Introduction to XGboost

FU Hanlin
fixed-term.hanlin.fu@bosch.com
AE/MFP2.1-CN
April 10, 2019

Outline

1. **Unsupervised Learning**
2. **Supervised Learning**
3. **Ensemble Learning**
4. **XGboost**

1. Unsupervised Learning

Unsupervised Learning

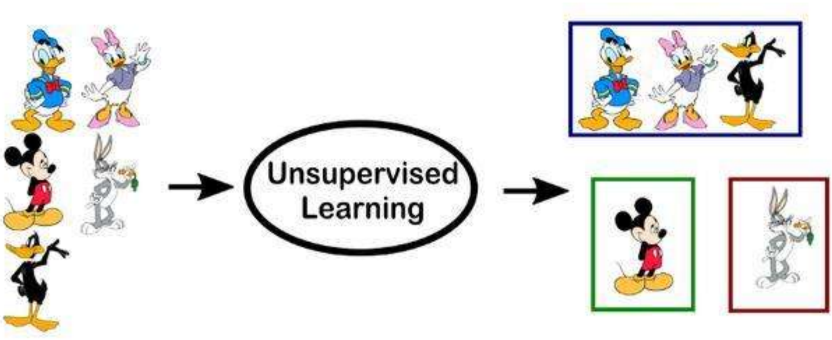


Figure: Unsupervised Learning(e.g., Mickey Mouse and Donald Duck)

Clustering

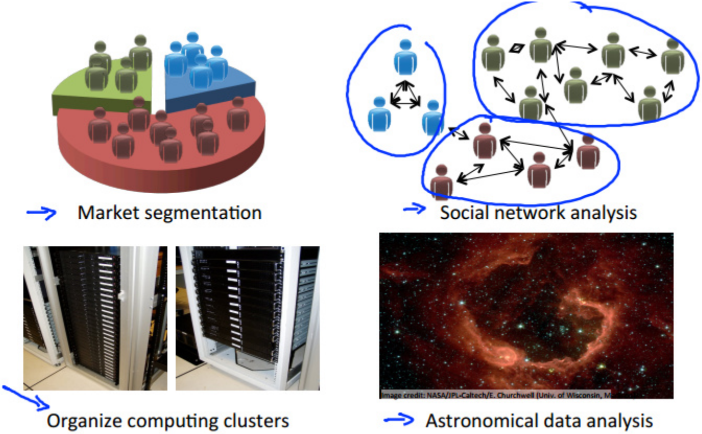


Figure: Clustering examples

K-means

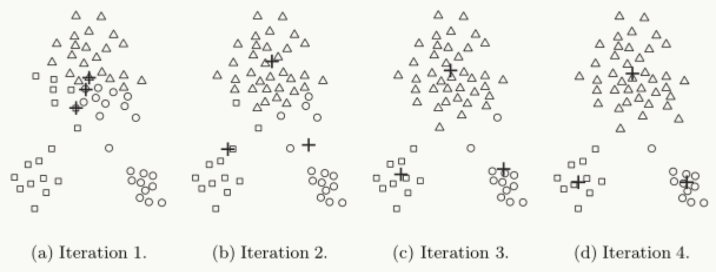


Figure: K-means Iteration(3 clusters)

K-means algorithm

Input:

- K(number of clusters)
- Training set$x_{(1)}, x_{(2)}, \odot, x_{(m)}$

2. Supervised Learning

Classification

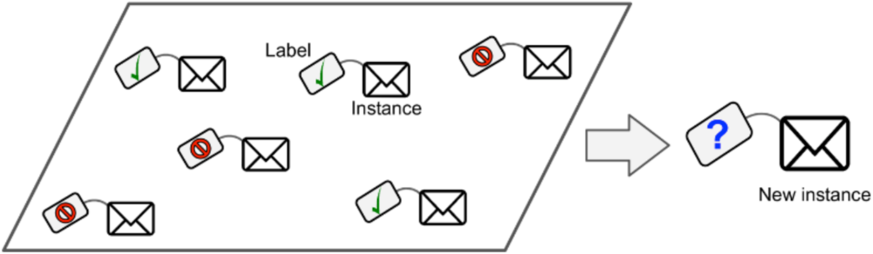


Figure: A labeled training set for supervised learning(e.g.,spam classficiation)

Regression

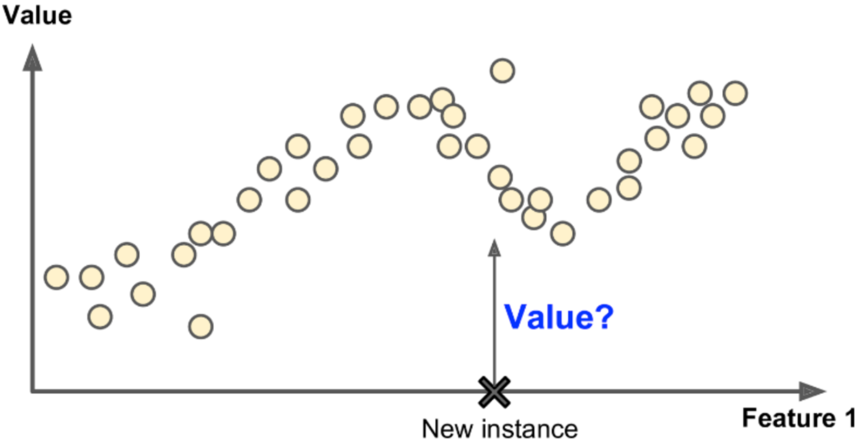


Figure: Regression

Classification Tree

Classification tree analysis is when the predicted outcome is the class (discrete) to which the data belongs.

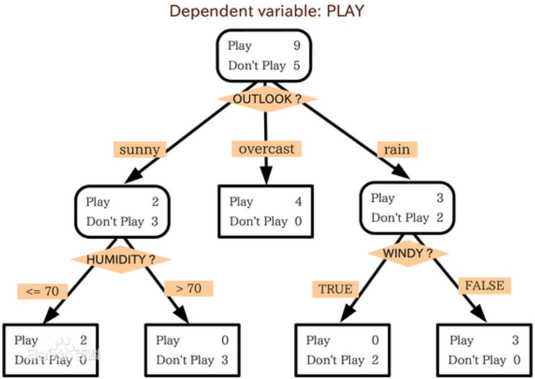


Figure: A decision tree with binary splits for classification

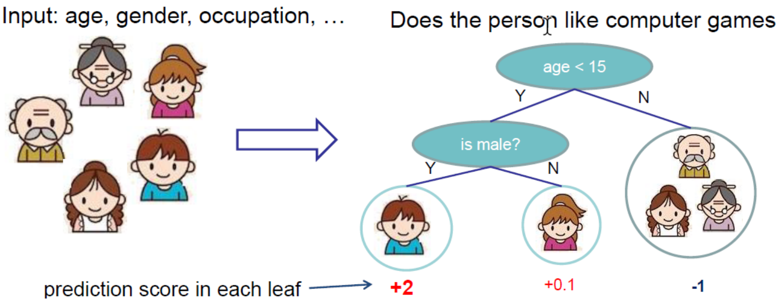
How to split

generative algorithm	categorization
$ID3$	information entropy
$C4.5$	gain ratio
$CART$	Gini index

Regression Tree(CART)

Regression tree analysis is when the predicted outcome can be considered a real number.

- regression tree (also known as classification and regression tree)
- contains one score in each leaf value
- base learner



How to predict

Step 1: Choose split point $\sum(y_i - f(x_i))^2$

- Area1:$R_1(j,s) = \{x|x^{(j)} \le s\}$
- Area2:$R_2(j,s) = \{x|x^{(j)} > s\}$

Step 2:Minize loss function

- $\min\limits_{c_1} \sum\limits_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min\limits_{c_2} \sum\limits_{x_i \in R_1(j,s)} (y_i - c_2)^2$

Step 3:Recursively call Step 1 and Step 2 in subareas

Step 4:Output predicted value

- $c_1 = ave(y_i|x_i \in R_1(j,s))$
- $c_2 = ave(y_i|x_i \in R_2(j,s))$

3. Ensemble Learning

Ensemble Theory

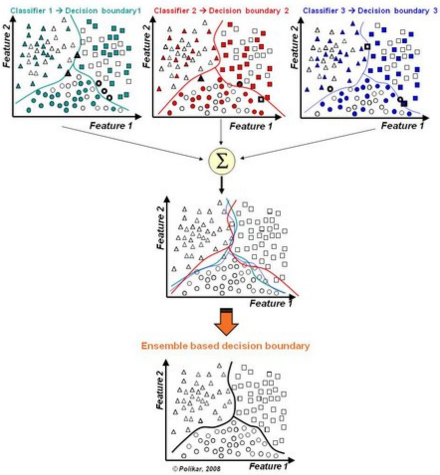


Figure: Model stacking

Model Combination

$\sum$ in previous slide

- averaging
  1. simple averaging: $H(x) = \frac{1}{T}\sum_{i=1}^{T} h_i(x)$
  2. weighted averaging: $H(x) = \sum_{i=1}^{T} \omega_i h_i(x)$
- voting
  1. majority voting
  2. plurality voting
  3. weighted voting
- stacking

Bagging and Boosting

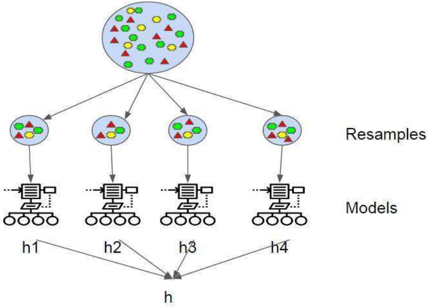


Figure: Bagging

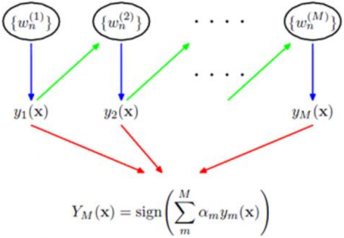


Figure: Bootstrap Aggregating

4. XGboost

eXtreme Gradient Boosting Objective

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta)$$

$L(\Theta)$ Trainning Loss, measures how well model fit on training data
$\Omega(\Theta)$ Regularization, measures complexity of model

- Optimizing **training loss** encourages **predictive** models
  Fitting well in training data at least get you close to training data which is hopefully close to the underlying distribution.
- Optimizing **regularization** encourages **simple** models
  Simpler models tends to have smaller variance in future predictions, making prediction **stable**

Learning Step Function(visually)

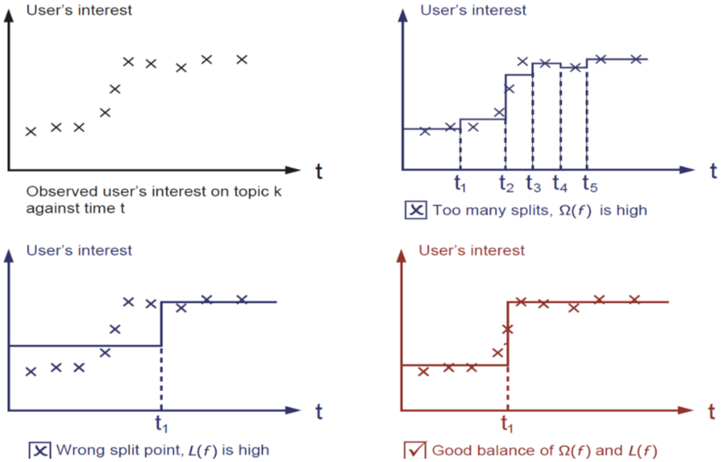


Figure: $L(f)$ VS $\Omega(f)$

How do we learn?

- Objective: $\sum_{i=1}^{n} l(\widehat{y}_i, y) + \Omega(f_k), \Omega(f_k) = \gamma T + \frac{1}{2}\lambda||w||^2, f_k \in F$
- Solution: Boosting

  Start from constant prediction,add a new function each time

  $\hat{y}_i^{(0)} = 0$

  $\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i)$

  $\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i)$

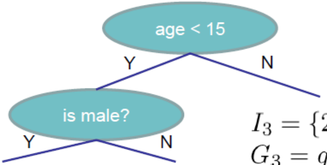
  $\dots$

  $\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)$
- Additive training
- Structure score: $Obj = -\frac{1}{2}\sum_{j=1}^{T} \frac{G_j^2}{H_j+\lambda} + \gamma T$

  $G_j = \sum_{i \in I_j} \partial_{\hat{y}(t-1)} l(y_i, \hat{y}_{(t-1)}), H_j = \sum_{i \in I_j} \partial^2_{\hat{y}(t-1)} l(y_i, \hat{y}_{(t-1)})$, T is number of leaves, w are leaf scores

The Stucture Score Calculation

Instance index	gradient statistics
1	g1, h1
2	g2, h2
3	g3, h3
4	g4, h4
5	g5, h5



$$I_1 = \{1\} \quad G_1 = g_1 \quad H_1 = h_1$$

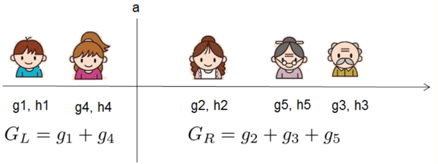
$$I_2 = \{4\} \quad G_2 = g_4 \quad H_4 = h_4$$

$$I_3 = \{2, 3, 5\} \quad G_3 = g_2 + g_3 + g_5 \quad H_3 = h_2 + h_3 + h_5$$

$$Obj = -\sum_j \frac{G_j^2}{H_j + \lambda} + 3\gamma$$

The smaller the score is, the better the structure is

How to split?



In practice, we grow the tree greedily

- Start from tree with depth 0
- For each leaf node of the ree, try to add a split.The change of objective after adding the split is

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$$

γ means the complexity cost by introducing additional leaf

References I

[1] Tianqi Chen's Homepage
`https://homes.cs.washington.edu/~tqchen/`

[2] Chen and C Gusterin. XGBoost: A Scalable Tree Boosting System. In KDD 16ăProceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, page 785-794.

[3] Apache
`http://xgboost.apachecn.org/cn/latest/model.html`