AI Education Matters: Lessons from a Kaggle Click-Through Rate Prediction Competition

Todd W. Neller (Gettysburg College; tneller@gettysburg.edu)

DOI: 10.1145/3236644.3236646

Introduction

In this column, we will look at a particular Kaggle.com click-through rate (CTR) prediction competition, observe what the winning entries teach about this part of the machine learning landscape, and then discuss the valuable opportunities and resources this commends to AI educators and their students.

Kaggle's Criteo Display Advertising Challenge

Kaggle[1] is a data science/statistics/machine learning website that offers an excellent platform for modeling and prediction competitions. Data for training and analysis is often provided by companies, and top performers in competitions are encouraged and often required to supply and document their winning entries, offering valuable snapshots to current best practices in varied machine learning and data mining tasks.

Four years ago, Criteo Labs ran a Kaggle competition concerning CTR prediction called the "Criteo Display Advertising Challenge"[2]. The February 10, 2014 Criteo dataset was no longer available via the Kaggle competition site, but is still currently available from Criteo Labs[3]. The dataset is described on the Kaggle competition site as follows:

File descriptions

`train.csv` The training set consists of a portion of Criteo's traffic over a period of 7 days. Each row corresponds to a display ad served by Criteo. Positive (clicked) and negatives (non-clicked) examples have both been subsampled at different rates in order to reduce the dataset size. The examples are chronologically ordered.

`test.csv` The test set is computed in the same way as the training set but for events on the day following the training period.

Data fields

Label Target variable that indicates if an ad was clicked (1) or not (0).

I1-I13 A total of 13 columns of integer features (mostly count features).

C1-C26 A total of 26 columns of categorical features. The values of these features have been hashed onto 32 bits for anonymization purposes. The semantic of the features is undisclosed.

The training set consists of 45,840,617 examples, so competitors had to consider the size of the data when approaching the problem. The number of unique categorical feature values, for example, meant that a normal one-hot encoding of categorical features was computationally infeasible. Many numeric feature distributions were significantly skewed, so discretization via equal-width binning was inadvisable.

Also significant was the number of missing values in the dataset. Many machine learning (ML) and statistical learning texts have little or no coverage of the handling of missing values, and my own ML game research applications often involve complete information, so this wrinkle in both numeric and categorical data provides opportunities for learning beyond familiar, clean datasets.

Lessons from the Winners

Winners of this and 3 other recent CTR prediction competitions most often used two types of algorithms: gradient-boosted trees (GBTs, e.g. XGBoost Chen & Guestrin (2016)[4]), and field-aware factorization machines (FFMs, e.g.

[1] https://www.kaggle.com

[2] https://www.kaggle.com/c/criteo-display-ad-challenge

[3] https://s3-eu-west-1.amazonaws.com/criteo-labs/dac.tar.gz

[4] https://xgboost.readthedocs.io

libffm[5] Juan et al. (2016)). Even the winning team of Criteo's challenge made use of gradient-boosted decision trees to generate features for their FFMs[6].

Decision trees, a.k.a. classification and regression trees (CARTs), can handle missing values with ease, so the shortest path for a practitioner to see success in CTR prediction or related problems would be to learn the use of XGBoost. As an AI educator, I would want my students to *understand* GBTs, so I would want to guide them through the concept dependencies leading up to the understanding of GBTs.

In a previous column (Neller (2017)), I recommended general machine learning teaching resources for introducing the general problem of supervised learning. In that context, provide a basic introduction to decision trees using one of many good references (e.g. Quinlan (1986), James et al. (2014), §8.1, Russell & Norvig (2009), §18.1-18.3, Murphy (2012), §16.1-16.2, Mitchell (1997), Ch. 3). Next, introduce the concept of boosting (e.g. James et al. (2014), §8.1, Hastie et al. (2009), Ch. 10) and then gradient boosting (e.g. Hastie et al. (2009), §10.10, Chen & Guestrin (2016)).

Given the dominance of Python in the Kaggle community[7], I would recommend pairing these readings with practical Python exercises through Kaggle machine learning tutorials[8], the well-crafted, ongoing introductory competition on survivor prediction given Titanic passenger data[9], and even working with a subset of the Criteo dataset. I would further note that Kaggle now offers Kaggle InClass[10], a free, self-service platform that allows instructors to create classroom competitions.

As I explored Kaggle's Criteo CTR prediction competition and considered how I would guide a student to an appreciation of that work, I gained a great appreciation for the many authors that provide a foundational understanding for boosting trees, the excellent Kaggle data science community and their amazing platform, and the companies that partner with Kaggle to bring interesting challenges for the great educational benefit of all. I hope this column sparks your curiosity to explore the exciting educational opportunities these abundant resources offer.

[5] https://github.com/guestwalk/libffm

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

[7] https://www.kaggle.com/surveys/2017

[8] https://www.kaggle.com/learn/machine-learning, XGBoost-specific tutorial: https://www.kaggle.com/dansbecker/learning-to-use-xgboost

[9] https://www.kaggle.com/c/titanic

[10] https://www.kaggle.com/about/inclass/overview

References

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2939672.2939785 doi: 10.1145/2939672.2939785

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction* (2nd ed.). Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning: with applications in R*. Springer. (http://www-bcf.usc.edu/~gareth/ISL/)

Juan, Y., Zhuang, Y., Chin, W.-S., & Lin, C.-J. (2016). Field-aware factorization machines for CTR prediction. In *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 43–50). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2959100.2959134 doi: 10.1145/2959100.2959134

Mitchell, T. M. (1997). *Machine learning* (1st ed.). New York, NY, USA: McGraw-Hill, Inc.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. The MIT Press.

Neller, T. W. (2017, July). AI education: Machine learning resources. *AI Matters*, *3*(2), 14–15. Retrieved from http://doi.acm.org/10.1145/3098888.3098893 doi: 10.1145/3098888.3098893

Quinlan, J. R. (1986, March). Induction of decision trees. *Mach. Learn.*, *1*(1), 81–106.

Retrieved from http://dx.doi.org/10.1023/A:1022643204877 doi: 10.1023/A:1022643204877

Russell, S., & Norvig, P. (2009). *Artificial intelligence: A modern approach* (3rd ed.). Upper Saddle River, NJ, USA: Prentice Hall.

Todd W. Neller is a Professor of Computer Science at Gettysburg College. A game enthusiast, Neller researches game AI techniques and their uses in undergraduate education.