LigthGBM

Timetable

Question List to Chapter 7

BREAK

Jupyter-Notebook

Presentation LightGBM

LightGBM

= **light** **g**radient-**b**oosting **m**achine

- Published 2016
- Free and open-source, https://github.com/Microsoft/LightGBM
- Originally developed by Microsoft
- Won many machine learning competitions
- Faster for data sets with more than 10,000 instances
- Can work in parallel

Background

- Gradient Boosting Decision Trees build a series of decision trees which correct the losses of the processor.
- For big data set there is a *tradeoff between accuracy and efficiency*: The longer the series of trees the better the accuracy but also the number of computations to fit the trees:
  Gradient Boosting Decision Trees computational complexities are proportional to the number of features and the number of instances.

Three special techniques to reduce both:

- Histogram-based building of the decision tree
- Gradient-Based One-Side Sampling (GOSS)
- Exclusive Feature Bundling (EFB)

Histogram-based decision tree

Ordinary Decision Tree:

- Finding the best split in a decision tree is the most time-consuming part
- The features values has to be pre-sorted for every split.

Histogram-based decision tree:

- Histogram-based algorithms buckets continuous feature values into discrete bins
- These bins are used to construct histograms during training.
- The histograms are used to find the best split.
- Since the number of bins is much smaller than the number of instances, it is much faster.

In Scikit-learn HistGradientBoostingClassifier and –Regressor use this technique as well.

Histogram-based decision tree

Example for building the histogram:

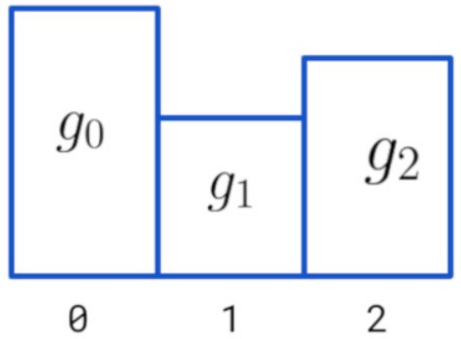
Original data set has 4 instances with continuous parameters:

$$\begin{bmatrix} 1.5 & 0.0 \\ 0.0 & 5.5 \\ 0.3 & 7.0 \\ 5.5 & 8.5 \end{bmatrix}$$

We bucket them into 3 bins. The we get:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 2 \\ 2 & 2 \end{bmatrix}$$

The residual error = the gradient is stored just for each bin and not for each instance:



Taken from: https://robotenique.github.io/posts/gbm-histogram/

Gradient-Based One-Side Sampling (GOSS)

Down sampling of the data instances

Idea:

- Keep those instances with large gradients (than a pre-defined threshold or among the top percentiles)
- Drop only those instances with small gradients randomly

Exclusive Feature Bundling (EFB)

In real applications usually the feature space is quite sparse.

Many features are almost exclusive, i.e. they rarely take nonzero values simultaneously, for example one-hot-encoded features.

Idea:

- Bundle such exclusive features

Exclusive Feature Bundling uses an efficient algorithm by transforming the bundling problem into a graph coloring problem and solving this with a greedy algorithm.