Gradient Boosted Trees at 20:
A View from Industry

Jan Pedersen
_Distinguished Scientist_
_Uber Technologies_
Outline

- AI Revolution
- A Pioneer
- Quick View of Gradient Boosting
- Application to Ranking
- Twenty Years Later
AI Revolution
Why the Excitement?

- Breakthrough in Traditional AI Tasks
  - Data-intensive methods
  - Massive compute
  - Improved algorithms

- Phase Transition
  - From curiosity to useable technology
  - Application explosion

- Many Tasks Remain Inaccessible
  - Planning and control
  - Fully autonomous agents
Origins

- Earlier Attempts were Engineered
  - Hand-tuned
  - Not robust or scaleable
- Automation is Critical
  - Not just imitation
  - but replacement
- Example: Universal Function Approximation
  - Wide range of application
  - Simplified training
  - E.g. Gradient Boosted Trees
A Pioneer
● Greedy Function Approximation: A Gradient Boosting Machine; Feb 1999a
● Stochastic Gradient Boosting; March 1999b

Source: A Conversation with Jerry Friedman; N.I. Fisher, Statistical Science Vol. 30 No. 2; 2015
Couldn’t Resist

From the PRIM-9 Video
A Quick Review
Greedy Function Approximation

- Minimize Generalization Error
  - Arbitrary differentiable loss
  - Typically squared error

- For an Ensemble of Weak Learners
  - Typically regression trees

- Via Gradient Descent
  - In function space
  - Fit a weak learner to pseudo residuals

\[
\hat{F} = \arg \min_F \mathbb{E}_{x,y} [L(y, F(x))].
\]

\[
\hat{F}(x) = \sum_{i=1}^{M} \gamma_i h_i(x) + \text{const}.
\]

\[
F_m(x) = F_{m-1}(x) - \gamma_m \nabla_{F_{m-1}} L(y_i, F_{m-1}(x_i)),
\]

\[
\gamma_m = \arg \min_{\gamma} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i)) - \gamma \nabla_{F_{m-1}} L(y_i, F_{m-1}(x_i)).
\]
Properties

- **Universal Function Approximation**
  - Given enough trees
  - In practice, capacity is limited
- **Resistant to Overfitting**
  - Regularization via learning rate
  - Stochastic Gradient Descent
    - Out-of-Bag Error
- **Variable Importance**
  - Understandability
- **High Quality Open-Source Implementations**
- **Requires Feature Engineering and Tuning**
  - Interactions
  - Hyper parameters
Application to Ranking
Ad Hoc Retrieval

- Return Documents Matching a Query
  - Ranked in some order
  - Later in “relevance” order
- From Boolean to Keyword Search
  - Gerard Salton (Cornell)
  - Bag-of-Words model
    - Cosine scoring
- Judged Relevance
  - Expert editors
  - Graded relevance scores
  - nDCG

\[
DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i + 1)}
\]
Ranking as Regression Problem

- Trec Conference Drove Innovation
  - Standard task and training Set
- Hand-Crafted Ranking Functions
  - Inspired by analysis
  - BM25 (S. Robertson 1998) was best-of-breed
- Early work on Ranking as Regression
  - W. Cooper, A. Chen and F. Gey
    Experiments in the Probabilistic Retrieval of Full Text Documents; 1997
  - Query/Document features
Alta Vista: The First Machine Learned Ranking Function

- Extreme competition for search share
- Incumbent ranking technology failed to improve
  - Open-loop optimization
- Top-down decision to switch technologies
- Gradient Boosted Trees as learning method
- Launched in June 2003
Yahoo and Bing

- **ML as competitive advantage**
  - Factors feature development from optimization
  - Supports distributed innovation
  - Continuous improvement cycle

- **Online code is automatically generated**
  - Compiled from tree representation
  - Efficiency is a concern

- **Independently developed at MSR**
  - Ranknet and LambdaRank (Burges 2005)
  - Ultimately adopted Gradient Boosted Trees
Twenty Years Later
An Indispensable Technology

- Default Learning Method
  - Embedded in many industrial ML platforms
  - E.g. Bing Aether, Uber Michelangelo

- Ubiquitous Deployment
  - Bing: entire search stack
  - Twitter: ads, news feed
  - eBay: search ranking
  - Uber: eta prediction, risk, safety