SCALING UP CLASSIFICATION RULE INDUCTION THROUGH PARALLEL PROCESSING

HAO LI

OUTLINE

- Introduction
- Parallel data mining
- Scaling up data mining: data reduction (feature selection and sampling)
- Scaling up data mining: distributed data mining for parallel processing
- Parallel formulations of classification rule induction algorithms
- PMCRI and J-PMCRI: an approach to efficiently parallelizing parallel formulations of modular classification rule induction algorithms
- Summary

INTRODUCTION

- The age of big data
  - System of earth-orbiting satellites and other space borne probes sends back 1 terabyte of data a day to receiving stations.
  - Twitter generate approximately 12 TB of data per day.
- Therefore, the commercial and the scientific world are confronted with increasingly large and complex data sets to be mined. Tools and techniques that are scalable are required.

PARALLEL DATA MINING

- What is parallel data mining
  - In general, the workload is distributed to several computing nodes by assigning different portions of the data to different processors.
  - The problem of mining geographically distributed data sources is known as Distributed data mining, so parallel data mining is sometimes also referred to distributed data mining due to the fact that the data is distributed to several computing nodes. The difference is parallel data mining concerns with achieving a shorter execution time due to parallelization.
  - Parallel Data mining is utilizing multiprocessor architecture.

PARALLEL DATA MINING

- Two type of multiprocessor architectures.
  - A tightly coupled architecture is constructed of multiple processors sharing one common memory.
  - A loosely coupled architecture is a collection of multiple computers in different locations. Each computer and thus each processor has its local private memory.

PARALLEL DATA MINING

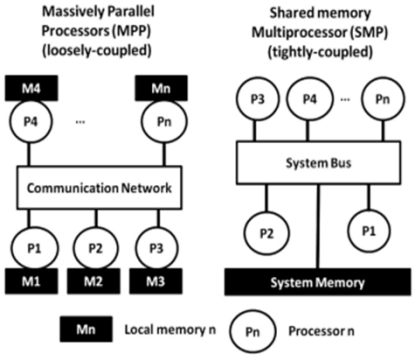


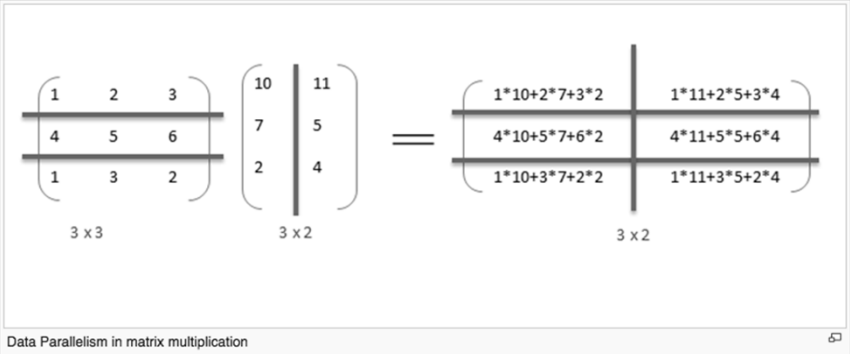
Figure 1 Tightly and loosely coupled multiprocessor computer architectures

Data are communicated using a memory bus system in the tightly coupled architecture, and using a communication network in a loosely coupled architecture.

PARALLEL DATA MINING

- Two principle forms of parallelism.
  - Task parallelism and Data parallelism.
    - **Task parallelism** is the simultaneous execution on multiple cores of many different functions across the same or different datasets.
    - **Data parallelism** (aka SIMD) is the simultaneous execution on multiple cores of the same function across the elements of a dataset.

PARALLEL DATA MINING



Data Parallelism in matrix multiplication

PARALLEL DATA MINING

```
program:
...
if CPU="a" then
   do task "A"
else if CPU="b" then
   do task "B"
end if
...
end program
```

SCALING UP DATA MINING: DISTRIBUTED DATA MINING FOR PARALLEL PROCESSING

- All distributed data mining models can be divided into three basic steps: a sample selection procedure, learning local concepts and combining local concepts using a combining procedure into a final concept description.

SCALING UP DATA MINING: DISTRIBUTED DATA MINING FOR PARALLEL PROCESSING

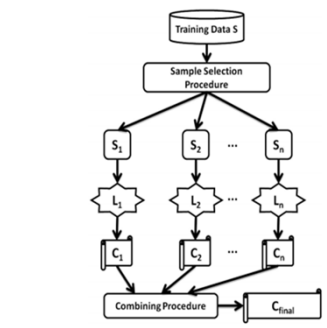


Figure 3 Independent multi-sample mining

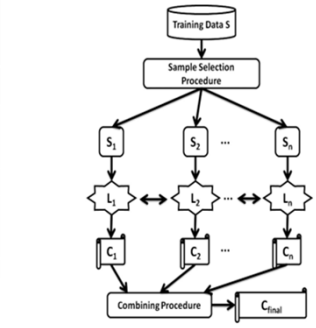


Figure 4 Cooperating data mining model

Left figure: Independent multi-sample mining model. (DDM)

Right figure Cooperating data mining model.

PARALLEL FORMULATIONS OF CLASSIFICATION RULE INDUCTION ALGORITHMS

- The majority of classification rule induction algorithms can be categorized into "divide and conquer" and "covering" also known as 'separate and conquer' methods. The main differences between these methods are that they use different types of knowledge representations; 'divide and conquer' represents its classification rules in the form of decision trees and 'separate and conquer' in the form of rule sets.

PARALLEL FORMULATIONS OF CLASSIFICATION RULE INDUCTION ALGORITHMS

- Review of decision tree
  - IF all instances in the training set belong to the same class THEN return value of this class.
  - ELSE
    - Select attribute A based on information gain or Gini index to split on.
    - Divide instances in the training set into subsets, one for each value of A.
    - Return a tree with a branch for each non empty subset, each branch having a decedent subtree or a class value produced by applying the algorithm recursively.

PARALLEL FORMULATIONS OF CLASSIFICATION RULE INDUCTION ALGORITHMS

- Two approaches to parallelizing TDIDT(Top Down Induction Decision Tree)
  - Synchronous tree construction
    - The basic approach of 'Synchronous tree construction' is a data parallel approach, which constructs a decision tree synchronously by communicating class distribution information between the processors. (SPRINT Shafer et al., 1996 IBM)
    - SPRINT stands for Scalable Parallelizable Induction of Decision Tree
  - Portioned tree construction
    - In the 'Partitioned Tree Construction' approach, whenever feasible each processor works on different subtrees of the decision tree. (No specific algorithm introduced)

PARALLEL FORMULATIONS OF CLASSIFICATION RULE INDUCTION ALGORITHMS

rid	Age	Car Type	Risk
0	23	family	High
1	17	sports	High
2	43	sports	High
3	68	family	Low
4	32	truck	Low
5	20	family	High

(a) Training Set

Age	Class	rid
17	High	1
20	High	5
23	High	0
32	Low	4
43	High	2
68	Low	3

Car Type	Class	rid
family	High	0
sports	High	1
sports	High	2
family	Low	3
truck	Low	4
family	High	5

Figure 3: Example of attribute lists

Attribute List

Age	Class	tid
17	High	1
20	High	5
23	High	0
32	Low	4
43	High	2
68	Low	3

Position of cursor in scan: ← position 0; ← position 3; ← position 6

State of Class Histograms

		H	L
cursor position 0:	C_{below}	0	0
	C_{above}	4	2
cursor position 3:	C_{below}	3	0
	C_{above}	1	2
cursor position 6:	C_{below}	4	2
	C_{above}	0	0

Figure 5: Evaluating continuous split points

$$i(S) = \sum_{j \neq i} P(C_j)P(C_i) = 1 - \sum_{j=1}^{n} (P(C_j))^2$$

Gini index for the original set:
1 – (4/6)^2 – (2/6)^2 = 4/9

Gini index after split on position three:
½ * 0 – ½*4/9 = 2/9

Split on position 3

PARALLEL FORMULATIONS OF CLASSIFICATION RULE INDUCTION ALGORITHMS

Attribute List

Car Type	Class	tid
family	High	0
sports	High	1
sports	High	2
family	Low	3
truck	Low	4
family	High	5

⇒

Count Matrix

	H	L
family	2	1
sports	2	0
truck	0	1

Gini index before the split
1 – (4/6)^2 – (2/6)^2 = 4/9

Gini index after split on family
(3/6)*(1-(2/3)^2-(1/3)^2) + (3/6)*(1-(2/3)^2-(1/3)^2) = 4/9

Gini index after split on sport
(1/3)*0 + (2/3)*(1-(1/2)^2 – (1/2)^2) = 1/3

Gini index after split on truck = 1/6*0 + (5/6)(1-(4/5)^2-(1/5)^2) = 4/15

PARALLEL FORMULATIONS OF CLASSIFICATION RULE INDUCTION ALGORITHMS

- SPRINT achieves uniform data placement and workload balancing by distributing the attribute lists evenly over N processors of a shared-nothing machine.

Processor 0

Age	Class	rid
17	High	1
20	High	5
23	High	0

Car Type	Class	rid
family	High	0
sports	High	1
sports	High	2

Processor 1

Age	Class	rid
32	Low	4
43	High	2
68	Low	3

Car Type	Class	rid
family	Low	3
truck	Low	4
family	High	5

Figure 8: Parallel Data Placement

PARALLEL FORMULATIONS OF CLASSIFICATION RULE INDUCTION ALGORITHMS

Age	Class	Rid
17	High	1
20	High	5
23	High	0

Count class →

High	Low
3	0

Age	Class	Rid
32	Low	4
43	High	2
68	Low	3

Count class →

High	Low
1	2

combine →

High	Low
4	2

PARALLEL FORMULATIONS OF CLASSIFICATION RULE INDUCTION ALGORITHMS

Age	Class	Rid
17	High	1
20	High	5
23	High	0

High	Low
4	2

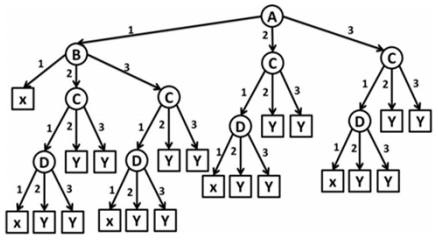
High	Low
3	0

Age	Class	Rid
32	Low	4
43	High	2
68	Low	3

High	Low
1	2

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

- Definition of Modular rule: Modular rules are rules that do not generally fit together naturally in a decision tree.



IF A = 1 AND B = 1 THEN class = x

IF C = 1 AND D = 1 THEN class = x

IF A = 1 AND B = 1 THEN Class = x
IF A = 1 AND B = 2 AND C = 1 AND D = 1 THEN Class = x
IF A = 1 AND B = 3 AND C = 1 AND D = 1 THEN Class = x
IF A = 2 AND C = 1 AND D = 1 THEN Class = x
IF A = 3 AND C = 1 AND D = 1 THEN Class = x

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

- Given data set

Table 1 Opticians' decision table for fitting contact lenses

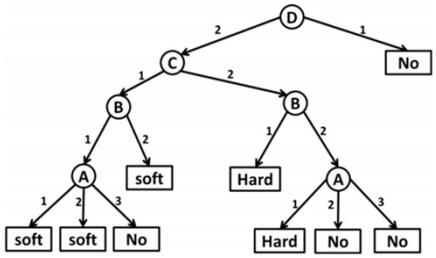
Record id	A	B	C	D	Lenses?	Record id	A	B	C	D	Lenses?
1	1	1	1	1	No	13	2	2	1	1	No
2	1	1	1	2	Soft	14	2	2	1	2	Soft
3	1	1	2	1	No	15	2	2	2	1	No
4	1	1	2	2	Hard	16	2	2	2	2	No
5	1	2	1	1	No	17	3	1	1	1	No
6	1	2	1	2	Soft	18	3	1	1	2	No
7	1	2	2	1	No	19	3	1	2	1	No
8	1	2	2	2	Hard	20	3	1	2	2	Hard
9	2	1	1	1	No	21	3	2	1	1	No
10	2	1	1	2	Soft	22	3	2	1	2	Soft
11	2	1	2	1	No	23	3	2	2	1	No
12	2	1	2	2	Hard	24	3	2	2	2	No

IF A = 3 AND B = 2 AND C = 2 THEN recommendation = No

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

Table 1 Opticians' decision table for fitting contact lenses

Record id	A	B	C	D	Lenses?	Record id	A	B	C	D	Lenses?
1	1	1	1	1	No	13	2	2	1	1	No
2	1	1	1	2	Soft	14	2	2	1	2	Soft
3	1	1	2	1	No	15	2	2	2	1	No
4	1	1	2	2	Hard	16	2	2	2	2	No
5	1	2	1	1	No	17	3	1	1	1	No
6	1	2	1	2	Soft	18	3	1	1	2	No
7	1	2	2	1	No	19	3	1	2	1	No
8	1	2	2	2	Hard	20	3	1	2	2	Hard
9	2	1	1	1	No	21	3	2	1	1	No
10	2	1	1	2	Soft	22	3	2	1	2	Soft
11	2	1	2	1	No	23	3	2	2	1	No
12	2	1	2	2	Hard	24	3	2	2	2	No



PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

Table 1 Opticians' decision table for fitting contact lenses

Record id	A	B	C	D	Lenses?	Record id	A	B	C	D	Lenses?
1	1	1	1	1	No	13	2	2	1	1	No
2	1	1	1	2	Soft	14	2	2	1	2	Soft
3	1	1	2	1	No	15	2	2	2	1	No
4	1	1	2	2	Hard	16	2	2	2	2	No
5	1	2	1	1	No	17	3	1	1	1	No
6	1	2	1	2	Soft	18	3	1	1	2	No
7	1	2	2	1	No	19	3	1	2	1	No
8	1	2	2	2	Hard	20	3	1	2	2	Hard
9	2	1	1	1	No	21	3	2	1	1	No
10	2	1	1	2	Soft	22	3	2	1	2	Soft
11	2	1	2	1	No	23	3	2	2	1	No
12	2	1	2	2	Hard	24	3	2	2	2	No

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{m} \frac{|S_i|}{|S|} Entropy(S_i)$$

The initial entropy is 1.3261 bits
The resulting average entropy after performing the split on D is reduced to 0.7775.
The entropy of the subset of S that covers all instances for which D = 1 is 0; however, the subset that covers all instances for which D = 2 is 1.555 and so even higher than the initial entropy of S

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

- The Prism algorithm does that by maximizing 'the actual amount of information contributed by knowing the value of the attribute to the determination of a specific classification' (Cendrowska, 1987).
- Unlike decision trees, Cendrowska's algorithm looks at one TC at a time and specializes one rule at a time for this TC. In contrast, decision trees specialize several rules simultaneously when splitting on an intermediate node.

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

- Cendrowska views Table 1 as a discrete decision system where attribute values including the classification are seen as discrete messages. She then gives a formula for the amount of information about an event in a message:

$$I(i) = \log_2\left(\frac{probability\ of\ event\ after\ receiving\ the\ message}{probability\ of\ event\ before\ receiving\ the\ message}\right)\ bits$$

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

- Let us say that classification lenses = No is the classification of current interest (TC). Then the message lenses = No provides the following amount of information in the initial decision system about lenses = No:

$$I(lenses = No) = \log_2\left(\frac{1}{p(lenses = No)}\right) = -\log_2\left(\frac{15}{24}\right) = 0.678\ bits$$

- The probability that lenses = No before receiving the message is 15/24 and the probability that lenses = No when given the information that lenses = No is 1. In other words, the maximum amount of information that can be achieved inducing a rule term on the current decision system for the concept lenses = No is 0.678 bits.

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

- A concrete rule term (D = 1) is considered for further specialization of a rule for predicting lenses = No. Then the probability of the event lenses = No before the message (D = 1) is p(lenses = No) =15/24 = 0.625 and the probability of event lenses = No after the message (D = 1) is the conditional probability p(lenses=No|D=1)=12/12=1. So the information about this message is

$$I(lenses = No|D = 1) = \log_2\left(\frac{p(lenses = No|D = 1)}{p(lenses = No)}\right)$$
$$= \log_2\left(\frac{1}{0.625}\right) = 0.678\ bits$$

- The information provided about lenses = No by knowing D = 1 is already the maximum amount of information we can achieve by inducing a rule term about the TC, which in both cases is 0.678 bits. So a further specialization by adding more rule terms to the rule IF(D = 1) THEN lenses = No would not increase the information about class lenses = No, also D = 1 covers only instances of the TC.

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

- The basic idea of PMCRI is to build attribute lists similar to those in SPRINT of the structure ,<attribute value, tuple id, class index>. and then distribute them evenly over p processors. The learning algorithms then search for candidate rule terms and build the classifier in parallel.

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

The Figure shows the building of attribute lists from a training data set. A rule term for class B has been found (Salary >60), which covers in the 'Salary' attribute list instances 5, 0, 2, 4. Thus, the remaining list instances matching ids 1 and 3 need to be deleted in order to induce the next rule term.

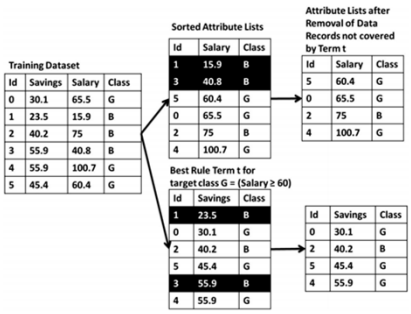


Figure 15 The left-hand side shows how sorted attribute lists are built and the right-hand side shows how list records, in this case records with the ids 1 and 3, are removed in Prism, after a rule term has been found

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

- Step 1: Each processor induces rule terms 'locally' on attribute lists it holds in memory by calculating all the conditional probabilities for the target class in all the attribute lists and taking the largest one. If there is a tie break, the rule term that covers the highest count of the target class is selected as the locally best rule term. The processors now need to communicate in order to find out which processor induced the globally best rule term using the probabilities and target class counts.
- Step 2: After finding the best rule term, each processor needs to delete all attribute list records that are not covered by the globally best rule term. In order to do so all processors need to retrieve the ids from the processor that induced the globally best rule term.
- Step 3: A rule term has been induced and Prism now needs to check whether the rule is completed or not. Each processor checks this independently. If the local attribute lists only contain the target class, then the rule is finished and the processor continues with step 4, otherwise the rule is incomplete and the processor jumps back to step 1.

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

- Step 4: The current rule is completed and Prism needs to restore the data and delete all instances that are covered by the rules induced so far. For each processor the ids of the list records left after the final rule term has been induced are the ones that are covered by the completed rule. These ids are accumulated and remembered by the processor. The original attribute lists are restored and list records matching the accumulated ids are deleted.
- Step 5: Each processor checks if there is still more than one list record left in each attribute list, and each of the attribute lists comprises records associated with more than just one class, then the next rule is induced, otherwise the stopping criterion of Prism is fulfilled and the processor will stop and exit.

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

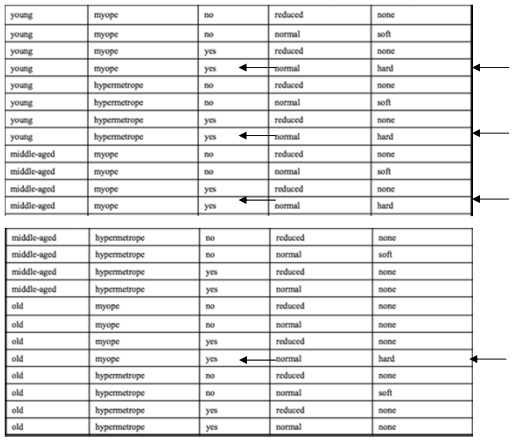
Distribute tables to two processors.

Processor 1 antigmatism = yes for the first term

Processor 2 any attribute value pair

No matter which attribute processor 2 select

The antigmatism = yes will be selected as a term of the global rule.



young	myope	no	reduced	none
young	myope	no	normal	soft
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	no	reduced	none
young	hypermetrope	no	normal	soft
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
middle-aged	myope	no	reduced	none
middle-aged	myope	no	normal	soft
middle-aged	myope	yes	reduced	none
middle-aged	myope	yes	normal	hard

middle-aged	hypermetrope	no	reduced	none
middle-aged	hypermetrope	no	normal	soft
middle-aged	hypermetrope	yes	reduced	none
middle-aged	hypermetrope	yes	normal	none
old	myope	no	reduced	none
old	myope	no	normal	none
old	myope	yes	reduced	none
old	myope	yes	normal	hard
old	hypermetrope	no	reduced	none
old	hypermetrope	no	normal	soft
old	hypermetrope	yes	reduced	none
old	hypermetrope	yes	normal	none

PMCRI AND J-PMCRI: AN APPROACH TO EFFICIENTLY PARALLELIZING PARALLEL FORMULATIONS OF MODULAR CLASSIFICATION RULE INDUCTION ALGORITHMS

age	Spectacle prescription	astigm atism	Tear production rate	Recommen ded lenses
young	myope	yes	reduced	none
young	myope	yes	normal	hard
young	hypermetrope	yes	reduced	none
young	hypermetrope	yes	normal	hard
middle-aged	myope	yes	reduced	none
middle-aged	myope	yes	normal	hard
middle-aged	hypermetrope	yes	reduced	none
middle-aged	hypermetrope	yes	normal	none

old	myope	yes	reduced	none
old	myope	yes	normal	hard
old	hypermetrope	yes	reduced	none
old	hypermetrope	yes	normal	none

The current rule is completed and Prism needs to restore the data and delete all instances that are covered by the rules induced so far. For each processor the ids of the list records left after the final rule term has been induced are the ones that are covered by the completed rule. These ids are accumulated and remembered by the processor. The original attribute lists are restored and list records matching the accumulated ids are deleted.

KEY POINTS OF THE PAPER

- Goal of the paper: this paper mainly surveys advances in parallelization in the field of classification rule induction.
- Related work: this paper is not about presenting new work, it is about related work.
  - This paper reviewed two approaches to parallelizing Top Down Induction of Decision Trees (TDIDT; Quinlan, 1986)
  - Also Parallel Modular Classification Rule Inducer methodology to parallelize a family of algorithms that follow the 'separate and conquer' approach to classification rule inducdtion.
- Advantage and disadvantage of the reviewed algotihms