MONASH University

MONASH
INFORMATION
TECHNOLOGY

Parallel Classification

How to parallelize the training process of ML models?

GROUP OF EIGHT AUSTRALIA

Parallel Classification: Decision Tree

Data parallelism

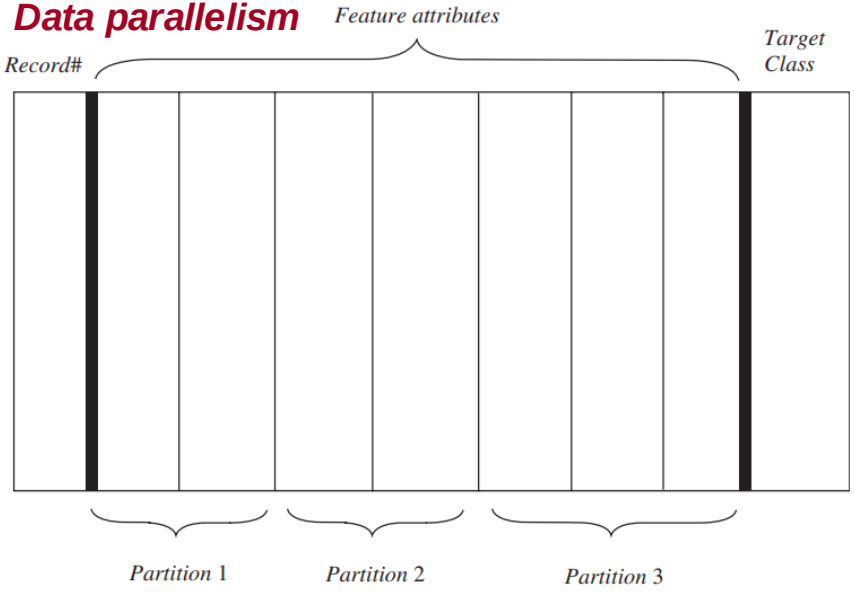


Figure 17.16 Vertical data partitioning of training data set

Rec#	Weather	Temperature	Time	Day	Jog (*Target Class*)
1	Fine	Mild	Sunset	Weekend	Yes
2	Fine	Hot	Sunset	Weekday	Yes
3	Shower	Mild	Midday	Weekday	No
4	Thunderstorm	Cool	Dawn	Weekend	No
5	Shower	Hot	Sunset	Weekday	Yes
6	Fine	Hot	Midday	Weekday	No
7	Fine	Cool	Dawn	Weekend	No
8	Thunderstorm	Cool	Midday	Weekday	No
9	Fine	Cool	Midday	Weekday	Yes
10	Fine	Mild	Midday	Weekday	Yes
11	Shower	Hot	Dawn	Weekend	No
12	Shower	Mild	Dawn	Weekday	No
13	Fine	Cool	Dawn	Weekday	No
14	Thunderstorm	Mild	Sunset	Weekend	No
15	Thunderstorm	Hot	Midday	Weekday	No

Figure 17.11. Training dataset

Parallel Classification: Decision Tree

Data parallelism: ***Vertical Partitioning of Training dataset***

- Feature attributes are partitioned
- Record# & Target are replicated in each partition

Rec#	Weather	Temperature	Jog (*Target Class*)
1	Fine	Mild	Yes
2	Fine	Hot	Yes
3	Shower	Mild	No
4	Thunderstorm	Cool	No
5	Shower	Hot	Yes
6	Fine	Hot	No
7	Fine	Cool	No
8	Thunderstorm	Cool	No
9	Fine	Cool	Yes
10	Fine	Mild	Yes
11	Shower	Hot	No
12	Shower	Mild	No
13	Fine	Cool	No
14	Thunderstorm	Mild	No
15	Thunderstorm	Hot	No

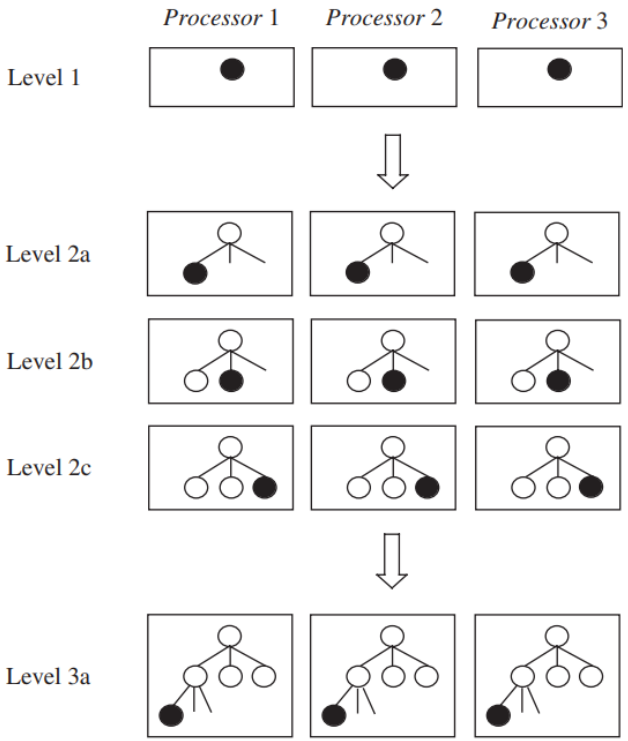
Partition 1

Rec#	Time	Day	Jog (*Target Class*)
1	Sunset	Weekend	Yes
2	Sunset	Weekday	Yes
3	Midday	Weekday	No
4	Dawn	Weekend	No
5	Sunset	Weekday	Yes
6	Midday	Weekday	No
7	Dawn	Weekend	No
8	Midday	Weekday	No
9	Midday	Weekday	Yes
10	Midday	Weekday	Yes
11	Dawn	Weekend	No
12	Dawn	Weekday	No
13	Dawn	Weekday	No
14	Sunset	Weekend	No
15	Midday	Weekday	No

Partition 2

Parallel Classification: Decision Tree

Data parallelism



Black-shaded nodes indicate one currently processed

Each branch is processed one after another (and processed by all P's)

Figure 17.17 Data parallelism of parallel decision tree construction

Rec#	Weather	Temperature	Time	Day	Jog (*Target Class*)
1	Fine	Mild	Sunset	Weekend	Yes
2	Fine	Hot	Sunset	Weekday	Yes
3	Shower	Mild	Midday	Weekday	No
4	Thunderstorm	Cool	Dawn	Weekend	No
5	Shower	Hot	Sunset	Weekday	Yes
6	Fine	Hot	Midday	Weekday	No
7	Fine	Cool	Dawn	Weekend	No
8	Thunderstorm	Cool	Midday	Weekday	No
9	Fine	Cool	Midday	Weekday	Yes
10	Fine	Mild	Midday	Weekday	Yes
11	Shower	Hot	Dawn	Weekend	No
12	Shower	Mild	Dawn	Weekday	No
13	Fine	Cool	Dawn	Weekday	No
14	Thunderstorm	Mild	Sunset	Weekend	No
15	Thunderstorm	Hot	Midday	Weekday	No

Figure 17.11. Training dataset

- All processors will process one node at each level or sub-level at a time
- Parallelism: Each processor will compute IGs for certain attributes (in parallel), which are then shared to determine splitting
- 'Intra tree node parallelism'

Parallel Classification: Decision Tree

***Level 1 (Root Node)*:**

Data parallelism

Processor 1

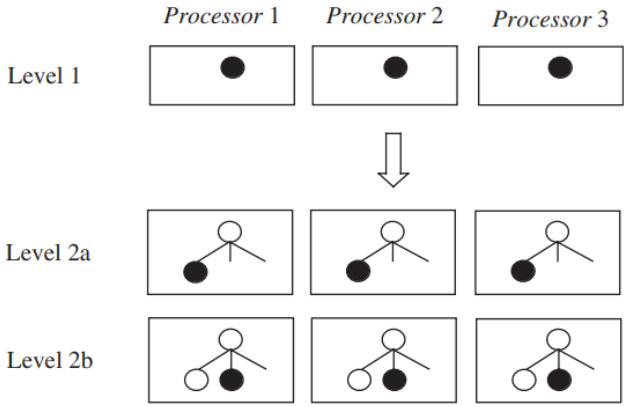
Rec#	Weather	Temperature	Target Class
1			
2			
…			
15			

Processor 2

Rec#	Time	Day	Target Class
1			
2			
…			
15			

Locally calculate the information gain values for: *Weather* and *Temperature*

Locally calculate the information gain values for: *Time* and *Day*



***Global information sharing stage*:**

a. Share target class counts to calculate dataset entropy value
b. Exchange dataset entropy value to determine splitting attribute (e.g. Time attribute is decided to be the splitting attribute)
c. Distribute selected records# to all processor for the next phase (e.g. records 3, 6, 8, 9, 10, 15 for Time *Midday*, and records 1, 2, 5, 14 for Time *Sunset*)

***Decision tree for Level 1*:**

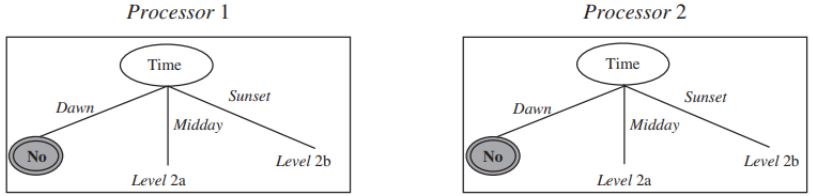


Figure 17.18 Data parallelism in decision tree

$$IG(S,A) = H(S) - H(S,A)$$

Parallel Classification: Decision Tree

Data parallelism

***Level* 2a:**

Processor 1

Rec#	Weather	Temperature	TargetClass
3			
6			
8			
9			
10			
15			

Processor 2

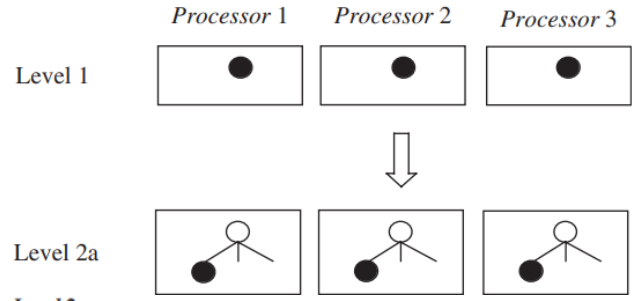
Rec#	Time	Day	TargetClass
3			
6			
8			
9			
10			
15			

Locally calculate the information gain values for: *Weather* and *Temperature*

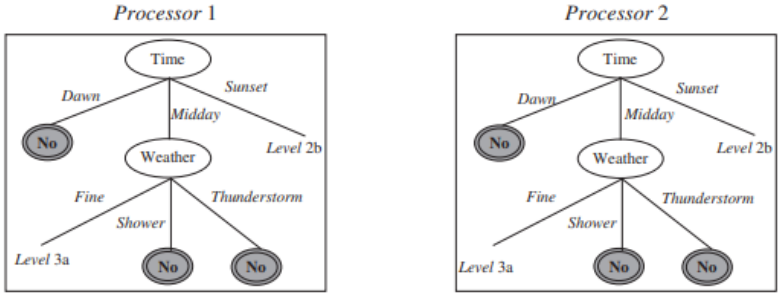
Locally calculate the information gain values for: *Day*

Global information sharing stage:

a. Share target class counts of each partition to calculate dataset entropy value
b. Exchange dataset entropy value to determine splitting attribute (e.g. Weather attribute is decided to be the splitting attribute)
c. Distribute selected records# to all processor for the next phase



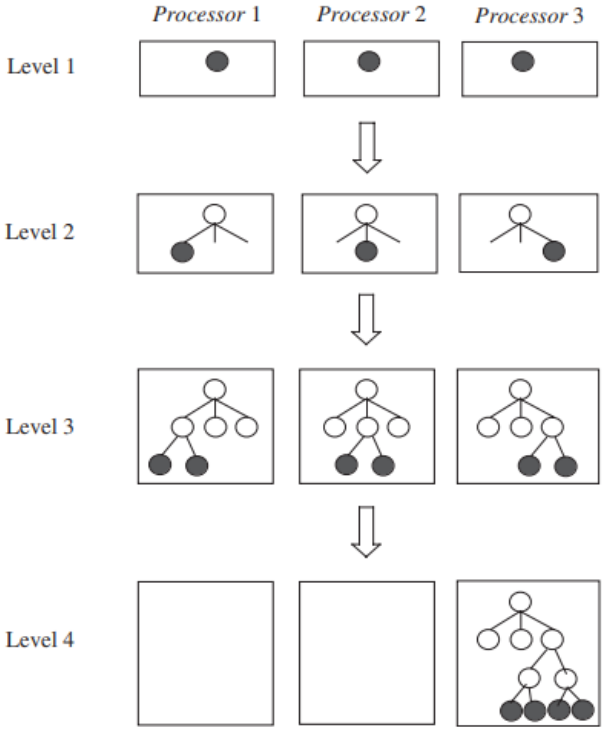
Result decision tree for Level 2:



***Level* 2b: to continue…**

Parallel Classification: Decision Tree

Result parallelism* - *Horizontal Partitioning of dataset



Level 1: Root node processed by all P's

Level 2: Each node processed independently

Level 3: After splitting, each P will process the associated child nodes

Figure 17.20 Result parallelism of parallel decision tree construction

Rec#	Weather	Temperature	Time	Day	Jog (*Target Class*)
1	Fine	Mild	Sunset	Weekend	Yes
2	Fine	Hot	Sunset	Weekday	Yes
3	Shower	Mild	Midday	Weekday	No
4	Thunderstorm	Cool	Dawn	Weekend	No
5	Shower	Hot	Sunset	Weekday	Yes
6	Fine	Hot	Midday	Weekday	No
7	Fine	Cool	Dawn	Weekend	No
8	Thunderstorm	Cool	Midday	Weekday	No
9	Fine	Cool	Midday	Weekday	Yes
10	Fine	Mild	Midday	Weekday	Yes
11	Shower	Hot	Dawn	Weekend	No
12	Shower	Mild	Dawn	Weekday	No
13	Fine	Cool	Dawn	Weekday	No
14	Thunderstorm	Mild	Sunset	Weekend	No
15	Thunderstorm	Hot	Midday	Weekday	No

Figure 17.11. Training dataset

- Multiple nodes are processed concurrently using several processors at each level
- Main rule: A processor that processes a child node will also process its parent nodes
- 'Inter tree node parallelism' (different nodes are processed in parallel)

Parallel Classification: Decision Tree

$$IG(S,A) = H(S) - H(S,A)$$

Result parallelism

***Horizontal Data Partitioning*:**

Processor 1

Rec#	Weather	Temp	Time	Day	Target Class
1					
2					
…					
8					

Processor 2

Rec#	Weather	Temp	Time	Day	Target Class
9					
10					
…					
15					

***Level* 1 (Root Node):**

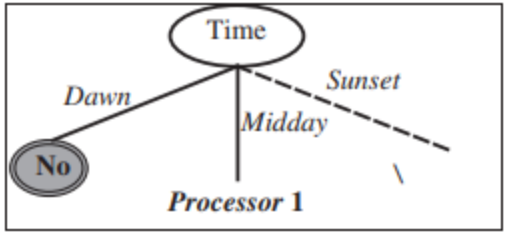
a. Count target class on each partition

b. Perform intra-nod eparallelism the same as for data parallelism to share target class counts to calculate dataset entropy value, exchange dataset entropy value todetermine splitting attribute, and distribute selected records# to all other processors for the next phase)

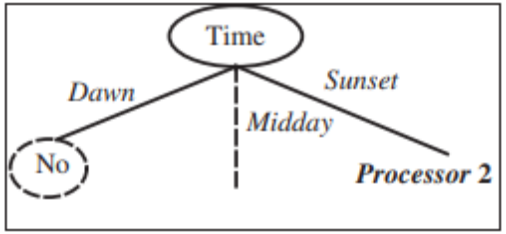
- Recall: To compute entropy & IG, count information from the entire dataset is needed
- Sharing of count information between P's

***Decision tree for Level* 1:**

Processor 1



Processor 2



Parallel Classification: Decision Tree

Result parallelism

Jog	
Yes	No
5	10

Used to compute $H(S)$

		Jog		
		Yes	No	
Time	Dawn	0	5	5
	Midday	2	4	6
	Sunset	3	1	4
				15

		Jog		
		Yes	No	
Day	Weekend	4	6	10
	Weekday	1	4	5
				15

		Jog		
		Yes	No	
Weather	Fine	4	3	7
	Shower	1	3	4
	Thunderstorm	0	4	4
				15

		Jog		
		Yes	No	
Temperature	Hot	2	3	5
	Mild	3	2	5
	Cool	1	4	5
				15

Used to compute $H(S, A)$ *then* $IG(S, A)$

Parallel Classification: Decision Tree

Result parallelism

Level 2:

Processor 1

Rec#	Weather	Temp	Time	Day	Target Class
3					
6					
8					
9					
10					
15					

Processor 2

Rec#	Weather	Temp	Time	Day	Target Class
1					
2					
5					
14					

Global information sharing stage:

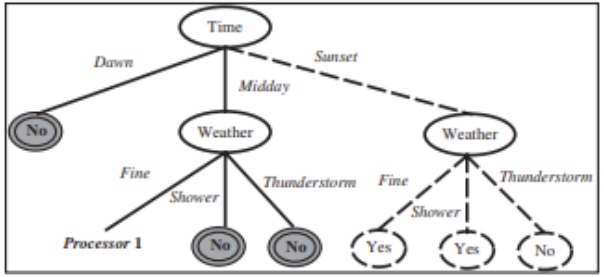
a. Count target class on each partition

b. Perform intra-node parallelism the same as for data parallelism to share target class counts to calculate dataset entropy value, exchange dataset entropy value to determin esplitting attribute,and distribute selected records# to allother processors for the next phase)

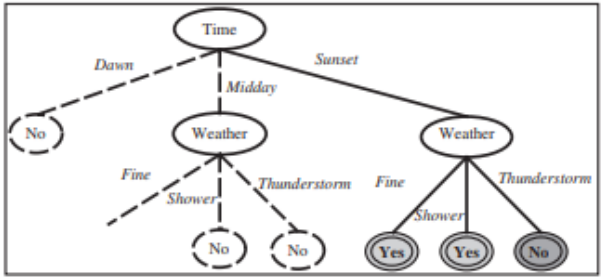
All the necessary information/counts are available in each partition to compute IG to determine splitting

Result decision tree for Level 2:

Processor 1



Processor 2



Parallel Classification: Decision Tree

Level 3:

Processor 1

Rec#	Weather	Temp	Time	Day	Target Class
6					
9					
10					

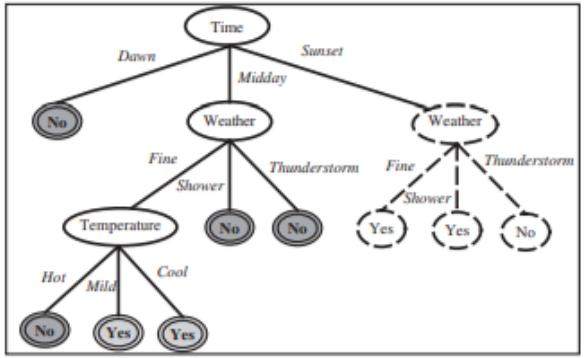
Processor 2

Rec#	Weather	Temp	Time	Day	Target Class

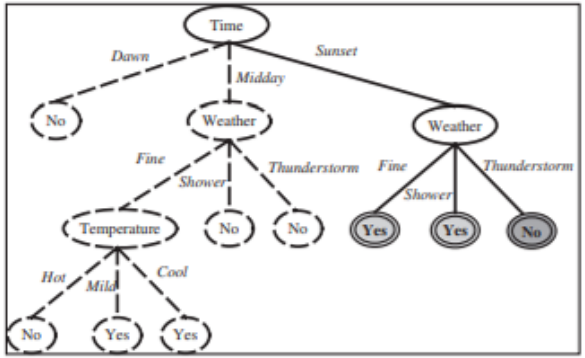
Global information sharing stage:… as like in Level 2 …

Result decision tree for Level 3:

Processor 1



Processor 2



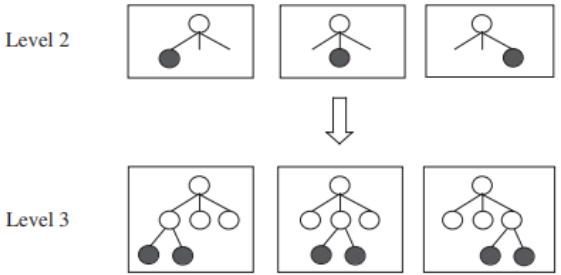


Figure 17.20 Result parallelism of parallel decision tree construction