When Life Gives You Lemons. . .
A Survey of Gradient Boosting
ENGW3302: Project 2 Draft 2
Citation: IEEE

Paul Langton
langton.p@husky.neu.edu
Northeastern University

November 5, 2019

Abstract

Automated decisions have infiltrated almost every part of modern life due in part to modern advancements in machine learning. Gradient Boosting (GB) has played a key role in many of these advancements and has seen significant and interesting scientific development as a statistical method over the past 20 years. This literature review seeks to summarize the progress made in GB methods and demonstrate its impact in solving previously intractable real world problems. Writing style, publication metadata, methods including the origin of GB and its improvements, and some applications where GB is the state-of-the-art will also be covered.

1 Introduction

Automated decisions have infiltrated almost every part of modern life due in part to modern advancements in machine learning. Society has seen huge improvements in the baselines of previously intractable problems in operations research, search ranking, human speech, and many others. Gradient boosting (GB) is one advancement which has contributed significantly to each of these fields.

A mechanic can build a well-functioning car from the pieces of lemons, which are non functional but often new cars. Analogously, GB can build an excellent model from a set of "weak learners". More formally, GB is a gradient descent algorithm which produces a better classifier from an ensemble of small decision trees. The models it produces are *interpretable* because they are learned from simple decision trees. It is highly robust to outlier data, and generalizes well because it uses an approximate gradient most correlated with the true gradient. This unusual and highly sought-after set of traits has earned it an esteemed position among scientists and hobbyists alike.

This survey will enumerate and describe the significant theoretical improvements to the GB method, and connect these theoretical developments to their impact on solving real world problems. As of its publication, this paper will serve to outline the state-of-the-art of gradient boosting in machine learning.

1.1 Related Work

Recent survey publications on GB are plentiful, especially with the popularity enjoyed by machine learning in the current research climate. Surveys over advancements in learning-to-rank [1], ECG analysis [2], and object detection in images [3] all cite gradient boosting as a major step forward in tackling their specific problem. These surveys confirm the importance of GB in the current machine learning research climate, but the lack of a published survey of theoretical advances in gradient boosting which motivates this paper.

2 Writing in the Field

2.1 Structure

The structure of GB papers is roughly consistent. They follow the scientific research standard consisting of title, abstract, introduction, methods, results, conclusion. There are some notable differences in theoretical vs. applied

GB papers.

On the theory side, all emphasis is placed on the methods, which are split into sections determined by the different techniques used to improve a bound or baseline. These papers [4] [5] [6] may or may not include an experimental section, and if they do it is ancillary to theorems, proofs, and algorithms. Others [7] and [8] read more like a long-form lectures than research papers. They include include essentially no introduction and no related work, preferring instead to dive straight into methods and experiments.

Applied papers in GB address a specific problem which was solved with GB. They emphasize the structure of their data, the accuracy of the GB-produced classifier, and any computational tricks which made the algorithm perform better. All included a detailed experimental section with at least one plot of accuracy and a table of performance characteristics on various data sets. [9], [10], and [11] all exhibit these characteristics.

2.2 Metadata

The papers surveyed span the range 1999-2018, with 5 published in the last 3 years of that range. Three papers [7] [8] [10] are published in journals. The other 4 [5] [11] [6] [9] are published as part of conference proceedings. By subject area, two are published in mathematics, one in computational biology, and 4 in computer science. With regards to length, no paper lies outside the range of 8-16 pages besides the first paper on gradient boosting [7], which spans almost 32 pages without references. By location, authors come mainly from the US and China, with one from Europe [11]. Three articles contained industry collaboration from Microsoft Research [9] [6] and Yahoo Labs [4], while the rest had authors solely from academic institutions.

3 Methods

Gradient boosting is a method by which classifier is produced from an ensemble of "weak learners", generally decision trees, by iteratively minimizing a differentiable loss function. GB is therefore a type of a gradient descent algorithm.

This section will analyze the major theoretical developments in gradient boosting. Papers included must present a significant improvement in the gradient boosting method with theoretical justification. Due to the relative recency of its invention, the section will start with the seminal gradient boosting algorithm.

3.1 Greedy Function Approximation

Gradient boosting was proposed as "Greedy Function Approximation" in [7]. Consider a learning problem with N input variables $\boldsymbol{x}$ and corresponding desired outputs y. Generate a function $F(\boldsymbol{x})$ which minimizes the expected value of some loss function $L(y, F(\boldsymbol{x}))$ over the joint distribution of all $(y, \boldsymbol{x})$ values. In particular, approximate F from a set of "weak learners" $h(\boldsymbol{x}, \boldsymbol{a})$, where each h is taken to be a small regression tree parameterized by $\boldsymbol{a}$.

This problem gives rise to Alg. 1. At the mth iteration, the gradient of the loss function is calculated with respect to F_{m-1} of each input variable. To avoid overfitting on these input variables, smoothness is imposed on the function by calculating a set of parameters $\boldsymbol{a_m}$ which minimizes the squared distance between our previous loss gradient $\tilde{y_i}$ and our weak learners h, which creates a "constrained gradient" that generalizes better. Next the "step size" ρ_m is calculated from the minimization of the sum of losses a step magnitude ρ was taken in the direction of our constrained gradient. Finally the function F_m is additively adjusted by the loss-minimizing quantity $\rho_m h(\boldsymbol{x}; \boldsymbol{a}_m))$.

Algorithm 1 Gradient_Boost [7]

$F_0(\boldsymbol{x}) = \arg\min_\rho \sum_{i=1}^{N} L(y_i, \rho)$
for $m = 1$ to M **do**
  $\tilde{y_i} = -\left[\frac{\partial L(y_i, F(\boldsymbol{x_i}))}{\partial F(\boldsymbol{x_i})}\right]_{F(\boldsymbol{x})=F_m-1(\boldsymbol{x})}$, $i = 1 \dots N$
  $\boldsymbol{a}_m = \arg\min_{\boldsymbol{a},\beta} \sum_{i=1}^{N} [\tilde{y_i} - \beta h(\boldsymbol{x}_i; \boldsymbol{a})]^2$
  $\rho_m = \arg\min_\rho \sum_{i=1}^{N} L(y_i, F_{m-1}(\boldsymbol{x}_i) + \rho h(\boldsymbol{x}_i; \boldsymbol{a}_m))$
  $F_m(\boldsymbol{x}) = F_{m-1}(\boldsymbol{x}) + \rho_m h(\boldsymbol{x}; \boldsymbol{a}_m))$
end for

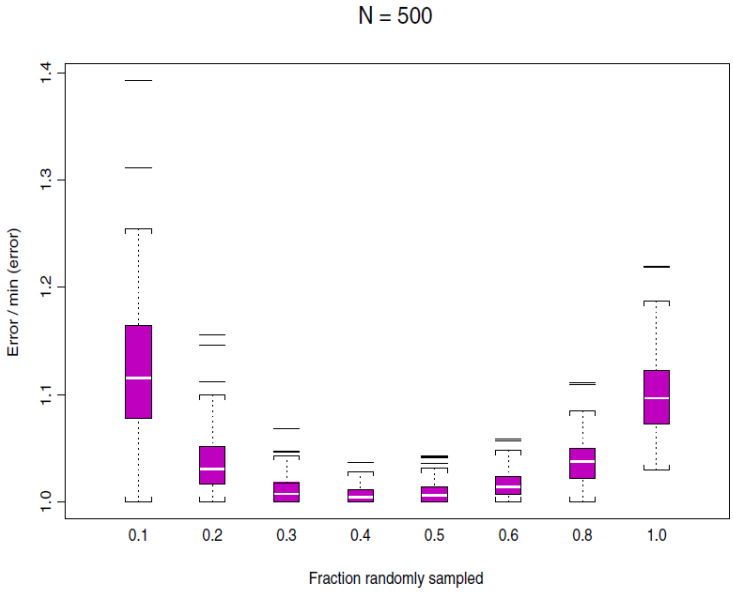


Figure 1: Error reduction from stochastic GB versus deterministic GB (Fraction randomly sampled= 1.0) [8]

3.2 Stochastic GB

The next step forward for the gradient boosting method came from the author of [7]. Friedman proposes the following in [8]. At each iteration m, randomly permute the input data and replace Line 3 of Alg. 1 with

$$\tilde{y_i} = -\big[\frac{\partial L(y_i, F(\boldsymbol{x_i}))}{\partial F(\boldsymbol{x_i})}\big]_{F(\boldsymbol{x})=F_{m-1}(\boldsymbol{x})}, i = 1 \dots \tilde{N}$$

where $\tilde{N} \leq N$. This reduces computation by a constant factor $\frac{\tilde{N}}{N}$, but also increases the variance of the individual weak learner estimates because they have less data to train on at each iteration. Perhaps surprisingly, this leads to a drastic reduction in overall error (22% improvement on the squared error scale). Friedman's (uncertain) explanation for this is that each training iteration is less correlated, which reduces the variance of the final model. Empirically, stochastic gradient boosting performs best for $\tilde{N} \approx \frac{1}{2}N$ (seen in Fig. 1). [8]

3.3 Online GB

The rise of high-volume and -throughput data makes online versions of many popular algorithms a necessity, and gradient boosting is no different. The first step towards true online gradient boosting comes from [4], which gives algorithms for learning limited classes of functions in the online setting from a weak learner A over the base function class F. They show that both span(F) and the convex hull (CH) of F can be learned in an online setting by providing concrete regret bounds on both.

Generally, the algorithm works as follows. Maintain N copies of the weak online learner A. Each of these copies will correspond to one stage in boosting. When the data stream returns a point $\boldsymbol{x}_t$, pass the point to each copy of A and observe the predictions. It then combines the predictions via linear combination to make its own prediction $\boldsymbol{y}_t$ and obtains corresponding losses l_t. Each A^i is then updated with a scaled linear approximation to the loss function $\frac{1}{L_D}\nabla l_t(\boldsymbol{y}_t^{i-1})$, which also coincides with a descent direction.

Alg. 2 (the simpler of the two) shows the learning process for the convex hull of F. It has a regret bound (optimal to a constant factor) of

$$R'(T) \leq \frac{8\beta_D D^2}{N}T + L_D R(T)$$

where D is a bound on the norm of the loss function output, β_D is a smoothness parameter on the loss function, L_D is a Lipschitz constant, and $R(T)$ is the regret the weak online learner incurs on the base function class F.

This result is not fully general as it gives bounds for only a small subset of desirable learnable functions (span and CH). There remains significant work to be done in the field of online gradient boosting.

Algorithm 2 Online Gradient Boosting for CH(F) [4]

Maintain N copies of the algorithm A, denoted $A^1, A^2, \ldots, A^N$, and let $\eta_i = \frac{2}{i+1}$ for $i = 1, 2, \ldots, N$.
for $t = 1$ to T **do**
  Receive example $\boldsymbol{x}_t$
  Define $\boldsymbol{y}_t^0 = \mathbf{0}$
  for $i = 1$ to N **do**
    Define $\boldsymbol{y}_t^i = (1 - \eta_i)\boldsymbol{y}_t^{i-1} + \eta_i A^i(\boldsymbol{x}_t)$
  end for
  Predict $\boldsymbol{y}_t = \boldsymbol{y}_t^N$
  Obtain loss function l_t and suffer loss $l_t(\boldsymbol{y}_t)$
  for $i = 1$ to N **do**
    Pass loss function $l_t^i(\boldsymbol{y}) = \frac{1}{L_D}\nabla l_t(\boldsymbol{y}_t^{i-1}) \cdot \boldsymbol{y}$ to A^i
  end for
end for

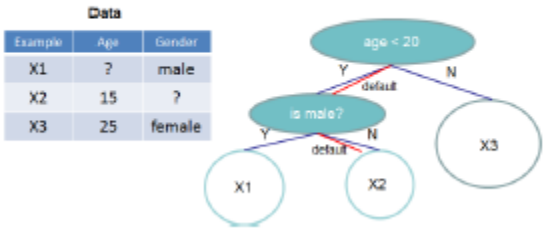


Figure 2: Example tree with default split directions. Default will be taken when feature is missing from the data [5].

3.4 Weighted Quantile Sketching

In theory, gradient boosting assumes access to some pre-constructed weak learners. In practice, it is the responsibility of the software to build these weak learners from data. The combination of building the decision trees in addition to boosting is known as Gradient Boosting Decision Trees (GBDT). Since these weak learners are almost always decision trees built sequentially, a common problem is finding the best "split", or the best leaf on the current decision tree to split to account for another feature. For data that fits into memory, this can be solved by enumerating all possible splits and choosing the best one according to the current loss function. For data that must be processed in a distributed setting, this is not possible. XGBoost contains the first theoretically justified implementation of an efficient, distributed algorithm which proposes splits via a weighted quantile sketch, which extends previous results on unweighted quantile sketching [5]. The algorithm is not brief, and interested readers are encouraged to consult the Appendix of [5] for further details.

3.5 Sparsity-Aware Split Finding

GB performs especially well on problems with categorical features. In real world data it is common for inputs representing categorical features to be sparse. Taking advantage of this trait is essential in optimizing GB's algorithmic performance. As in the "best split" problem described above, the goal is to find the best leaf on a decision tree to split to account for a certain feature. XGBoost implements a process called Sparsity-Aware Split Finding, which handles all sparsity patterns in the input data to find splits in a unified way [5]. Upon seeing the input data, XGBoost learns an optimal default pattern for splitting trees on sparse data entries. When a sparse (zero) feature entry is encountered, XGBoost takes its learned default split instead of taking the time to run weighted quantile sketching. Fig. 2 gives an intuitive visualization of the algorithm. In practice this is shown to be around 50x faster on certain data sets, seen in Fig. 3.

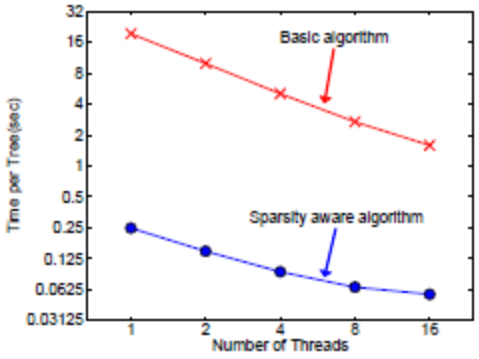


Figure 3: Performance of GBDT with Sparsity-Aware Split Finding vs. Base Algorithm [5]

3.6 Gradient-based One-Side Sampling

Gradient-based One-Side Sampling (GOSS) arises from observations gained from working with real-world data where, even with optimizations from XGBoost, working with a high number of feature dimensions and decision trees causes GBDT to slow to the point of unusability. One approach comes from the observation that, when constructing decision trees, those with gradients of greater magnitude (the "under-trained" trees) contribute more to the information gain of the final classifier. In standard GBDT, all decision trees (or a uniform random sample) are passed to the split-finding function discussed above. GOSS, proposed in [6], keeps the top percentile of trees by gradient in addition to a random sample of the trees with smaller gradients. This allows GOSS to maintain the faster runtime of downsampling the training trees, but also outperform random sampling in terms of training accuracy. Its asymptotic ratio of approximation error is shown to be

$$O\left(\frac{1}{n_l^j(d)} + \frac{1}{n_r^j(d)} + \frac{1}{\sqrt{n}}\right)$$

with n as the size of the input data, and $n_r^j(d)$ and $n_l^j(d)$ are the sizes of the right and left splits after splitting feature j at point d. If the split is not too unbalanced, as $n \to \infty$ this term approaches 0, which means the approximation is very accurate for big data. [6]

3.7 Exclusive Feature Bundling

Exclusive Feature Bundling (EFB) is another GBDT optimization which focuses on sparse input data. LightGBM [6] implements a method of feature compressing by bundling features which are mutually exclusive into a single feature. This significantly decreases training time with a very small decrease in accuracy. The most general definition of this problem, partitioning features into the smallest number of exclusive bundles, is NP-Hard. However, there exists a simple approximation algorithm (Fig. 4) which achieves good results with constant approximation ratio. Further, if the algorithm is relaxed to bundle a few *mostly* mutually exclusive (but still conflicting) features, training accuracy will be affected by at most $O([(1-\gamma)n]^{-2/3})$ where γ is the max feature conflict rate in a bundle. [6]

Exclusive Feature Bundling in combination with GOSS represents the current state of the art in Gradient Boosting Decision Trees. Comparative performance analyses over binary and multiple-classification problems can be found in Fig. 5.

4 Applications

Gradient boosting is sufficiently general to the point that surveying all instances where GB was applied to achieve high or even state-of-the-art accuracy on a problem is impractical. This section outlines a few key GB applications to convince the reader of its significance.

Papers were selected to be part of this section based on two criteria. First, they must address an important problem or class of problems. Second, they must achieve the state of the art in their given problem, or solve an entirely new problem. Finally, they must use some variant of gradient boosting to achieve their results.

Algorithm 3: Greedy Bundling

Input: F: features, K: max conflict count
Construct graph G
searchOrder $\leftarrow$ G.sortByDegree()
bundles $\leftarrow$ {}, bundlesConflict $\leftarrow$ {}
for i ***in*** *searchOrder* **do**
  needNew $\leftarrow$ True
  for $j = 1$ ***to*** *len(bundles)* **do**
    cnt $\leftarrow$ ConflictCnt(bundles[j],F[i])
    if *cnt* + *bundlesConflict[i]* $\leq K$ **then**
      bundles[j].add(F[i]), needNew $\leftarrow$ False
      break
  if *needNew* **then**
    Add $F[i]$ as a new bundle to *bundles*
Output: *bundles*

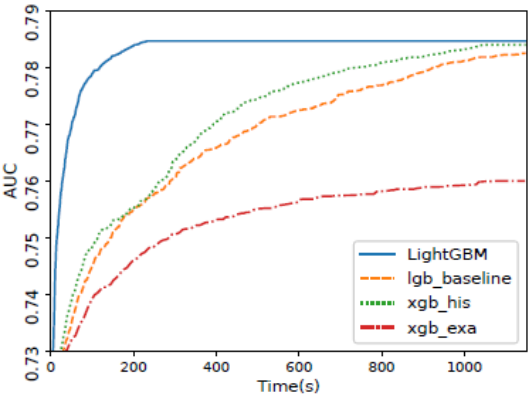
(a) Creating bundles of features with no more than K conflicts

Algorithm 4: Merge Exclusive Features

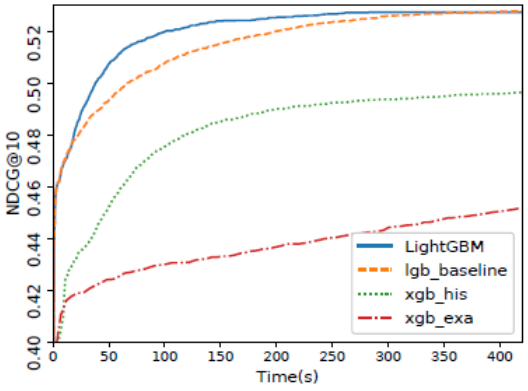
Input: *numData*: number of data
Input: F: One bundle of exclusive features
binRanges $\leftarrow$ {0}, totalBin $\leftarrow$ 0
for f ***in*** F **do**
  totalBin += f.numBin
  binRanges.append(totalBin)
newBin $\leftarrow$ new Bin(numData)
for $i = 1$ ***to*** *numData* **do**
  newBin[i] $\leftarrow$ 0
  for $j = 1$ ***to*** *len(F)* **do**
    if *F[j].bin[i]* $\neq 0$ **then**
      newBin[i] $\leftarrow$ F[j].bin[i] + binRanges[j]
Output: *newBin*, *binRanges*

(b) Process for merging a bundle of features into a single feature

Figure 4: The two subprocesses which compose the EFB algorithm [6]



(a) Time-AUC curve on Flight Delay



(b) Time-NCDG curve on LETOR

Figure 5: Diagrams from [6] detailing the performance of LightGBM on binary and multiple classification tasks when compared to XGBoost, the previous GBDT state of the art

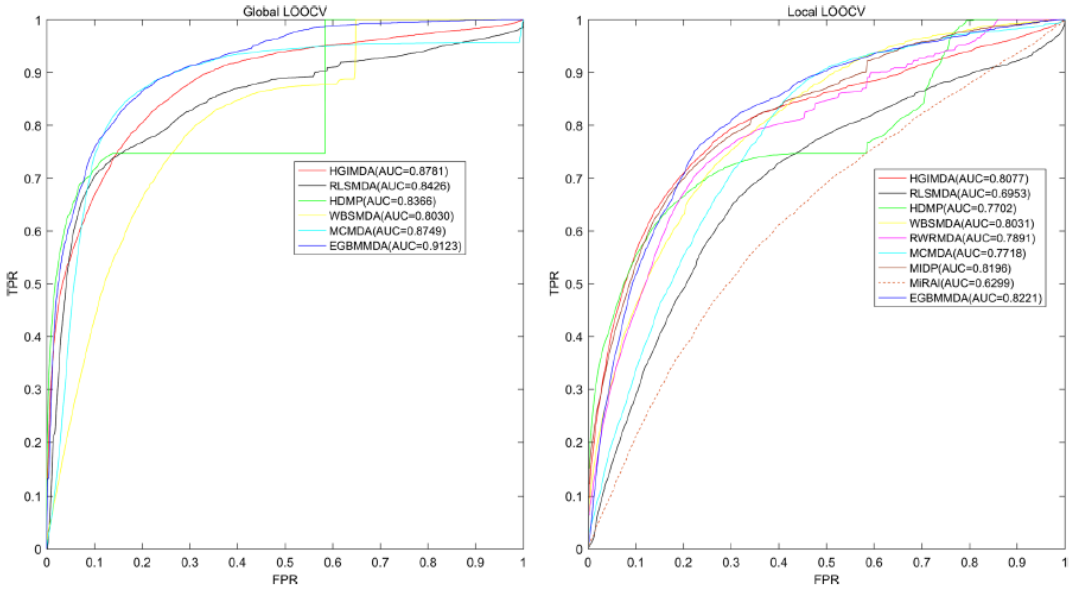


Figure 6: Accuracy of EGBMMDA compared with previous state of the art disease-miRNA predictors using ROC on the HMDD v2.0 dataset [10]

4.1 Bioinformatics

Gradient boosting is used in the field of bioinformatics to predict connections between human microRNAs (miRNAs) and diseases. In [10], the authors construct a novel state-of-the-art model called Extreme Gradient Boosting Machine for MiRNA-Disease Association (EGBMMDA) built from gradient boosted regression trees which predicts a number of potentially related miRNAs for a given disease. They build three main features:

- From a dataset (HMDD v2.0) consisting of 5430 experimentally confirmed associations between 495 miRNAs and 383 diseases, a matrix which has entry 1 whereever disease i is shown to be related to miRNA j
- A matrix whose entries represent the functional similarity of a miRNA pair
- From a directed graph of disease ancestry, a matrix whose entries are the semantic similarity between a pair of diseases

Their model is trained using XGBoost [5], and cross validated. The model performs well in the author's case studies. They perform case studies predicting potential miRNAs for Colon Neoplasms, Lymphoma, Prostate Neoplasms, Breast Neoplasms, and Esophageal Neoplasms and achieve impressive results: 98%, 90%, 98%, 100%, and 98% of the top 50 predictions for the five diseases were experimentally confirmed to influence the disease in question. Fig. 6 shows that EGBMMDA outperforms all previous models on the HMDD v2.0 dataset. [10]

4.2 Search Ranking

Machine-learned search ranking, also called the Learning to Rank (LETOR) problem, takes a set of documents returned by a search query and ranks them according to their importance with respect to the query. This is a hugely important problem for search engines and receives frequent state-of-the-art updates. This is not a binary classification problem, thus our key metric for comparison is different. The LETOR problem seeks to optimize the normalized discounted cumulative gain (NDCG), given in Eq. Equation 1, where the numerator is the discounted cumulative gain and the denominator is the ideal discounted cumulative gain. rel_i is the graded relevance value of document i, and REL_p is the list of relevant documents ordered by relevance up to rank p. Intuitively, at a particular rank p, a good metric will give us the quality of documents seen with respect to the query, and penalize for ranking less relevant results above more relevant ones. This is expressed via the logarithmic penalty with respect to position i in Eq. Equation 1.

$$NDCG_p = \frac{\sum_{i=1}^{p} \frac{2^{rel_i}-1}{\log_2(i+1)}}{\sum_{i=1}^{\|REL_p\|} \frac{2^{rel_i}-1}{\log_2(i+1)}} \qquad \text{(Equation 1)}$$

Method	Dataset	NDCG
LambdaRank [12]	Yahoo LETOR	0.757
McRank [9]	Web-2	0.758
LambdaMART [13]	Yahoo LETOR	0.778
Ensemble [13]	Yahoo LETOR	0.7879

Table 1: LETOR state-of-the-art baseline updates due to gradient boosting. LambdaRank is the previous baseline and does not use gradient boosting. McRank was not tested on the Yahoo LETOR dataset.

The use of gradient boosting in LETOR was first proposed in [9], where the authors beat (their own) recently published state-of-the-art LETOR algorithm LambdaRank [12]. They build a gradient boosted model which can be easily tuned for three different classification strategies. The same authors immediately went back to work and produced LambdaMART, a gradient boosted MART ranking algorithm which currently holds the state-of-the-art, winning Track 1 of Yahoo's Learning To Rank competition by being used in an ensemble with LambdaRank [13]. It is currently in use by Microsoft's Bing search engine. Each algorithm's performance can be found in Table 1.

5 Limitations

For all its demonstrated successes, GB is not without weaknesses. The limitations of GB include barriers encountered by all ML models and a few all its own. Optimization in continuous space and data quality are universal ML barriers, while incomplete online algorithms and reliance on decision trees are GB-specific problems.

GB optimizes an arbitrary continuous loss function [7] and thus shares the basic limitations of any optimizer in continuous non-convex space, especially getting stuck in local optima. This limitation will always exist since globally optimizing an arbitrary non-convex function can take infinite time.

GB is often less effective when feature engineering is difficult or non-obvious, common in the areas of image classification and speech modelling where it is empirically outperformed by traditional deep learning. This is unlikely to change due to GB's preference for categorical features encoded in small decision trees, ideal for regression problems.

Strides have been made in developing online GB algorithms, but realistically the research community is far from having an optimal-regret online GB algorithm. The only existing proof of an optimal regret bound for online GB applies only for learning the convex hull of basis functions. This is a limited scenario that almost never arises in practice, and thus GB cannot effectively be used in practice in an online setting. [4]

Like any machine learning model, GB is only as good as the data it consumes. If it learns decision trees from biased data, poorly cleaned data, or improperly collected data the final model will exhibit those flaws. These problems are not in scope for GB researchers, but advances in data cleaning or bias identification and removal would have enormous positive impact on the accuracy of GB models.

6 Conclusion

Gradient boosting has seen significant scientific development in recent years, addressing problems like online boosting, automatic handling of sparse data, and targeting of undertrained instances. All of these advances have allowed for state-of-the-art results in other fields, such as bioinformatics and search engine result ranking. Current research trends suggest that gradient boosting will continue to be the target of research, and that further study may reveal important concepts about ensemble learning and machine learning at large.

Based on the conclusions of the literature in this review, the way forward for GB research is in optimizing for GB's strong points and improving data cleaning techniques. First, finding better ways to handle categorical features could improve both training time and final model accuracy significantly. Second, improving training time over high-dimensional data has shown to be a consistent challenge not only for GB, but for all machine learning models. Classical algorithms consistently struggle with the "curse of dimensionality", where run times are often exponential with respect to the dimensionality of the data and GB is no different. Improving training time over ever-greater numbers of weak learners as data size grows will allow GB to remain relevant in the age of big data. Finally, since the quality of GB models is only as good as the data it is provided, a clear path for future research lies in data cleaning and feature extraction. Studies should be done on the best way to ensure datasets are complete, lack bias, and have high information to noise ratios.

7 Acknowledgements

Thanks to Ohad and Nick for bringing me food and reminding me to take breaks.

References

[1] A. Phophalia, "A survey on learning to rank (letor) approaches in information retrieval," in *2011 Nirma University International Conference on Engineering*, pp. 1–6, Dec 2011.

[2] C. Roopa and B. Harish, "A survey on various machine learning approaches for ecg analysis," *International Journal of Computer Applications*, vol. 163, no. 9, pp. 25–33, 2017.

[3] D. K. Prasad, "Survey of the problem of object detection in real images," *International Journal of Image Processing (IJIP)*, vol. 6, no. 6, p. 441, 2012.

[4] A. Beygelzimer, E. Hazan, S. Kale, and H. Luo, "Online gradient boosting," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2458–2466, Curran Associates, Inc., 2015.

[5] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[6] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," in *NIPS*, 2017.

[7] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[8] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics Data Analysis*, vol. 38, no. 4, pp. 367 – 378, 2002. Nonlinear Methods and Data Mining.

[9] P. Li, C. J. C. Burges, and Q. Wu, "Mcrank: Learning to rank using multiple classification and gradient boosting," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, (USA), pp. 897–904, Curran Associates Inc., 2007.

[10] X. Chen, L. Huang, D. Xie, and Q. Zhao, "Egbmmda: Extreme gradient boosting machine for mirna-disease association prediction," in *Cell Death Disease*, 2018.

[11] T. Hoch, "An ensemble learning approach for the kaggle taxi travel time prediction challenge," in *Proceedings of the 2015th International Conference on ECML PKDD Discovery Challenge - Volume 1526*, ECMLPKDDDC'15, (Aachen, Germany, Germany), pp. 52–62, CEUR-WS.org, 2015.

[12] C. J. C. Burges, R. Ragno, and Q. V. Le, "Learning to rank with nonsmooth cost functions," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, (Cambridge, MA, USA), pp. 193–200, MIT Press, 2006.

[13] C. Burges, K. Svore, P. Bennett, A. Pastusiak, and Q. Wu, "Learning to rank using an ensemble of lambda-gradient models.," *Journal of Machine Learning Research - Proceedings Track*, vol. 14, pp. 25–35, 01 2011.