Lecture 7

Classification Analysis and Ensemble Models

Ashish Dandekar

Lecture Overview

Naive Bayes Classifier

Decision Trees

Ensemble Models

Naive Bayes Classifier

Formulation

Age	Edu	Marital	Income	Credit
23	Masters	Single	75k	Yes
35	Bachelor	Married	50k	No
26	Masters	Single	70k	Yes
41	PhD	Single	95k	Yes
18	Bachelor	Single	40k	No
55	Masters	Married	85k	No
30	Bachelor	Single	60k	No
35	PhD	Married	60k	Yes
28	PhD	Married	65k	Yes

- Will a 24 years old unmarried bachelor with an annual income of 50k get the credit card approved?
- Will a 45 years old single PhD person with the annual income of 95 k be eligible for the credit card?

Can we use logistic regression to solve this problem?

Formulation

The Problem

Given a labeled dataset $\mathsf{D} = \{(x_i, y_i)\}$ of n points where $\mathbf{x_i}$s are the predictors and $y_i \in \mathsf{C}$ is the label of the datapoint. $\mathsf{C} = \{c_1, c_2, .., c_l\}$ is the set of labels.

A datapoint $\mathbf{x}$ is assigned a label as follows:

$$\hat{y} = \arg\max_{c \in \mathsf{C}} Pr[y = c \mid \mathbf{x}]$$

Using Baye's Rule,

$$Pr[y = c \mid \mathbf{x}] \propto Pr[\mathbf{x} \mid y = c] Pr[y = c]$$

How to compute $Pr[\mathbf{x} \mid y = c]$?

Conditional Independence

Conditional Independence

Given three random variables X, Y and Z. X and Y are said to be conditionally independent given Z if and only if:

$$Pr[X \mid Y, Z] = Pr[X \mid Z]$$

X and Y may or may not be indepdendent variables.

Confounding variable!

There are two friends who are going for a party. Both of them are late. Let's consider their tardiness as a random variable. Are these dependent on each other? What if it was raining outside?

Conditional Independence

Naive Bayes Assumption

All predictors are conditionally independent given the label of the datapoint.

$$Pr[\mathbf{x} \mid y = c] = Pr[x_1 \mid y = c] Pr[x_2 \mid y = c] \dots Pr[x_d \mid y = c]$$

Applying to the earlier example:

$$\begin{aligned}
&Pr[credit = yes \mid age = 24, edu = bachelor, income = 50k] \\
=&Pr[age = 24, edu = bachelor, income = 50k \mid credit = yes] \\
&\times Pr[credit = yes] \\
=&Pr[age = 24 \mid credit = yes] \times Pr[edu = bachelor \mid crdit = yes] \times \\
&\times Pr[income = 50k \mid crdit = yes] \times Pr[credit = yes]
\end{aligned}$$

Conditional Independence

Pros

- Works with non-numerical data.
- Not sensitive to the outliers in the data.
- Easy to implement and highly interpretable.

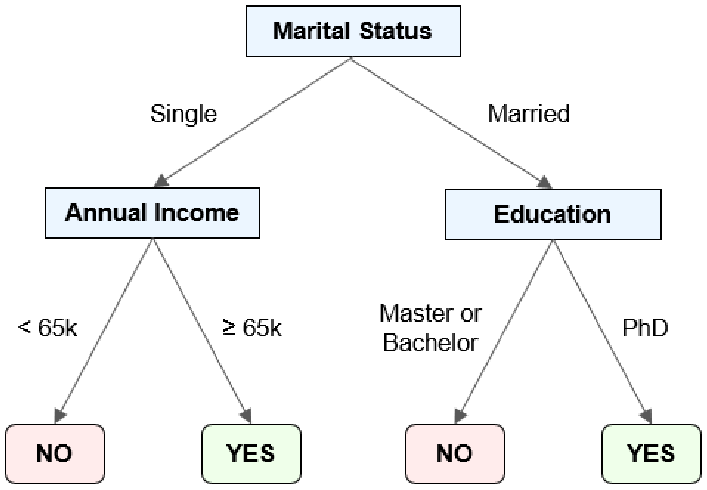
Cons

- Conditional independence may not be a realistic assumption.
- Sensitive to imbalanced data.
- Suffers from *black-swan events*.

Decision Trees

Introduction

Age	Edu	Marital	Income	Credit
23	Masters	Single	75k	Yes
35	Bachelor	Married	50k	No
26	Masters	Single	70k	Yes
41	PhD	Single	95k	Yes
18	Bachelor	Single	40k	No
55	Masters	Married	85k	No
30	Bachelor	Single	60k	No
35	PhD	Married	60k	Yes
28	PhD	Married	65k	Yes

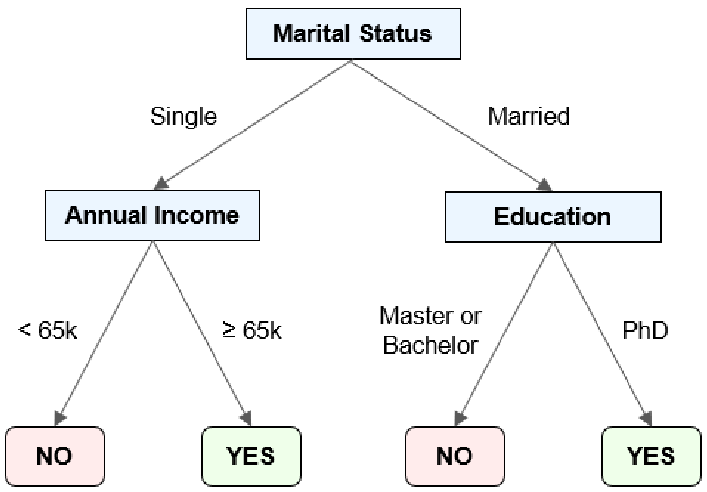


Components of Decision Tree

- **Inner node.** - test a feature
- **Leaf node.** - label

Which tree to use?

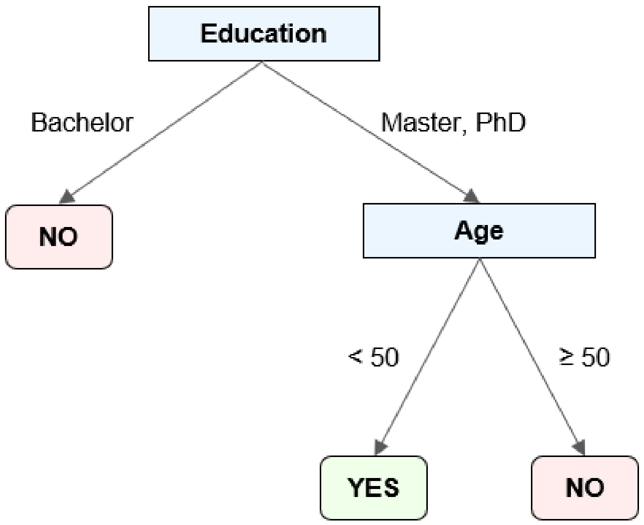
Age	Edu	Marital	Income	Credit
23	Masters	Single	75k	Yes
35	Bachelor	Married	50k	No
26	Masters	Single	70k	Yes
41	PhD	Single	95k	Yes
18	Bachelor	Single	40k	No
55	Masters	Married	85k	No
30	Bachelor	Single	60k	No
35	PhD	Married	60k	Yes
28	PhD	Married	65k	Yes



Will the credit card be approved for a person with the following attributes: Age:50, Education: PhD, Marital_Status: Singsle and Income:70k?

Which tree to use?

Age	Edu	Marital	Income	Credit
23	Masters	Single	75k	Yes
35	Bachelor	Married	50k	No
26	Masters	Single	70k	Yes
41	PhD	Single	95k	Yes
18	Bachelor	Single	40k	No
55	Masters	Married	85k	No
30	Bachelor	Single	60k	No
35	PhD	Married	60k	Yes
28	PhD	Married	65k	Yes



Will the credit card be approved for a person with the following attributes: Age:50, Education: PhD, Marital_Status: Single and Income:70k?

Building the tree!

General Procedure

- Start at the root node with all records.
- If all records in the node has the same label - it is a leaf node.
- Otherwise, "choose" a test (feature and condition) to split the node into smaller subsets.
- Recursively apply the procedure on each of the nodes.

Building the tree!

Age	Edu	Marital	Income	Credit
23	Masters	Single	75k	Yes
35	Bachelor	Married	50k	No
26	Masters	Single	70k	Yes
41	PhD	Single	95k	Yes
18	Bachelor	Single	40k	No
55	Masters	Married	85k	No
30	Bachelor	Single	60k	No
35	PhD	Married	60k	Yes
28	PhD	Married	65k	Yes

Building the tree!

Age	Edu	Marital	Income	Credit
23	Masters	Single	75k	Yes
35	Bachelor	Married	50k	No
26	Masters	Single	70k	Yes
41	PhD	Single	95k	Yes
18	Bachelor	Single	40k	No
55	Masters	Married	85k	No
30	Bachelor	Single	60k	No
35	PhD	Married	60k	Yes
28	PhD	Married	65k	Yes

Age	Edu	Marital	Income	Credit
23	Masters	Single	75k	Yes
35	Bachelor	Married	50k	No
26	Masters	Single	70k	Yes
41	PhD	Single	95k	Yes
18	Bachelor	Single	40k	No
55	Masters	Married	85k	No
30	Bachelor	Single	60k	No
35	PhD	Married	60k	Yes
28	PhD	Married	65k	Yes

Building the tree!

General Procedure

- Start at the root node with all records.
- If all records in the node has the same label - it is a leaf node.
- Otherwise, *"choose" a test* (feature and condition) to split the node into smaller subsets.
- Recursively apply the procedure on each of the nodes.

How to choose a feature? How to handle non-binary attributes?

Purity Metrics

AAAAAAAA	AAAABBCD	AABBCCDD
Bucket 1	Bucket 2	Bucket 3
Low Entropy	**Medium Entropy**	**High Entropy**

Entropy

$$H(X) = -\sum_{x} p(x) \log p(x)$$

Higher randomness leads to higher entropy, ergo to higher impurity!

	Distribution	Entropy
Bucket 1	$[8, 0, 0, 0]$	0
Bucket 2	$[4, 2, 1, 1]$	1.75
Bucket 3	$[2, 2, 2, 2]$	2

Purity Metrics

AAAAAAAA	AAAABBCD	AABBCCDD
Bucket 1	Bucket 2	Bucket 3
Low Entropy	**Medium Entropy**	**High Entropy**

Gini Index

$$G(X) = 1 - \sum_{x} P(x)^2$$

Higher randomness leads to higher gini index, ergo to higher impurity!

	Distribution	**Gini**
Bucket 1	$[8, 0, 0, 0]$	0
Bucket 2	$[4, 2, 1, 1]$	0.66
Bucket 3	$[2, 2, 2, 2]$	0.75

Informtion Gain

Assume that a node u is split into k children $(v_1, v_2, ..., v_k)$ such that

- n_i is the number of records in n^{th} child.
- n is the total number of records.
- $I(\cdot)$ is the impurity of the node.

Information gain is computed as:

$$IG = I(u) - \sum_{i=1}^{k} \frac{|\ v_i\ |}{|\ u\ |} I(v_i)$$

Computing Information Gain

Age	Edu	Marital	Income	Credit
23	Masters	Single	75k	Yes
35	Bachelor	Married	50k	No
26	Masters	Single	70k	Yes
41	PhD	Single	95k	Yes
18	Bachelor	Single	40k	No
55	Masters	Married	85k	No
30	Bachelor	Single	60k	No
35	PhD	Married	60k	Yes
28	PhD	Married	65k	Yes

Entropy before splitting:

$$H[(5,4)] = 0.99$$

Splitting on Marital Status.

- $H[v_1] = H[(2,2)] = 1$
- $H[v_2] = H[(3,2)] = 0.97$

Information gain is computed as follows:

$$IG = 0.99 - (\frac{4}{9} + \frac{5}{9} \cdot 0.97) = 0.006$$

Computing Information Gain

Age	Edu	Marital	Income	Credit
23	Masters	Single	75k	Yes
35	Bachelor	Married	50k	No
26	Masters	Single	70k	Yes
41	PhD	Single	95k	Yes
18	Bachelor	Single	40k	No
55	Masters	Married	85k	No
30	Bachelor	Single	60k	No
35	PhD	Married	60k	Yes
28	PhD	Married	65k	Yes

Entropy before splitting:

$$H[(5,4)] = 0.99$$

Splitting on Educational Status.

- $H[v_1] = H[(0,3)] = 0$
- $H[v_2] = H[(5,1)] = 0.65$

Information gain is computed as follows:

$$IG = 0.99 - (\frac{3}{9} \cdot 0 + \frac{6}{9} \cdot 0.65) = 0.556$$

Comments

Splitting non-binary attributes.

- Multiway split
- Binary split by merging categories

Splitting continuous attributes.

- Try every possible value.
- Sort the data and use the quantiles as a a splitting criteria.

Optimality

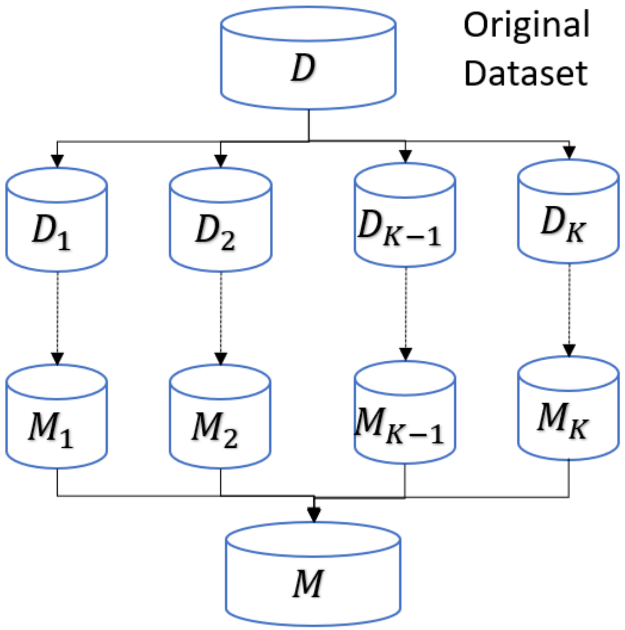
- Finding optimal decision tree is NP-Complete.

Drawbacks

- Prone to overfitting
- Interpretability reduces as the size grows.
- Inefficient on very large atasets.

Ensemble Models

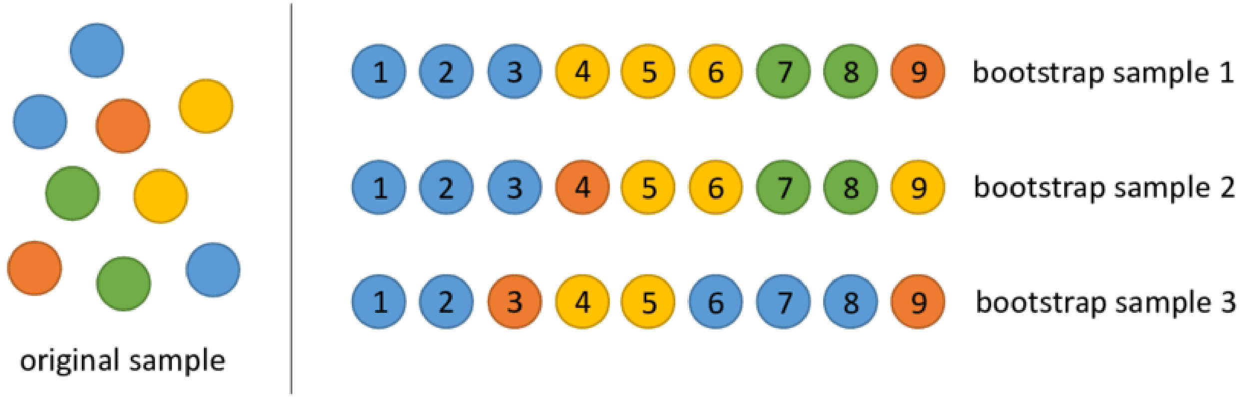
Motivation



The aim of ensemble method is to take a simple, perhaps an inefficient, *base learner* and boost its efficiency through collective efforts.

- Diverse models
- Diverse hyper-parameters
- Diverse input representations
- Diverse traing data

Bagging



Source: ResearchGate

- **Bootstrap Sampling.** Repeatedly drawing samples with replacements.
- **Bagging.** It refers to bootstrap aggregation.

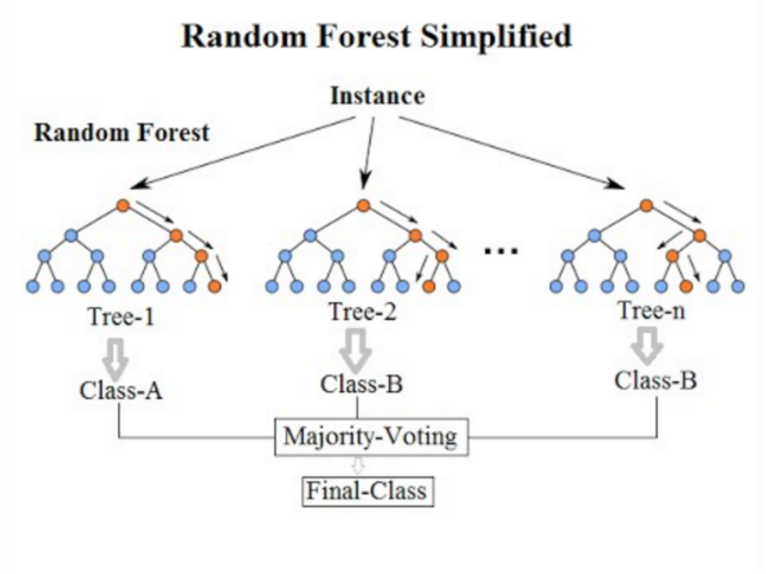
Algorithm

- A learner is said to be *unstable* if a small change in the training data results in large deviation in the output of the learner.
- Bagging works well with unstable base learners.

Algorithm 1 Bagging

for $j \in [1..L]$ **do**
  Create D_j by sampling N points from D with replacement.
  Train a hypothesis h_j on the dataset D_j.
  Put a weight $\alpha_j = 1/L$.
end for
Final classifier is given by $H(x) = sign(\sum_j \alpha_j h_j(x))$

Random Forest



Source: Medium

Heuristics

- Randomly chooses $\sqrt{k}$ features in every bootstrap sample out of k features in the training dataset.

Pros

- High effectiveness (fairly state-of-the-art)
- Parallelisable training
- No fine-tuning necessary

Cons

- Less interpretable

How does it work?

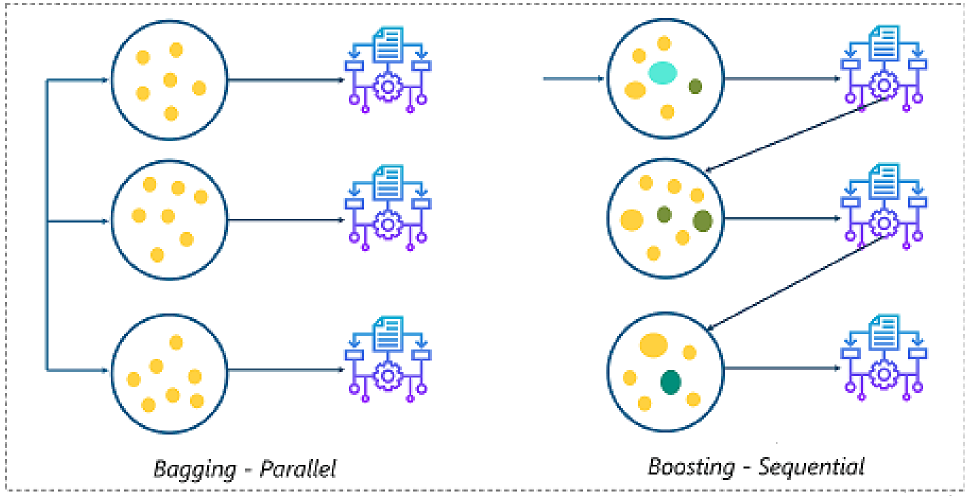
Let $E[h_j]$ and $Var[h_j]$ denote the expected value value for the individual base learners.

$$E[h] = E\left[1/L\sum_j h_j\right] = \frac{1}{L}LE[h_j] = E[h_j]$$

$$Var[h_j] = Var\left[1/L\sum_j h_j\right] = \frac{1}{L^2}LVar[h_j] = \frac{Var[h_j]}{L}$$

- *Bagging reduces the variance of base learners without affecting the base learners.*
- *Unstable learners are overfit and thye already have a lower bias.*

Boosting



Boosting is a seuqential re-training of the *weak* learners by re-weighting the training datapoints based on their classification result.

Boosting

Algorithm 2 AdaBoost

```
for j ∈ [1..L] do
    if j = 1 then
        Initialise weights D_{1,i} = 1/n
    else
        Update weights D_{j,i} ← D_{j−1,i} exp(−α_{j−1} y^i h_{j−1}(x^i))
    end if
    Train hypothesis h_j on dataset D_j.
    Evaluate weighted error ε_j = Σ_i D_{j−1,i} 𝕀(h_j(x^i) ≠ y^i).
    Put a weight α_j = ½ ln((1−ε_j)/ε_j)
end for
Final classifier is given by H(x) = sign(Σ_j α_j h_j(x))
```