling Street
Beijing, China
chengu@microsoft.com

Huchong Zhou
Microsoft Research
No. 5 Dan Ling Street
Beijing, China
hzh@live.com

Cui Li
Microsoft Research
No. 5 Dan Ling Street
Beijing, China
v-cui@live.com

Feng Sun
Microsoft
No. 5 Dan Ling Street
Beijing, China
fengsun@microsoft.com

ABSTRACT

Accurate estimation of the click-through rate (CTR) in sponsored ads significantly improves the user experience and revenue for advertisers. With larger training data, there is nearly 0.9% AUC improvement on offline training and significant click yield gains in online. In this paper, we share our experience and learnings on improving quality of training.

Keywords: click prediction, DNN, GBDT; model ensemble.

1. INTRODUCTION

Search engine advertising has become a significant revenue source for web browsing companies. Choosing the right ad for a user and the order in which they are displayed greatly affects the ability to monetize users. Accurate training the click-through rate (CTR) of ads [10, 11, 12] has a significant impact on revenue across businesses. Even a 0.1% increase in the click-through rate would yield hundreds of millions of dollars in additional earnings. An ad's CTR is usually modeled as a classification problem and can be estimated by multi-class learning models. The training data is collected from historical impressions and the corresponding click-through rates. Because of the high quality and reliable features, logistic regression models are widely used as the basis of the ad click prediction system.

Practical Lessons from Predicting Clicks on Ads at Facebook

Xinan He, Junfang Pan, Gu Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, Joaquim Quiñonero Candela

Facebook
1601 Willow Road, Menlo Park, CA, United States
{panjunteng, oujin, joaquin, sbowers}@fb.com

ABSTRACT

Online advertising systems at Facebook are responsible for over 700 million daily active users and over 1 million active advertisers, predicting clicks on Facebook ads is a challenging machine learning problem. In this paper, we introduce a model that can combine decision trees with logistic regression, outperforming either of these methods on its own, with significant improvements on a variety of test sets. We explore how to control the number of parameters that impact click prediction performance.

Keywords: click prediction, DNN, GBDT; model ensemble.

1. INTRODUCTION

Digital advertising is a multi-billion dollar industry and is growing dramatically each year. In most online advertising platforms the allocation of ads is dynamic, tailored to user interests based on their observed behavior. Machine learning plays a central role in computing the expected utility of a candidate ad to a user, and this in turn affects the efficiency of the marketplace.

Online advertising systems allow advertisers to bid and pay for measurable user interactions, such as clicks on ads. As a consequence, click prediction systems are central to online advertising systems, and over 700 million daily active users and over 1 million active advertisers, predicting clicks on Facebook ads is a challenging machine learning problem. In this paper, we introduce a model that can combine decision trees with logistic regression, outperforming either of these methods on its own, with significant improvements on a variety of test sets. We explore how to control the number of parameters that impact click prediction performance.

Keywords: click prediction, DNN, GBDT; model ensemble.

1. INTRODUCTION

Digital advertising is a multi-billion dollar industry and is growing dramatically each year. In most online advertising platforms the allocation of ads is dynamic, tailored to user interests based on their observed behavior. Machine learning plays a central role in computing the expected utility of a candidate ad to a user, and this in turn affects the efficiency of the marketplace.

Online advertising systems allow advertisers to bid and pay for measurable user interactions, such as clicks on ads. As a consequence, click prediction systems are central to online advertising systems, and over 700 million daily active users and over 1 million active advertisers, predicting clicks on Facebook ads is a challenging machine learning problem. In this paper, we introduce a model that can combine decision trees with logistic regression, outperforming either of these methods on its own, with significant improvements on a variety of test sets. We explore how to control the number of parameters that impact click prediction performance.

Keywords: click prediction, DNN, GBDT; model ensemble.
Published Research from the Trenches

• Initial efforts focused on **logistic regression**
• Most big production systems reportedly kept it simple in the final stage of prediction:
  – **Google (2013)**: prob. feature inclusion + Bloom filters $\rightarrow$ **logistic regression**
  – **Facebook (2014)**: boosted decision trees $\rightarrow$ **logistic regression**
  – **Yahoo (2014)**: hashing trick $\rightarrow$ **logistic regression**
• However...
Towards Neural Network Prediction

• More recently, Microsoft (2017) research
  – reports “factorization machines (FMs), gradient boosting decision trees (GBDTs) and deep neural networks (DNNs) have also been evaluated and gradually adopted in industry.”
  – recommends boosting neural networks with gradient boosting decision trees
Perspective

• The last sigmoid layer of a neural network (deep or otherwise) for binary classification is logistic regression.
• Previous layers of a deep neural network learn an internal representation of inputs, i.e. perform automatic feature engineering.
• Thus, most efforts to engineer successful, modern CTR prediction systems focus on layered feature engineering using:
  – Hashing tricks
  – Features engineered with GBDTs, FFMs, and deep neural networks (DNNs), or a layered/ensembled combination thereof.
• Future: Additional automated feature representation learning with deep neural networks
CTRP Conclusions

• To get prediction performance quickly and easily, hash data to binary features and apply logistic regression.
• For + few % of accuracy, dig into Kaggle forums and the latest industry papers for a variety of means to engineer features most helpful to CTR prediction. We’ve surveyed a number here.
• Knowledge is power. (↑ data → ↑ predictions)
• Priority of effort: ↑ data > ↑ feature engineering > ↑ learning/regression algorithms.
Next Steps

• Interested in learning more about Data Science and Machine Learning?
  – Create a Kaggle Account
  – Enter a learning competition, e.g. “Titanic: Machine Learning from Disaster”
  – Take related tutorials, learn from kernels and discussions, steadily work to improve your skills, and share it forward