Portfolio Question 3.2.P - Behind Boosting Trees

Daniel Lawson University of Bristol

Portfolio 3.2.P (v4.0.0)

Portfolio Question 3.2.P - Behind Boosting Trees

- How are the state-of-the-art boosting algorithms implemented?
- What is the intuition behind why boosting works?

LightGBM and Histogram Boosting

- LightGBM[1] is a popular, efficient implementation of boosted decision trees
  - It uses a histogram approach to speed up learning
  - It is very fast, and can be faster than random forests
- Speed and efficiency comes from:
  - Binning continuous features into discrete bins
  - Binning categorical features into **binary** bins (no one-hot encoding)
  - Using a leaf-wise tree growth algorithm (over depth-wise)

[1] LightGBM: A Highly Efficient Gradient Boosting Decision Tree (2017) Microsoft Research, url.

Intuition

- From Medium: The Intuition Behind Gradient Boosting & XGBoost:
  - If we can account for our model's errors, we will be able to improve our model's performance.
  - We can train a new predictor to predict the errors made by the original model.
  - Repeat for each datapoint. Careful to regularise.

Boosting loss function

- Use the squared error $0.5(y - \hat{y})^2$ with gradient $\hat{y} - y$
- We're learning some parameter(s) θ
- When the loss is **min**$_\theta \sum_{i=1}^{N} 0.5(y_i - \hat{y}_i(\theta, \mathbf{x}_i))^2 + 0.5\lambda\theta^2$
- We can solve for θ by taking derivatives and setting equal to zero
- Note it simplifies if $\hat{y}(\theta)$ is simple
  - they use the mean in the article
  - also straightforward for the effect of one covariate on y

Discussion

- What is the role of λ?
  - When θ is small? large?
  - Does it relate to information criteria?
  - Are there other models that focus on residuals? Why?