Gradient Boosting, Continued

David Rosenberg

New York University

December 26, 2016
Review: Gradient Boosting
The Gradient Boosting Machine

1. Initialize $f_0(x) = 0$.

2. For $m = 1, 2, \ldots$ (until stopping condition met)
   
   1. Compute unconstrained gradient:
      
      $$g_m = \left( \frac{\partial}{\partial f(x_i)} \left( \sum_{i=1}^{n} \ell(y_i, f(x_i)) \right) \right|_{f(x_i) = f_{m-1}(x_i)} \right)^{n}$$

   2. Fit regression model to $-g_m$:
      
      $$h_m = \arg\min_{h \in \mathcal{F}} \sum_{i=1}^{n} ((-g_m)_i - h(x_i))^2.$$  

   3. Choose fixed step size $\nu_m = \nu \in (0, 1]$ [$\nu = 0.1$ is typical], or take
      
      $$\nu_m = \arg\min_{\nu > 0} \sum_{i=1}^{n} \ell(y_i, f_{m-1}(x_i) + \nu h_m(x_i)).$$

   4. Take the step:
      
      $$f_m(x) = f_{m-1}(x) + \nu_m h_m(x).$$
Unconstrained Functional Gradient Stepping

Where $R(f)$ is the empirical risk.
Issue: $\hat{f}_M$ only defined at training points.

From Seni and Elder's *Ensemble Methods in Data Mining*, Fig B.1.
Projected Functional Gradient Stepping

\[ T(x; p) \in \mathcal{F} \text{ is our actual step direction (projection of } -\mathbf{g} = -\nabla R \text{ onto } \mathcal{F}) \]

From Seni and Elder’s *Ensemble Methods in Data Mining*, Fig B.2.
The Gradient Boosting Machine: Recap

- Take any [sub]differentiable loss function.
- Choose a base hypothesis space for regression.
- Choose number of steps (or a stopping criterion).
- Choose step size methodology.
- Then you’re good to go!
Gradient Tree Boosting

- Common form of gradient boosting machine takes

  \[ \mathcal{F} = \{ \text{regression trees of size } J \}, \]

  where \( J \) is the number of terminal nodes.

- \( J = 2 \) gives decision stumps

- HTF recommends \( 4 \leq J \leq 8 \).

- Software packages:

  - Gradient tree boosting is implemented by the \texttt{gbm package} for R
  - as \texttt{GradientBoostingClassifier} and \texttt{GradientBoostingRegressor} in \texttt{sklearn}

- For trees, there are other tweaks on the algorithm one can do

  - See HTF 10.9-10.12
GBM Regression with Stumps
Sinc Function: Our Dataset

From Natekin and Knoll's "Gradient boosting machines, a tutorial"
Fitting with Ensemble of Decision Stumps

Decision stumps with 1, 10, 50, and 100 steps, step size $\lambda = 1$.

From Natekin and Knoll's "Gradient boosting machines, a tutorial"
GBM Regression with Stumps

Step Size as Regularization

A

B

Performance vs rounds of boosting and step size.

From Natekin and Knoll’s "Gradient boosting machines, a tutorial"
Variations on Gradient Boosting
Stochastic Gradient Boosting

- For each stage,
  - choose random subset of data for computing projected gradient step.
  - Typically, about 50% of the dataset size.
  - Fraction is often called the **bag fraction**.

- Why?
  - Faster.
  - Subsample percentage is additional regularization parameter.

- How small a fraction can we take?
Column / Feature Subsampling for Regularization

- Similar to random forest, randomly choose a subset of features for each round.

- XGBoost paper says: "According to user feedback, using column sub-sampling prevents overfitting even more so than the traditional row sub-sampling."
Newton Step Direction

- For GBM, we find the closest $h \in \mathcal{F}$ to the negative gradient
  \[ -g = -\nabla_f J(f). \]

- This is a “first order” method.

- Newton’s method is a “second order method”:
  - Find 2nd order (quadratic) approximation to $J$ at $f$.
    - Requires computing gradient and Hessian of $J$.
  - Newton step direction points towards minimizer of the quadratic.
  - Minimizer of quadratic is easy to find in closed form.

- Boosting methods with projected Newton step direction:
  - LogitBoost (logistic loss function)
  - XGBoost (any loss – uses regression trees for base classifier)