

Tree-based Methods (Cont.)

Petchara Pattarakijwanich

Introduction to Data Science, 11 November 2022

Goal of this week

- Binary Tree
 - Regression Tree
 - Classification Tree
- Purity Metrics
 - Classification Error Rate
 - Gini Index
 - Entropy
- Methods to Improve Binary Tree
 - Bagging
 - Random Forest
 - Boosting

About the Final Project

Approaches

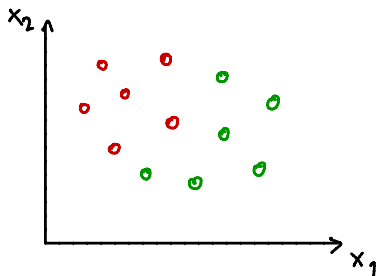
- Answer specific questions from data, using data science techniques
- Implement, invent, or test some data science techniques or algorithms

Data Source

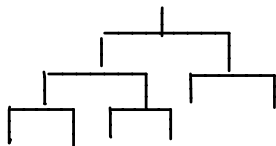
- Your own research
- Publicly available datasets
<https://geekflare.com/open-datasets-for-data-science/>
<https://www.dataquest.io/blog/free-datasets-for-projects/>
- Simulated data (?)

Presentation (some time around final week or a bit later)

Recap of Binary Trees



↓ ឆ្លើយ



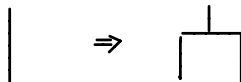
Input : $\{x_1, \dots, x_n\}$ output : y

Grow tree

① ជំពូកដំបូងបង្កើតជាជំពូក "ឆ្លើយ"

- Regression \Rightarrow MSE គណនា
- Classification \Rightarrow Information Gain

\swarrow Error Rate
 \downarrow Gini
 \searrow Entropy



② បំបែក ឧទាហរណ៍

③ ឧបទ្វីប \rightarrow ជំពូក ឆ្លើយ ឆ្លើយ
 \searrow ជំពូក depth

"Binary Tree is deterministic"

Recap of Binary Trees

ปัญหามี

① Variance สูง

② Accuracy ต่ำ

การวัดความไม่แน่นอน

⇒ ความแปรปรวน

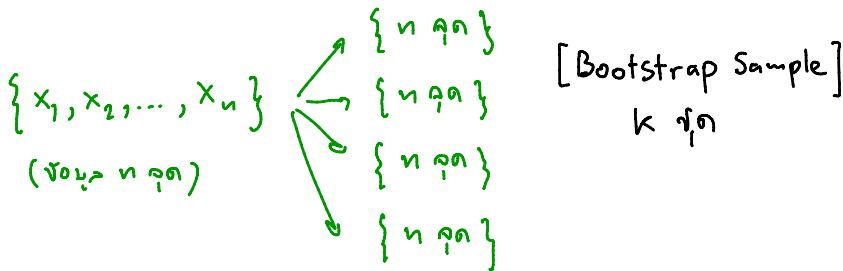
⇒ Variance สูง

[101000 Bagging + Random Forest]

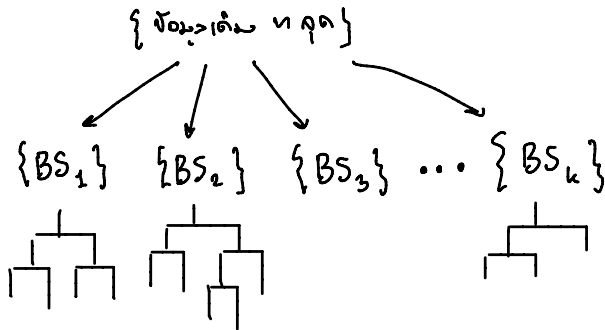
Bagging

Bagging = Bootstrap AGGregation

Bootstrap = နမူနာယူပုံစံတူတူ ချုပ်ဆိုပုံစံတူတူ
(နမူနာယူပုံစံတူတူ \Rightarrow ဘာတူတူ)



Bagging



← k ชุดข้อมูล-จำลอง
(เพื่อลด over fit)

ปัญหา: Variance ของ
ข้อมูลต้นฉบับ
สูงเกินไป BS correlations
(เมื่อ data มี noise)

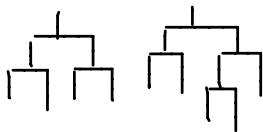
Prediction = "การเฉลี่ย" ของผลลัพธ์ k ครั้ง \Rightarrow Variance ของ
(regression = mean
classification = mode)

Random Forest

↓
↓

{ 1000 random P }
↓

{ BS_1 } { BS_2 } ...



Prediction : 1000

Input n data $\{x_1, \dots, x_n\}$

↓
↓

↓
↓

↓
↓

⇒ decorrelate data

⇒ 1000 → variance 20

($m \approx \sqrt{n}$ 1000 data
 $m = n \Rightarrow$ Bagging)

Boosting

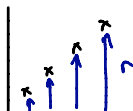
Goal: $y = f(x)$



in Boosting is LR
(weak classifier with Linear model
& additive property of it)

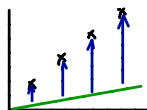
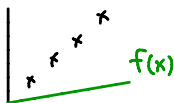
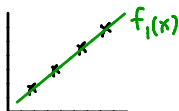
① Initializing
 $f(x) = 0$

\Rightarrow Residual
 $r = y - f(x) = y$



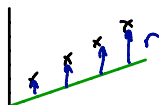
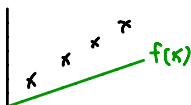
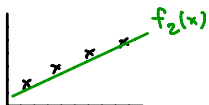
choose the
Learning Rate
 $\lambda \approx 0.01$

② Next step: $r = f_1(x) \Rightarrow f(x) = f(x) + \lambda f_1(x) \Rightarrow r = r - \lambda f_1(x)$



Boosting

③ $\text{weak classifier } \text{let } r = f_2(x) \Rightarrow f(x) = f(x) + \lambda f_2(x) \Rightarrow r = r - \lambda f_2(x)$



④ $\text{weak classifier } \left\{ \begin{array}{l} \text{let } r = f_i(x) \\ \text{update model } f(x) = f(x) + \lambda f_i(x) \\ \text{update residual } r = r - \lambda f_i(x) \end{array} \right\}$

$\text{weak classifier } \Rightarrow f(x) \text{ is a weak classifier}$

$\Rightarrow r \text{ is a weak classifier}$


$\lambda = \text{learning rate}$
 the "weight" of the weak classifier
 the "weight" of the weak classifier

$\left\{ \begin{array}{l} \lambda = 1 \text{ "hard fit" } \Rightarrow \text{overfit} \\ \lambda \text{ low "soft fit" } \Rightarrow \text{overfit} \end{array} \right.$

Boosting

Boosting as Binary Tree

① $f(x_1, \dots, x_n) = 0$, $r = y - f(x) = y$

② สร้าง binary tree จาก residual $\Rightarrow f_i(x)$ 

\Rightarrow update $f(x) = f(x) + \lambda f_i(x)$

(λ ให้น้อยๆ ดีนะ)

\Rightarrow update $r = r - \lambda f_i(x)$

(λ น้อยๆ ดีนะ)

\Rightarrow ทำซ้ำ

③ พอจบการทำซ้ำ $f(x) \approx y$, $r \approx 0$

λ น้อย \nearrow ดีกว่า

\searrow 1. overfit (ไม่ดีนะ)