Ensemble Methods: Bagging and Boosting

Yay fun times!

Ensemble Methods - What are they?

Ensemble Methods: They're supervised machine learning methods that combine several base models to improve predictive accuracy.

Pro: They are often more accurate than the base models they are composed of
Con: Lose interpretability

Two Types:

Averaging Method (Bagging): Building several models independently, then averaging the predictions

- The combined estimator is generally better than the single model because variance is reduced

Boosting Method: Base estimators are built sequentially(each successive model depends on the one before it), each trying to reduce the bias of the combined estimator

- Basically combining several weak models to build a strong one

Bagging (Bootstrap Aggregation)

Bagging involves manipulating the training set by resampling:

- We have ‘n’ number of base estimators/models (eg. Decision Trees), and ‘n’ number of samples of training data.
    - The sample are made by resampling the training data with uniform weights
    - Each model in the ensemble carries equal weight
    - To promote variance, bagging trains each model in the ensemble with a randomly drawn subset of the data
    - Creates new training sets uniformly and with replacement

Models are then fitted using those samples and combined by averaging the output (for Regression) or by voting (for Classification)

Pros and Con of Bagging

Pro: Since each sample of the training data is equally likely to be used, bagging is not very susceptible to overfitting

- Because of this, bagging works best with strong, complex models (eg. a well-developed decision tree)

Con: You lose interpretability

Common Parameters to Tweak

Base_estimator: You’re base model. it’s usually a Decision Tree(that’s also the default setting).

N_estimators: the number of estimators you want in the ensemble (default is 10)

Boosting

- Takes a weak base model and tries to make it stronger by retraining it on the misclassified samples
  - The base model fitting is iterative/sequential
  - Weights assigned to observations indicate their importance - the higher the weight, the more influence it has on the total error of the next model
  - Weights change at each iteration with the goal of correcting the error/misclassification of the previous iteration.

- Final prediction is typically constructed by a weighted vote - each base model is weights depending on their training error/misclassification, so each model in the ensemble has a different level of influence on the overall output

Pros & Cons

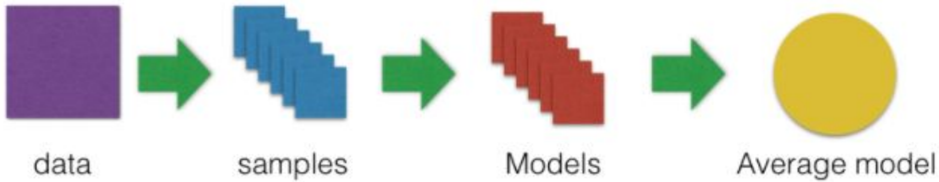
Pros:

- Achieves higher performance than bagging when hyper-parameters tuned properly.
- Can be used for classification and regression equally well.
- Easily handles mixed data types.
- Can use "robust" loss functions that make the model resistant to outliers.
- **Boosting aims to reduce bias!** (and can reduce variance a bit as well).

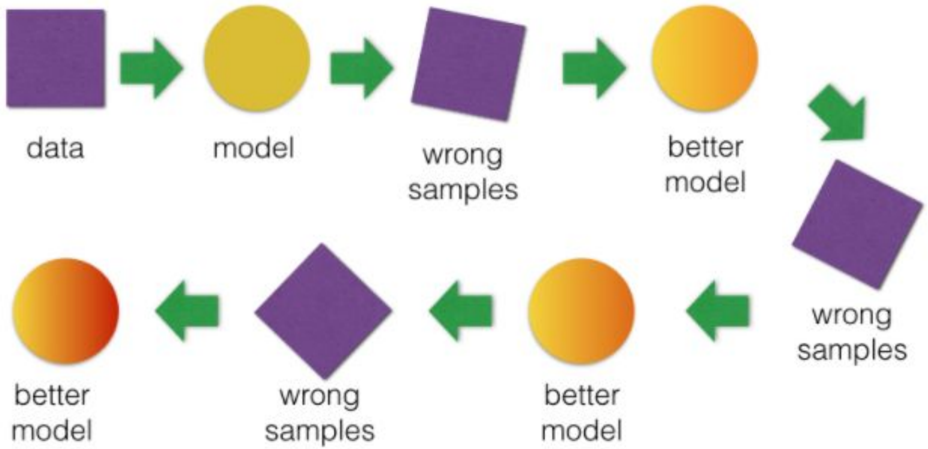
Cons:

- Difficult and time consuming to properly tune hyper-parameters.
- Cannot be parallelized like bagging (bad scalability when huge amounts of data).
- More risk of overfitting compared to bagging.

Bagging



Boosting



AdaBoost!

The core principle of AdaBoost is to **fit a sequence of weak models on repeatedly modified versions of the data**. After each fit, the importance weights on each observation need to be updated.

The predictions are then combined through a weighted majority vote (or sum) to produce a final prediction.

All training examples start with equal importance weighting.

As iterations proceed, observations that are difficult to predict receive increasing importance.

Gradient Boosting

- Gradient Boosting Classifier is a generalization of boosting to arbitrary differentiable loss functions.

- GBRT is an accurate and effective off-the-shelf procedure that can be used for both regression and classification problems. Gradient Tree Boosting models are used in a variety of areas including Web search ranking and ecology.

The advantages of GBRT are:

- Natural handling of data of mixed type (= heterogeneous features).
- Predictive power.
- Robustness to outliers in output space (via robust loss functions).

The disadvantages of GBRT are:

- Scalability, due to the sequential nature of boosting it can hardly be parallelized.
- Difficult hyperparameters to tune.


ANY QUESTIONS?
Shappie112