ECON7960 User Experience and A/B Test

Hong Kong Baptist University

Topic 8: GA Revisited, Boosting and Application

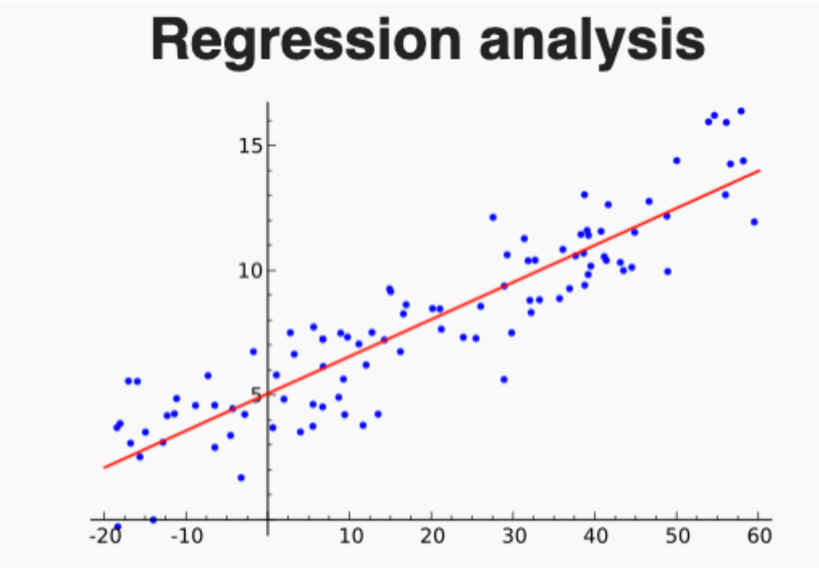
In previous 7 topics, we cover

- Tree Decision Concept & Clustering
- Machine Learning Algorithm: Random Forest
- Application of Random Forest in LAB1 to select Feature Importance

Multi-variate Testing:

Multi- Linear Regression and Logistic Regression

Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable and one or more independent variables.
The most common form of regression analysis is linear regression, in which a researcher finds the line that most closely fits the data according to a specific mathematical criterion.



Logistic regression model (or **logit model**) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. Each classification depends on variable determining the assignment of a probability between 0 and 1 and the sum adding to one.

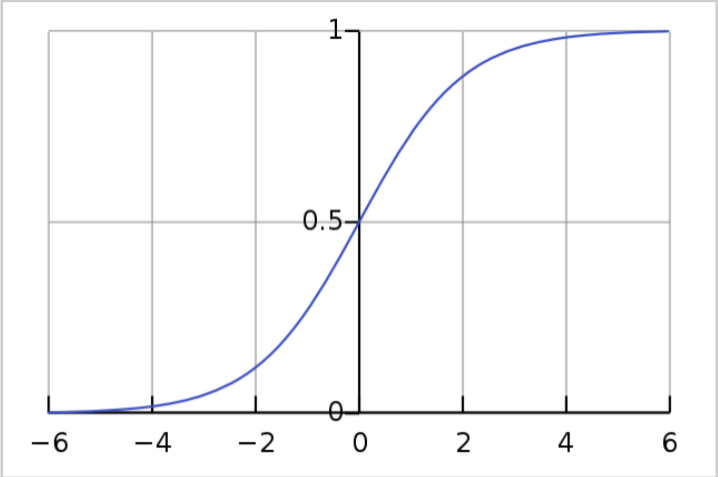


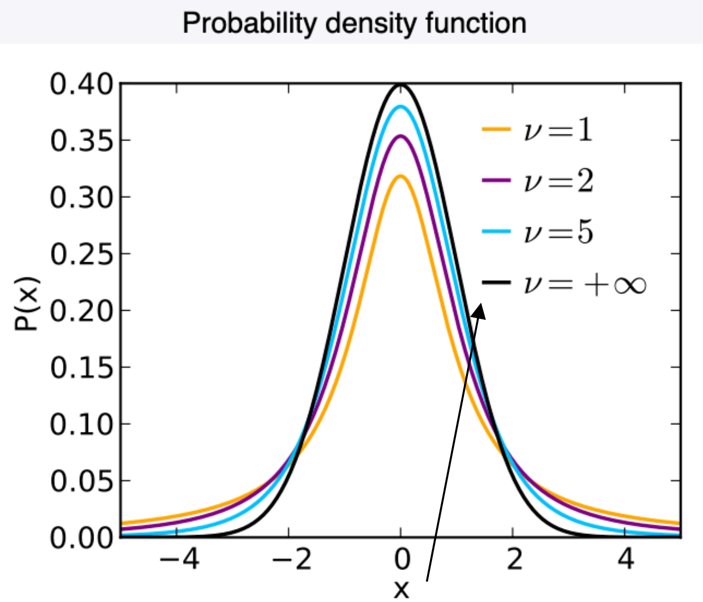
Figure 1. The standard logistic function $\sigma(t)$; note that $\sigma(t) \in (0,1)$ for all t.

$$\sigma(t) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

Workshop 5 Answer:

T-test versus ANOVA

Z-test, commonly applied test statistic would follow a normal distribution if the value of a scaling term (the variance) in the test statistic were known. When the scaling term is unknown and is replaced by an estimate based on the data (sample standard error), the test statistics (under certain conditions) follow a Student's *t* distribution (**t-test**).



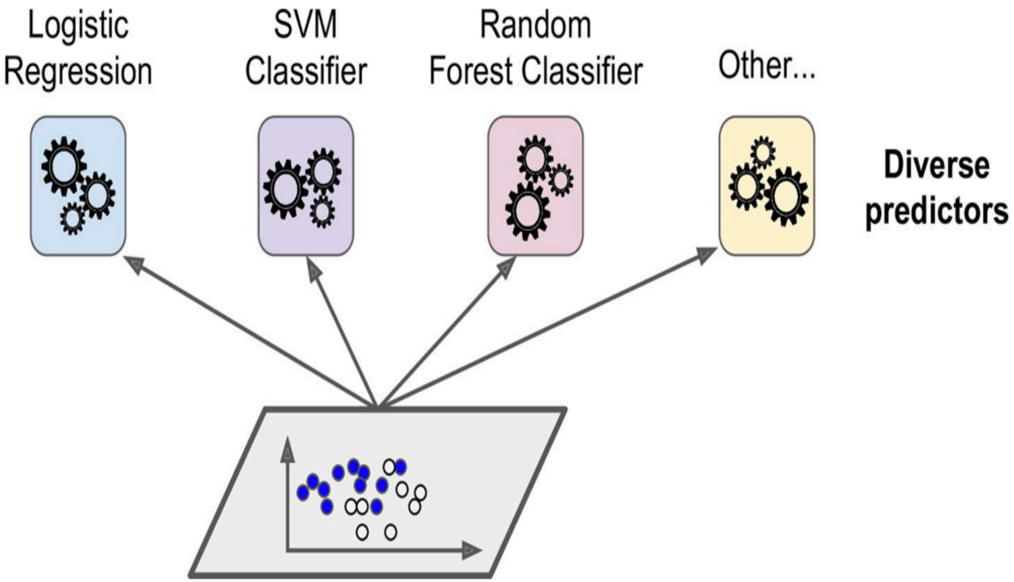
Normal distribution as degree of freedom goes to infinity

ANOVA provides a statistical test of whether two or more population means are equal, and therefore generalizes the *t*-test beyond two means if there is only one dependent variable.

BASIS FOR COMPARISON	T-TEST	ANOVA
Meaning	T-test is a hypothesis test that is used to compare the means of two populations.	ANOVA is a statistical technique that is used to compare the means of more than two populations.
Test statistic	$(\bar{x}-\mu)/(s/\sqrt{n})$	Between Sample Variance/Within Sample Variance

Training Diverse Classifiers:

- A group of predictors is called an ensemble ; thus, this technique is called Ensemble Learning , and an Ensemble Learning algorithm is called an Ensemble method.
- The wisdom of the crowd by aggregating the predictions of a group of predictors (such as classifiers or regressors), you will often get better predictions than with the best individual predictor.

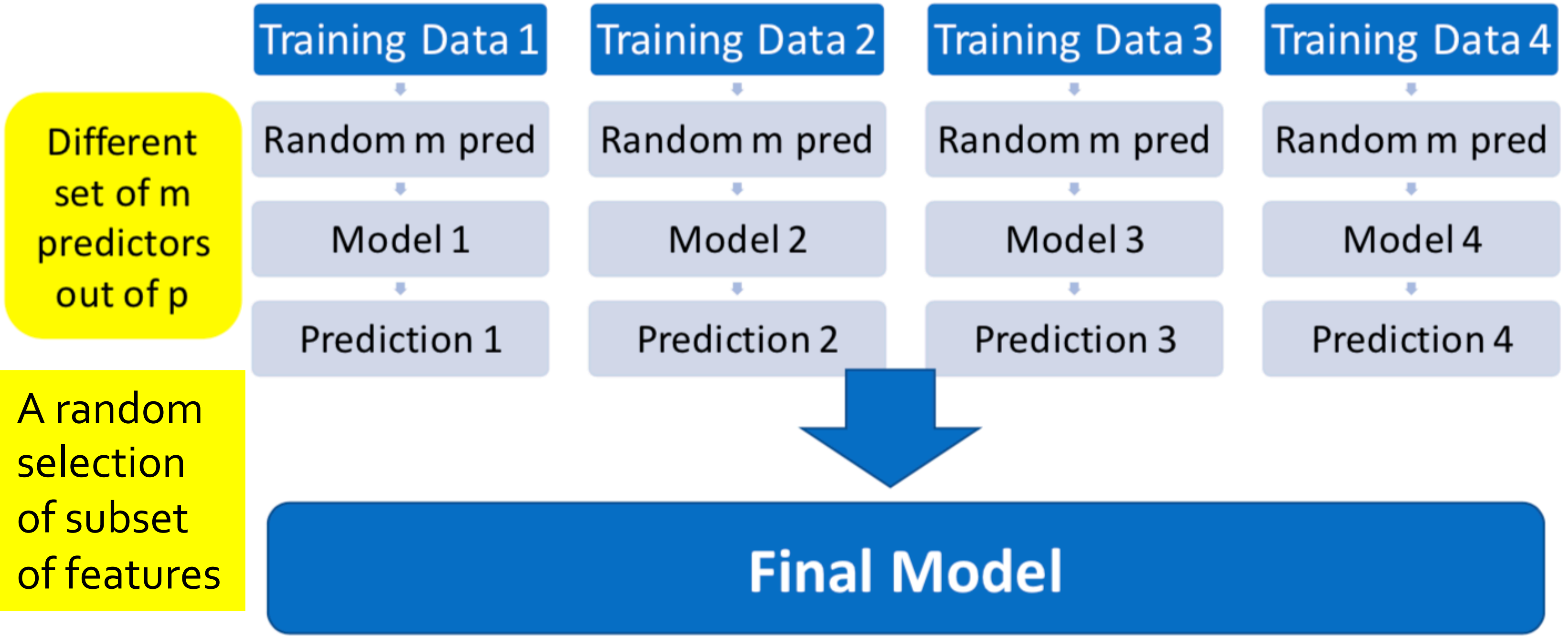



```
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC

log_clf = LogisticRegression()
rnd_clf = RandomForestClassifier()
svm_clf = SVC()

voting_clf = VotingClassifier(
    estimators=[('lr', log_clf), ('rf', rnd_clf), ('svc',
svm_clf)],
    voting='hard')
voting_clf.fit(X_train, y_train)
```

Primitive Type of Bagging is just Random: Its Shortcomings

Concept:-
We use subset of predictor variables so that we get different splits in each model

A random selection of subsample of data



Boosting Algorithm: Different kinds of Random Forest

Ada Boosting

Gradient Boosting

- LightGBM

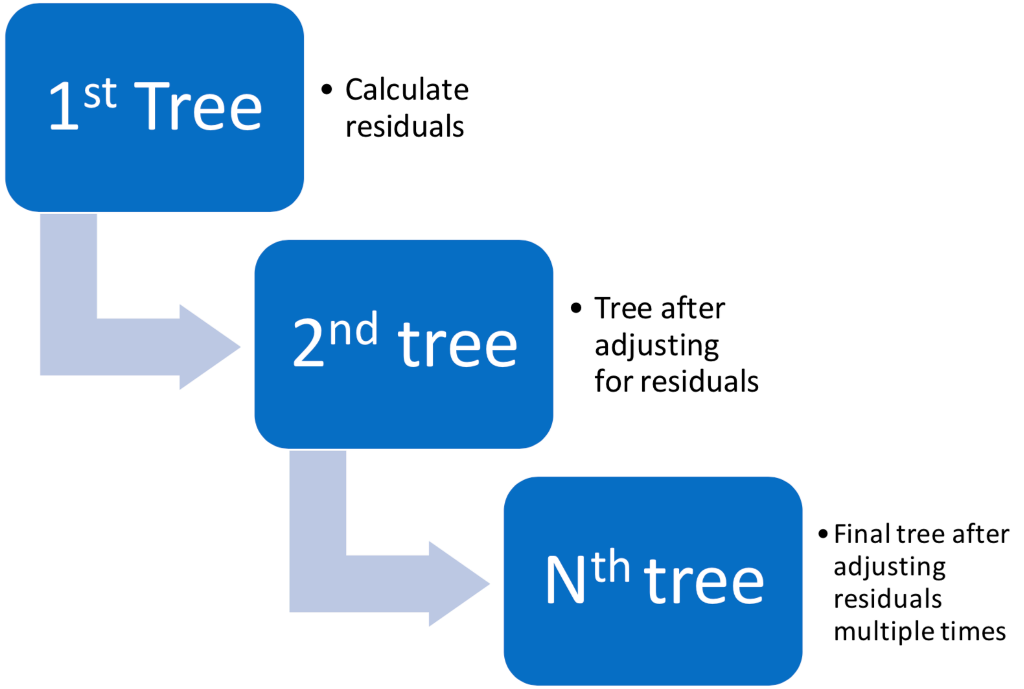
XGboost

Boosting

Process of turning a weak leaner into a strong learner

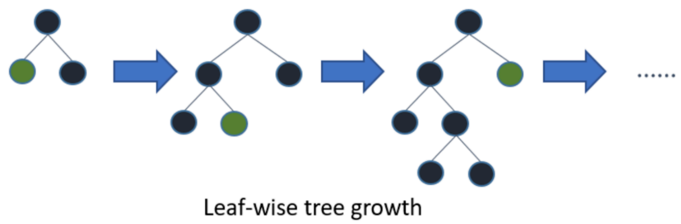
There are three algorithms of boosting Gradient Boost, Ada Boost and XG Boost based on different methodologies.

Gradient Boosting

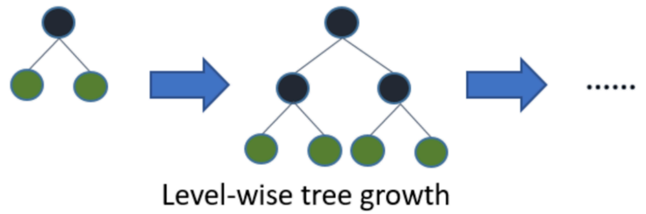


Another very popular Boosting algorithm is Gradient Boosting . Just like AdaBoost, Gradient Boosting works by sequentially adding predictors to an ensemble, each one correcting its predecessor. However, instead of tweaking the instance weights at every iteration like AdaBoost does, this method tries to fit the new predictor to the residual errors made by the previous predictor.

LightGBM



Explains how LightGBM works



How other boosting algorithm works

- Gradient Boosting works by sequentially adding predictors to an ensemble, each one correcting its predecessor. However, instead of tweaking the instance weights at every iteration like AdaBoost does, this method tries to fit the new predictor to the residual errors made by the previous predictor.

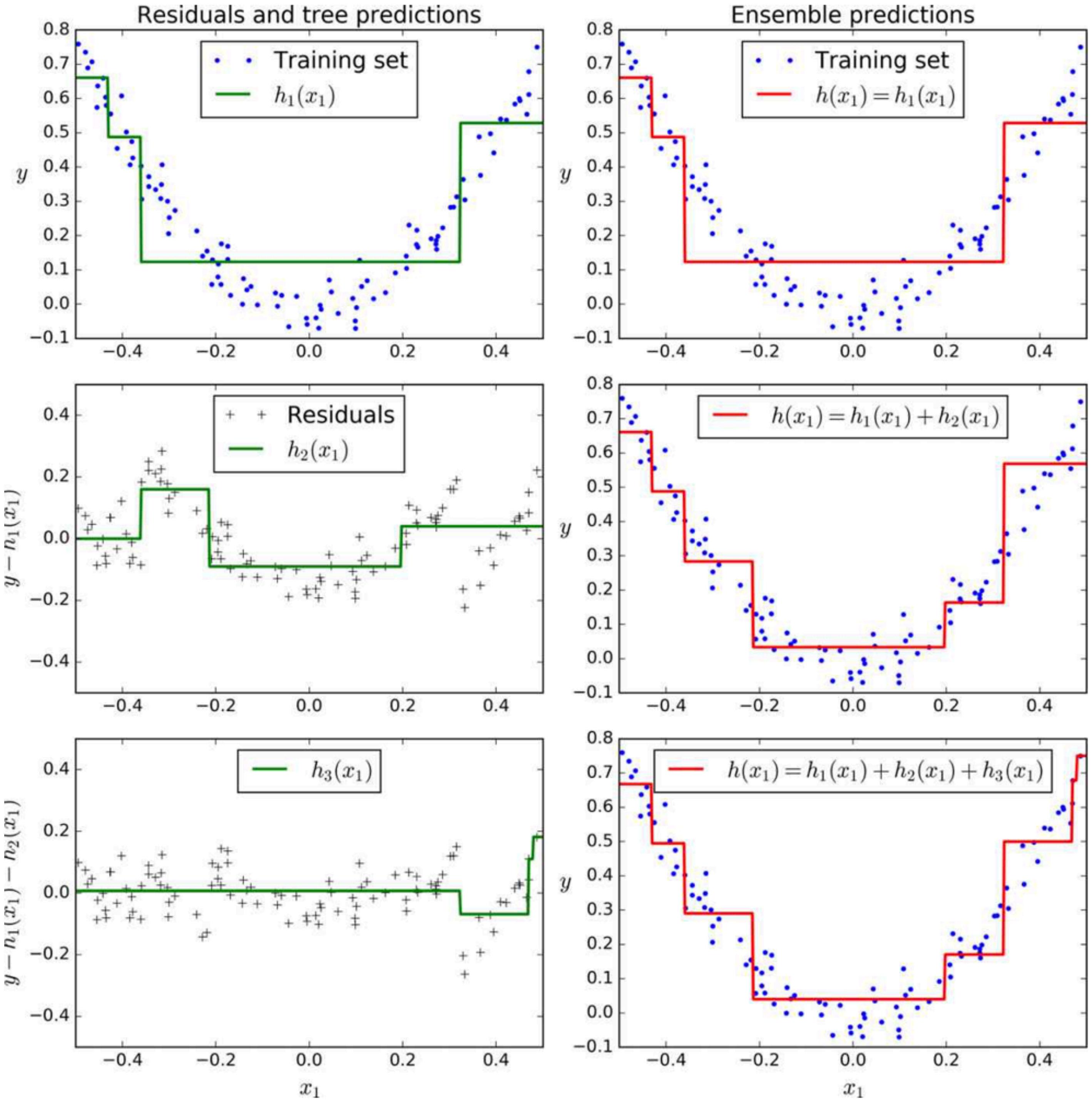
- LightGBM is a new gradient boosting tree framework, which is highly efficient and scalable and can support many different algorithms including GBDT, GBRT, GBM, and MART. LightGBM is evidenced to be several times faster than existing implementations of gradient boosting trees, due to its fully greedy tree-growth method and histogram-based memory and computation optimization.

Data Preprocessing and Model Building & Training

$ conda install -c conda-forge lightgbm

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


# Importing the dataset
dataset = pd.read_csv('...input\\Social_Network_Ads.csv')
X = dataset.iloc[:, [2, 3]].values
y = dataset.iloc[:, 4].values


# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size
= 0.25, random_state = 0)


# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

import lightgbm as lgb


d_train = lgb.Dataset(x_train, label=y_train)


params = {}
params['learning_rate'] = 0.003
params['boosting_type'] = 'gbdt'
params['objective'] = 'binary'
params['metric'] = 'binary_logloss'
params['sub_feature'] = 0.5
params['num_leaves'] = 10
params['min_data'] = 50
params['max_depth'] = 10


clf = lgb.train(params, d_train, 100)
```

Gradient Boosting in Regression



Ada Boosting

To build an AdaBoost classifier, a first base classifier (such as a Decision Tree) is trained and used to make predictions on the training set. The relative weight of misclassified training instances is then increased. A second classifier is trained using the updated weights and again it makes predictions on the training set, weights are updated, and so on

Ada Boosting

1^{st} Tree

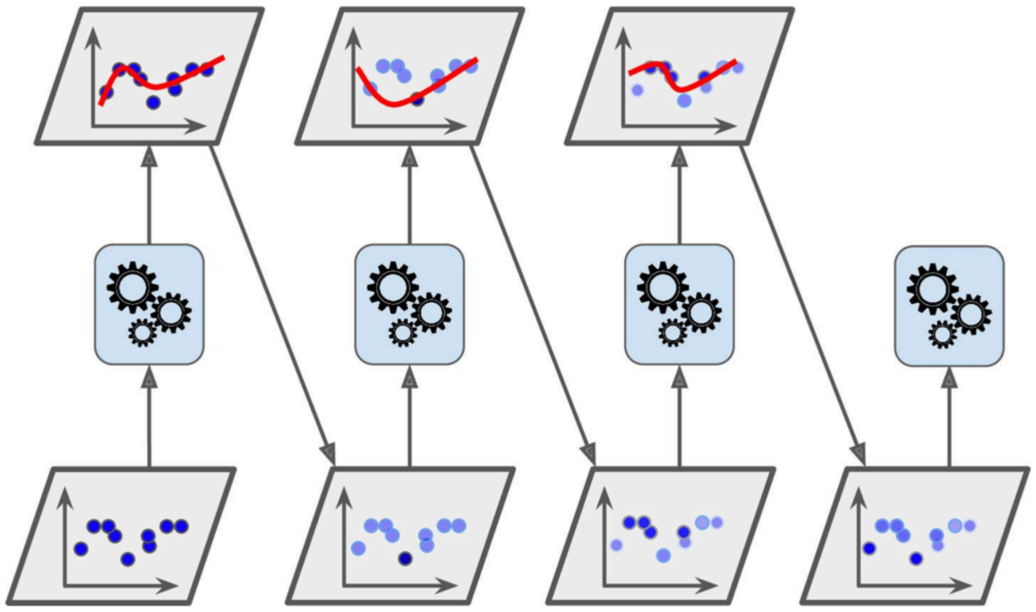
- Assign more weightage to misclassified observations

2^{nd} tree

- Retrain the tree after accounting for weightages

N^{th} tree

- Final tree after accounting for weightages N-1 Times



Adaboost:

Weighted Error Rate of the j^th Predictor, Predictor Weight and the Update

$$r_j = \frac{\displaystyle\sum_{\substack{i=1 \\ \hat{y}_j^{(i)} \neq y^{(i)}}}^{m} w^{(i)}}{\displaystyle\sum_{i=1}^{m} w^{(i)}} \quad \text{where } \hat{y}_j^{(i)} \text{ is the } j^{\text{th}} \text{ predictor's prediction for the } i^{\text{th}} \text{ instance.}$$

$$\alpha_j = \eta \log \frac{1 - r_j}{r_j}$$

$$\text{for } i = 1, 2, \cdots, m$$

$$w^{(i)} \leftarrow \begin{cases} w^{(i)} & \text{if } \hat{y}_j^{(i)} = y^{(i)} \\ w^{(i)} \exp(\alpha_j) & \text{if } \hat{y}_j^{(i)} \neq y^{(i)} \end{cases}$$

To make predictions, AdaBoost simply computes the predictions of all the predictors and weighs them using the predictor weights α j . The predicted class is the one that receives the majority of weighted votes

XGBoosting

Salient features of XGBoost which make it different from other gradient boosting algorithms include:

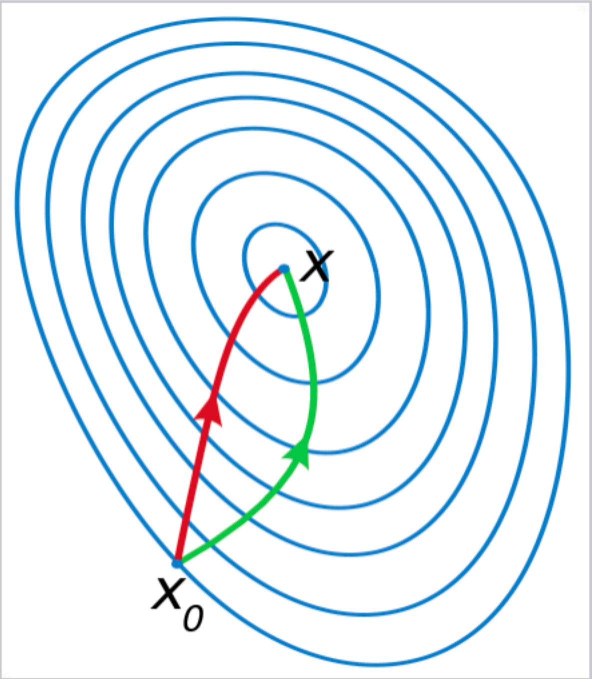
- Clever penalization of trees
- A proportional shrinking of leaf nodes
- Newton Boosting
- Extra randomization para meter

Almost similar to Gradient Boost

- XG-boost used a more regularized model formalization to control over-fitting, which gives it better performance.
- For model, it might be more suitable to be called as regularized gradient boosting.

Regularization

- The cost function we are trying to optimize (MSE in regression etc) also contains a penalty term for number of variables. In a way, we want to minimize the number of variables in final model along with the MSE or accuracy. This helps in avoiding overfitting
- XG-Boost contains regularization terms in the cost function.



A comparison of gradient descent (green) and Newton's method (red) for minimizing a function (with small step sizes). Newton's method uses curvature information (i.e. the second derivative) to take a more direct route.