Date:

ASSIGNMENT 1 "C"

1. **Bagging and Boosting:**

=> Bagging and boosting in Machine Learning are two types of Ensemble Learning. These two decrease the variance of a single estimate as they combine several estimates from different models. So, the result may be a model with higher stability.

Let's look at both of them in detail.

(a) Bagging:

Bagging, also known as Bootstrap Aggregating, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It decreases the variance and helps to avoid overfitting and is usually applied to decision tree methods.

Description:

Suppose a set D of d tuples, at each iteration i, a training set D_i of d tuples is selected via row sampling with a replacement method (i.e. there

Date:

can be repetitive elements from different d tuples) from D (i.e, bootstrap). Then a classifier model M_i is learned for each training set $D_i < i$. Each classifier M_i returns its class prediction. The bagged classifier M^* counts the votes and assigns the class with the most votes to X.

Implementation Steps of Bagging:

- **Step 1:** Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
- **Step 2:** A base model is created ~~from the original~~ on each of these subsets.
- **Step 3:** Each model is learned in parallel with each training set and independent of each other.
- **Step 4:** The final predictions are determined by combining the predictions from all the models.

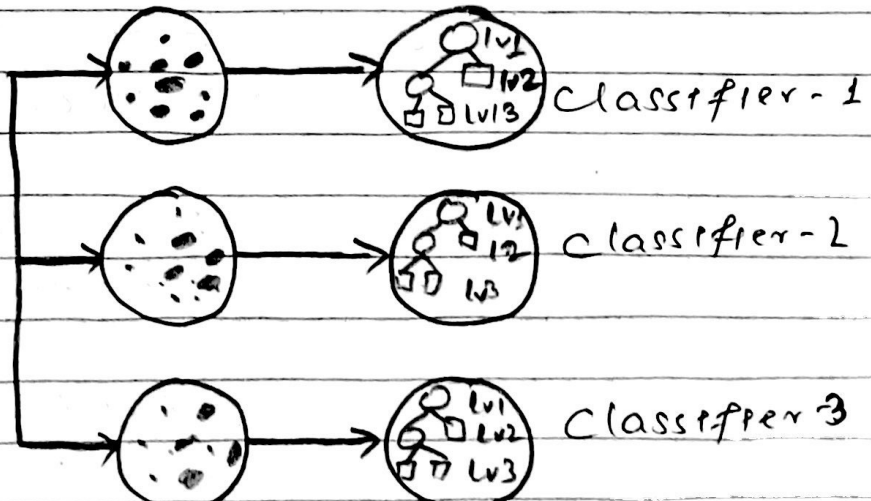

Fig: Bagging (Parallel)

Date:

Example of Bagging:

The Random Forest model uses Bagging, where decision tree models with higher variance are present.

(6) Boosting:

Boosting is an ensemble modeling technique designed to create a strong classifier by combining multiple weak classifiers. The process involves building models sequentially, where each new model aims to correct the errors made by the previous ones.

- Initially, a model is built using the training data.
- Subsequent models are then trained to address the mistakes of their predecessors.
- Boosting ~~aims to~~ assigns weights to the data points in the original dataset.
  - Higher weights: Instances that were misclassified by the previous model receive higher weights.
  - Lower weights: Instances that were correctly classified receive lower weights.
- Training on weighted data: The subsequent model learns from

Date:

weighted dataset, focusing ~~the~~ its attention on harder-to-learn examples.

- This iterative process continues until:
  - The entire training dataset is accurately predicted or,
  - A predefined maximum number of models is reached.

Algorithm:

1. Initialize the dataset and assign equal weight to each of the data point.
2. Provide this as input to the model and identify the wrongly classified data points.
3. Increase the weight of the wrongly classified data points and decrease the weights of correctly classified data points. And then normalize the weights of all data points.
4. If (got required results)
   Go to step 5
   else
   Go to step2
5. End

Date:

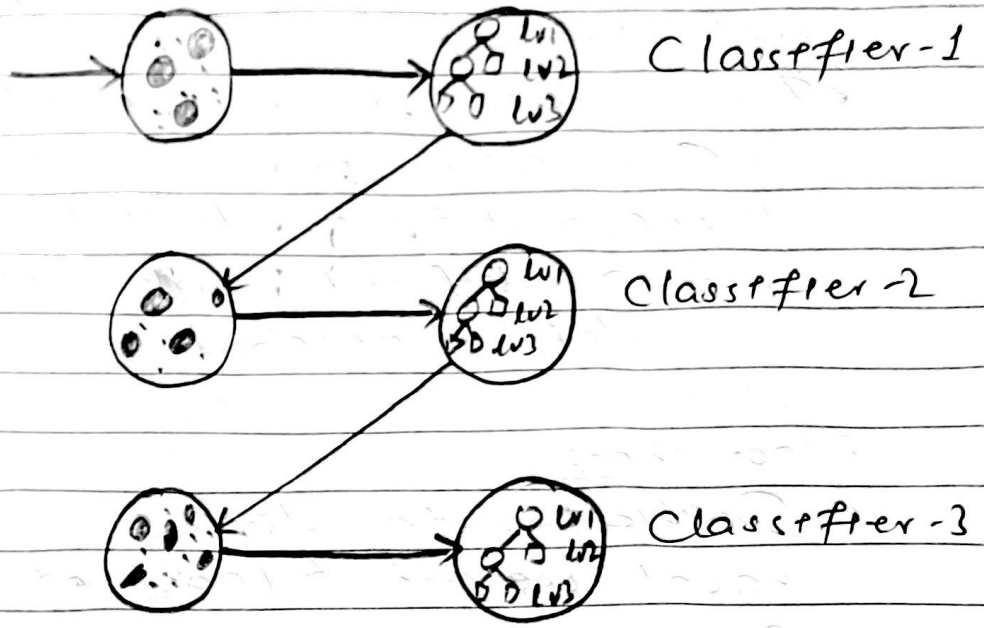

Fig: Boosting (Sequential)

- Differences between boosting and bagging:

SN	Bagging	SN	Boosting
i)	It is the simplest way of combining predictions that belong to the same type.	i)	It is a way of combining predictions that belong to different types.
ii)	It aims to decrease variance, not bias.	ii)	It aims to decrease bias, not variance.
iii)	Each model receives equal weight.	iii)	Models are weighted according to their performances.

Date:

iv)	In this base, classifiers are trained parallelly.	iv)	In this base, classifiers are trained sequentially.
v)	Example: The Random forest.	v)	Example: The AdaBoost

Date:

2. K-Mean

Question:

- A1(2,10), A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4), C1(1,2), C2(4,9)

Initial centroid (Iteration 1):

- A = (2,10)
- B = (5,8)
- C = (1,2)

⇒ Formula used:

for distance = $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

for centroid = $\frac{x}{\Sigma x}, \frac{y}{\Sigma y}$

Point	x	y	dist. to A	dist. to B	dist. to C	Cluster	New Cluster
A1	2	10	0.0	3.60	8.06		A
A2	2	5	5.0	4.24	3.16		C
A3	8	4	8.48	5.0	7.28		B
B1	5	8	3.60	0.0	7.21		B
B2	7	5	7.07	3.60	6.70		B
B3	6	4	7.21	4.12	5.38		B
C1	1	2	8.06	7.21	0.0		C
C2	4	9	2.23	1.41	7.61		B

Date:

Recompute centroid:

A = mean ({(2, 10)}) = (2.0, 10.0)

B = mean ({(8, 4), (5, 8), (7, 5), (6, 4), (4, 9)})
= (6.0, 6.0)

C = mean ({(2, 5), (1, 2)})
= (1.5, 3.5)

Now,

Point	x	y	dist. to A	dist. to B	dist. to C	Cluster	New Cluster
A1	2	10	0.00	~~4.12~~ 5.65	6.5	A	A
A2	2	5	5	4.12	1.58	C	C
A3	8	4	8.48	2.82	6.51	B	B
B1	5	8	3.60	2.23	5.70	B	B
B2	7	5	7.07	1.41	5.70	B	B
B3	6	4	7.21	2.0	4.52	B	B
C1	1	2	8.06	6.40	1.58	C	C
C2	4	9	2.23	3.60	6.04	B	A

Recompute centroid:

A = ({(2, 10), (4, 9)}) => (3, 9.5)

B = ({(8, 4), (5, 8), (7, 5), (6, 4)})
=> (6.5, 5.25)

C = ({(2, 5), (1, 2)})
=> (1.5, 3.5)

Again,

Date:

Point	x	y	dist. to A	dist. to B	dist. to C	Cluster	New cluster
A1	2	10	1.11	6.54	6.52	A	A
A2	2	5	4.60	4.50	1.58	C	C
A3	8	4	7.43	1.95	6.51	B	B
B1	5	8	2.50	3.13	5.70	A	A
B2	7	5	6.02	0.56	5.70	B	B
B3	6	4	6.26	1.35	4.52	B	B
C1	1	2	7.76	6.39	1.58	C	C
C2	4	9	1.11	4.51	6.04	A	A

New centroid:

A = ({(2, 10), (5, 8), (4, 9)})
=> (3.66, 9)

B = ({(8, 4), (7, 5), (6, 4)})
=> (7, 4.33)

C = ({(2, 5), (1, 2)})
=> ~~(1, 1)~~ (1.5, 3.5)

Again,

Date:

Point	x	y	dist to A	dist to B	dist. to C	Cluster	New cluster
A1	2	10	1.94	7.56	6.51	A	A
A2	2	5	4.33	5.04	1.58	C	C
A3	8	4	6.62	1.05	6.51	B	B
B1	5	8	1.67	4.18	5.70	A	A
B2	7	5	5.21	0.67	5.70	B	B
B3	6	4	5.52	1.05	4.52	B	B
C1	1	2	7.49	6.44	1.58	C	C
C2	4	9	0.33	5.55	6.04	A	A

Hence, the clusters are equal, so, it is the final solution.

Final centroid:

A = mean ({(2,10), (5,8), (4,9)})

=> (3.66, 9)

B = mean ({(8,4), (7,5), (6,4)})

=> (7, 4.33)

C = mean ({(2,5), (1,2)})

=> (1.5, 3.5)

Date:

3. Machine learning Model

⇒ Machine learning Model is a computational program that learns patterns from data and makes decision or prediction on new unseen data. It is created by training a machine learning algorithm on a dataset and optimizing it to minimize errors.

Key characteristics of ML models are:

i) Find hidden patterns from historical information.

ii) Can forecast values or classify inputs.

iii) Learns from additional data and feedback.

iv) Reduces human effort and increases efficiency.

• Types of ML models:

1. Supervised learning Models:

Supervised learning models learn from labeled data, where each input has a known output. The goal is to map input features to correct target value using a mathematical model.

* Types:

Date:

- **Regression:** Regression models predict continuous numerical values rather than categories. Some of its algorithms are:

a) **Linear Regression:** Fits a linear equation to predict numerical outcomes.

b) **Polynomial Regression:** Extends linear regression by fitting polynomial relationships.

c) **Decision-Tree Regression:** Uses tree structure to predict continuous values.

d) **Random Forest Regression:** Ensemble of decision tree regressors for better prediction.

e) **Support Vector Regression (SVR):** Uses SVM principles for regression tasks.

- **Classification:** Classification models assign input data to predefined categories. Some of the algorithm are:

Date:

a) Logistic Regression: Predicts the probability of categorical outcomes using a logical function.

b) Support Vector Machine (SVM): Finds the optimal hyperplane to separate classes with maximum margin.

c) Decision Tree: Splits data recursively based on features to classify samples efficiently.

d) Random Forest: Combines multiple decision trees to improve accuracy and reduce overfitting

e) Naive Bayes: Uses probability theory assuming feature independence to classify data.

f) K-Nearest Neighbors (KNN): Classifies based on the majority level of ~~ne~~ nearest neighbors.

g) Gradient Boosting, XGBoost, ~~LightGB~~ LightGBM: Ensemble method that sequentially combine weak learners to improve performance.

Date:

2. Unsupervised learning:

They work with unlabeled data, discovering hidden patterns, clusters or structures without predefined outputs.

Types:

- **Clustering:** Groups similar data points into clusters based on feature similarity. Some of its algorithm are:

a) **K-Means:** Divides data into k clusters using centroids.

b) **DBSCAN:** Detects dense clusters and identifies outliers automatically.

c) **Hierarchical Clustering:** Builds a nested tree structure of clusters based on similarity.

- **Dimensionality Reduction:** Reduces high-dimensional data while retaining important information for analysis or visualization. Some of its algorithms are:

a) **PCA (Principal Component Analysis):** It projects data into principal components to reduce dimensions.

Date:

b) LDA (Linear Discriminant Analysis): It maximizes class separability while reducing dimensionality.

- Anomaly Detection: Identifies rare or unusual patterns in datasets that deviate from normal behavior.

a) Isolation Forest: Detects anomalies by isolating data points that require fewer splits in a random tree structure.

b) Local Outlier Factor (LOF): Flags anomalies by comparing the local density of a point with the densities of its neighbors.

- Association: Discovers relationships or co-occurrence patterns between items in large datasets.