Enhanced SPRINT Algorithm for Faculty Workloads Application

Benchie L. Maribao, Bobby D. Gerardo and Ruji P. Medina

***Abstract*—**The Enhanced SPRINT (E-SPRINT) algorithm was developed to address some vulnerabilities of SPRINT regarding of attribute classification problems. The enhancement of E-SPRINT is in its growth phase; wherein new technique was integrated. The SLIQ pre-sorting algorithm was used as a basis to address the rewrites and resorts of attributes during classification process which will shorten the time taken for attributes classification. Time for attributes classification is one of the important metric for evaluating classifiers performance. The E-SPRINT was applied and tested for workloads distribution among faculty members. The performance result shows that the E-SPRINT algorithm addressed the weakness of SPRINT Algorithm regarding the rewrites and resorts of an attribute during classification process and provided new techniques. It also shows that it can be used for planning and decision-making by the management to balance the distribution of workloads among faculty members.

***Index Terms*—** Classification Algorithm, Data Mining, Decision Tree, E-SPRINT

I. Introduction

A classification is a form of data analysis that can be used to describe and extract models from important classes and predict future data that provides good decision support in various industries [1]. The management is using classification for forecasting and decision-making for publications like churn prediction, fraud detection, artificial intelligence, medical aspects and academic institutions [2]. There are many classification algorithms available such as Ruled-based, Neural Networks, Support Vector Machines, and Naïve Bayes, but the most commonly used is Decision Tree [3]. A Decision Tree is a practical method for inductive inference over supervised data [2]. Compared with other classification algorithms Decision Tree classifiers obtain better accuracy [4].

SPRINT is a Decision Tree classifier that is fast, and scalable. In the growth phase, it partitioned the datasets recursively using breadth-first greedy technique until each partition belongs to the same leaf node. It uses two data structure the attribute list and histogram [5] [6]. Its weakness is in the process of classifying attribute lists wherein, it iteratively rewrites and resorts attribute lists in each node. This process takes time for the SPRINT to classify attribute lists [7]. This problem serves as a basis to enhance SPRINT algorithm. The enhancement of SPRINT algorithm was based on SLIQ pre-sorting technique to develop a new algorithm. The E-SPRINT algorithm addressed the rewrite and resort of attribute list upon classification process, and it surpassed the classification performance of the SPRINT algorithm in terms of time [8].

This paper presented the application of E-SPRINT Algorithm for Faculty Workloads through simulation. It uses the database of the faculty workloads of Isabela State University. The application of the E-SPRINT Algorithm provides a new technique in the interpretation of data for the distribution of workloads among faculty members. The output can be used by the management for planning and decision- making.

Benchie Maribao Author is a student of Doctor in Information Technology at Technological Institute of the Philippines, Quezon City, Philippines

Bobby Gerardo Author is currently the Vice President for Administration and Finance and holds a rank as professor VI of West Visayas State University, Iloilo City, Philippines

Ruji Medina Author is Dean of the Graduate Programs and concurrent Chair of the Environmental and Sanitary Engineering Program of the Technological Institute of the Philippines, Quezon City, Philippines.

II. Related Literatures

This section discusses the application and importance of data mining in educational sectors and explains the concept of E-SPRINT Algorithm.

A. *Application of data mining in educational sectors*

They are using data mining techniques in the educational sector. These have been used to solve educational problems and perform essential data analysis. The result of data analysis was used to enhance educational standards and management [9].
There are increasing research interests using data mining in education, which is known as Educational Management which deals with methods development that discovers knowledge from databases. The data can be personal or academic which can be used to understand students' behavior, to assist instructors, to improve teaching performance, to evaluate and improve e-learning systems, to improve curriculums and others that can be used by management for planning and decision-making [10]. The application of data mining techniques in the higher educational institution has been examined through the extraction of useful data and provided an analytical tool to serve this information for decision-making [11].

Another study was conducted by [12] on the use of data mining technology to evaluate student’s academic achievement on the different aspect of enrolment. A similar study was completed to examine the educational background of the student’s performance in computer science program at Nigerian University by using the C4.5 algorithm. The result shown that the grade obtained from senior secondary school examination in mathematics is the highest determinant of students’ performance [13]. [14] was conducted a study to investigate the factors associated with the assessment of instructors teaching performance. They used two data mining techniques: regression analysis and decision trees. The stepwise regression method is used for regression analysis and CHAID and CART algorithms for decision trees. As a result of their study, they found that instructors, who have well-prepared course outlines, use satisfactory materials, help the student outside the lectures, grade exams fairly and on time receive higher evaluations.

Reference [15], [16] usedC4.5, ID3 and CART decision tree algorithms to predict engineering student's performance in their final examination. All personal, social, psychological and other environmental variables are the basis for the prediction models for the achievement of the students. The C4.5 has the highest accuracy as compared to other methods such as ID3 and CART algorithms. A study [17] analyzed some numbers of parameters for the derivation of performance prediction indicators needed for teachers performance assessment, monitoring, and evaluation.

B. E-SPRINT Algorithm

The E-SPRINT algorithm was composed of three (3) functions, the *PreSorting, EvaluateSplit*, and *MakeTree.* First, the *PreSorting ( )* function that covers the pre-sorting technique for attribute classifications. Based on the Training Datasets, it will determine all the attributes and store in an array. It creates an attribute list for each attribute and a class list then training samples will be iteratively be done. Second, the *EvaluateSplit ( )* function that deals for the evaluation of the best split point using *Gini Index*. The Gini Index was used to evaluate the best split point for an attribute. It is defined as $gini(GI) = 1\ \Sigma p^2_j$where p_j is the relative frequency of *class j* in GI. If a split divide GI into two subsets GI_1 and GI_2, the index of the divided data $gini_{split}(GI)$ is given by $gini_{split}(GI) = (n1/n)gini(GI_1) + (n2/n)gini(GI_2)$ and last, the *MakeTree ( )* function that covers the splitting of attribute for each leaf nodes.

The E-SPRINT Algorithm is as follows:

PreSorting ()

1. In *Training Data*, determine all the attributes then store the values in an array.

1.1 Create an *Attribute list* (value, ID, and class) for each attribute of the training data.

1.2 Create a *Class List* for each attribute

1.3 Iterate all the training sample

repeat

EvaluateSplit ()

2. Evaluate the best split-point using *Gini Index*

2.1 Determine the all the Gini Index of the attributes of the Training Data.

2.2 Get the *smallest Gini Index*

2.3 Determine the equivalent row then get the *Midpoint*

MakeTree()

3. Split the attributes based on the *Midpoint Values* *until* (no attribute to be split) *end*

III. PERFORMANCE RESULT

This section discusses the performance of E-SPRINT algorithm through simulation.

A. Application of E-SPRINT Algorithm

The E-SPRINT algorithm was applied and tested for the distribution of workload among faculty members. It uses the database of the faculty workloads of Isabela State University as a point of reference. In Table I, it shows that each record in the database was consists of six (6) attributes.

TABLE I
DESCRIPTION OF ATTRIBUTES FOR DATA

Attribute	Value
Employee Name	Name of Employees
Rank	12 to 30
Gender	Male, Female
Designation (equivalent units)	0, 3, 6, 9, 12, 15
Subjects (equivalent units)	0 to 30
FTE	6 to 42

B. Simulation of E-SPRINT Algorithm

The E-SPRINT algorithm simulation uses ten (10) records that were extracted and randomly selected from the database of Faculty workloads of Isabela State University, to test the classification process of the enhanced algorithm.

B.1 Training Data

The Training Data was composed of six (6) attributes namely: name, rank, and designation, subjects' number of units, FTE, and gender. It shows in fig. 1.

FACULTY WORKLOAD

NO.	NAME	RANK	DESIGNATION	NO. OF UNITS/ SUBJECT	FTE	GENDER
1	Faculty 1	12	0	18	18	F
2	Faculty 2	12	0	21	21	M
3	Faculty 3	23	15	26	41	F
4	Faculty 4	20	0	24	24	M
5	Faculty 5	22	6	21	27	M
6	Faculty 6	12	0	24	24	F
7	Faculty 7	16	3	18	21	F
8	Faculty 8	18	9	20	29	M
9	Faculty 9	15	3	21	24	F
10	Faculty 10	14	0	27	27	M

Fig. 1. Training Data

B.2 Pre-sorting Technique

The *PreSorting ( )* function covers the pre-sorting technique for attribute classifications. The Training Data serve as a basis for the pre-sorting of data. Based on the training data, it determines the *attribute lists* and *class list***.** The attribute lists are *rank* and *FTE*, while the class list is *gender***.** In the *Rank List* and *FTE List*, data was automatically being sorted in ascending order while the class list (*gender*) is normal. It shows in fig. 2, the pre-sorting output.

RANK	Class Index
12	1
12	2
12	6
14	10
15	9
16	7
18	8
20	4
22	5
23	3

Rank List

FTE	Class Index
18	1
21	2
21	7
24	4
24	6
24	9
27	5
27	10
29	8
41	3

FTE List

	Class	Leaf
1	F	N1
2	M	N1
3	F	N1
4	M	N1
5	M	N1
6	F	N1
7	F	N1
8	M	N1
9	F	N1
10	M	N1

Class

Fig. 2 Pre-sorting Output

B.3 Splitting Process

The *EvaluateSplit* **( )** function deals the evaluation of the best split point. The output of the *pre-sorting* technique will be the basis. It is the *root node* and named as *N1*.

To determine the best split point for the first split, initially to consider is the *Rank List*. It was used as a basis for the first split of the attributes. The class index of the *Rank List* was compared and matched to the *Class List* index to determine the position of the attribute value. The *Gini Index* was the splitting criterion used to evaluate the "goodness" of the alternative splits for an attribute. It is defined as $gini(GI) = 1\ \Sigma p^2_j$ where p_j is the relative frequency of *class j* in G. If a split divide GI into two subsets GI_1 and GI_2, the index of the divided data $gini_{split}(GI)$ is given by $gini_{split}(GI) = (n1/n)gini(GI_1) + (n2/n)gini(GI_2)$. Based from the computed value, it will determine the smallest *Gini Index*. The smallest value is *0.417* which was located at *position six (6)* and the computed *splitting point is seventeen (17)***.** The Splitting leaf node to the left side was named as *N2* while *N3* to the right side. It shows in fig. 3, the computational cost of the first split.

1	Position 0	0	0	G1	1-(((0/0)^2)+((0/0)^2))	=0.000
		5	5	G2	1-(((5/10)^2)+((5/10)^2))	=0.500
				G	((0/10)*0)+((10/10)*0.500)	=0.500
2	Position 1	0	1	G1	1-(((0/1)^2)+((1/1)^2))	=0.000
		5	4	G2	1-(((5/9)^2)+((4/9)^2))	=0.494
				G	((1/10)*0.000)+((9/10)*0.494)	=0.444
3	Position 2	1	1	G1	1-(((1/2)^2)+((1/2)^2))	=0.500
		4	4	G2	1-(((4/8)^2)+((4/8)^2))	=0.500
				G	((2/10)*0.500)+((8/10)*0.500)	=0.500
4	Position 3	1	2	G1	1-(((1/3)^2)+((2/3)^2))	=0.444
		4	3	G2	1-(((4/7)^2)+((3/7)^2))	=0.490
				G	((3/10)*0.444)+((7/10)*0.490)	=0.476
5	Position 4	2	2	G1	1-(((2/4)^2)+((2/4)^2))	=0.500
		3	3	G2	1-(((3/6)^2)+((3/6)^2))	=0.500
				G	((4/10)*0.500)+((6/10)*0.500)	=0.500
6	Position 5	2	3	G1	1-(((2/5)^2)+((3/5)^2))	=0.480
		3	2	G2	1-(((3/5)^2)+((2/5)^2))	=0.480
				G	((5/10)*0.480)+((5/10)*0.480)	=0.480
7	Position 6	2	4	G1	1-(((2/6)^2)+((4/6)^2))	=0.444
		3	1	G2	1-(((3/4)^2)+((1/4)^2))	=0.375
				G	((6/10)*0.444)+((4/10)*0.375)	=0.417
8	Position 7	3	4	G1	1-(((3/7)^2)+((4/7)^2))	=0.490
		2	1	G2	1-(((2/3)^2)+((1/3)^2))	=0.444
				G	((7/10)*0.490)+((3/10)*0.444)	=0.476
9	Position 8	4	4	G1	1-(((4/8)^2)+((4/8)^2))	=0.500
		1	1	G2	1-(((1/2)^2)+((1/2)^2))	=0.500
				G	((8/10)*0.500)+((2/10)*0.500)	=0.500
10	Position 9	5	4	G1	1-(((4/9)^2)+((5/9)^2))	=0.494
		0	1	G2	1-(((1/1)^2)+((0/1)^2))	=0.000
				G	((9/10)*0.494)+((1/10)*0.000)	=0.444
11	Position 10	5	5	G1	1-(((5/10)^2)+((5/10)^2))	=0.500
		0	0	G2	1-(((0/0)^2)+((0/0)^2))	=0.000
				G	((10/10)*0.500)+((0/10)*0)	=0.500

Fig. 3(a) First Split Computational Cost

RANK	Class Index
12	1
12	2
12	6
14	10
15	9
16	7
18	8
20	4
22	5
23	3

	Class	Leaf
1	F	N2
2	M	N2
3	F	N2
4	M	N2
5	M	N2
6	F	N2
7	F	N3
8	M	N3
9	F	N3
10	M	N3

Smallest Gini Index: Position 6
= 0.417

Midpoint = (16 + 18) / 2
= 17

Splitting Point = 17

Fig. 3(b) First Split Computational Cost

The basis for the second split is the output of the first split, which is shown in figure 6. To come up with the best split point, it will undergo the same process as finding the best split point in the first split, but it must be the *FTE List* to consider. *N2 (left-leaf node)*, was the basis for the computational cost of finding the best split point for the second split *(left-leaf node)*, which will be named as *N4 (left-leaf node)* and *N5 (right-leaf node)*. The computed smallest gini index is *0.267* which was located at *position five (5)* with the *splitting point value of 25.5*. *N3 (right-leaf node)*, was the basis for the computational cost of finding the best split point for the second split *(right-leaf node)*, which will be named as *N6 (left-leaf node)* and *N7 (right-leaf node)*. The computed smallest gini index is *0.000* which was located at *position three (3)* with the *splitting point value of 35*. The computational cost of the second split, are shows in fig. 4 (left side) and fig. 5 (right side).

Fig. 4 Second (left Node) Split Computational Cost

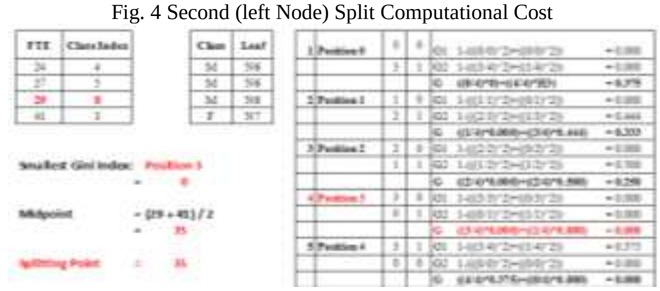

Fig. 5 Second (right Node) Split Computational Cost

B.4 Splitting Process

The *MakeTree ( )* function covers the splitting of an attribute for each leaf nodes. The splitting of an attribute for each leaf nodes was based on the result of the *EvaluateSplit ( )* function. The splitting point of *N1 (root node)* is *seventeen (17)*. It was based on the computational cost which is shown in figure 3. Based on the splitting point of *N1*, where *RANK < =17*, there are six (6) records that belong to *N2 (leaf node)* and four (4) records to the *N3 (leaf node).*The first split output is shown in fig. 6.

RANK <= 17

NAME	RANK	DESIGNATION	NO OF UNITS/ SUBJECT	FTE	Class
Faculty 1	12	[illegible]	18	18	F
Faculty 6	12	[illegible]	24	24	F
Faculty 2	12	[illegible]	21	21	M
Faculty 10	14	[illegible]	27	27	M
Faculty 9	17	3	21	24	F
Faculty 7	16	3	18	21	F

NAME	RANK	DESIGNATION	NO OF UNITS/ SUBJECT	FTE	Class
Faculty 8	18	[illegible]	29	29	M
Faculty 4	20	[illegible]	24	24	M
Faculty 5	22	6	21	27	M
Faculty 3	25	15	26	40	F

Fig. 6 First Split Output

The second split was based on the output of the first split. First to split is *N2*. There are six (6) records that belong to *N2*. It is shown in figure 4 the computational cost, wherein the *splitting point is 25.5*. Those records with an FTE of less than or equal to 25.5 *(FTE <=25.5)* will belong to *N4 (leaf node)* while the rest will be at *N5 (leaf node)*. There are five (5) records on *N4* while one (1) record on *N5*. The splitting point of *N3* is *35***.** It was based on the computational cost which is shown in figure 5. N3 was composing of four (4) records. Based on the splitting point, where *FTE <=35*, there are three (3) records that belong to *N6 (leaf node)* and one (1) record to the *N7 (leaf node).*The second split output is shown in fig. 7.

FTE <= 25.5

NAME	RANK	DESIGNATION	NO UNITS/SUBJECT	FTE	CLASS
Faculty 1	12	[illegible]	18	18	F
Faculty 2	12	[illegible]	21	21	M
Faculty 7	16	3	18	21	F
Faculty 6	12	[illegible]	24	24	F
Faculty 9	15	3	21	24	F

NAME	RANK	DESIGNATION	NO UNITS/SUBJECT	FTE	CLASS
Faculty 10	14	[illegible]	27	27	M

Fig. 7(a) Second (left node) Split Output

FTE <= 35

NAME	RANK	DESIGNATION	NO UNITS/SUBJECT	FTE	CLASS
Faculty 4	20	[illegible]	24	24	M
Faculty 5	22	6	21	27	M
Faculty 8	18	[illegible]	29	29	M

NAME	RANK	DESIGNATION	NO UNITS/SUBJECT	FTE	CLASS
Faculty 3	25	15	26	40	F

Fig. 7(b) Second (right node) Split Output

IV. CONCLUSION

To simulate the application of the E-SPRINT Algorithm, a faculty workload database of Isabela State University was used. The application was made possible using the developed E-SPRINT algorithm. The E-SPRINT algorithm enhancement was made in its growth phase, wherein new technique was integrated. The output of the application shows that the E-SPRINT algorithm addressed the weakness of SPRINT algorithm regarding of the rewrites and resorts of an attribute during the classification process. It also provides new techniques for the classification process.

Furthermore, the output of the application of the E-SPRINT algorithm can be used for planning and decision-making by the management to balance the distribution of workloads among faculty members.

REFERENCES

[1] Han, J. Data Mining Concepts and Techniques [M].2001.Beijing: Machinery Industry Press.

[2] Anyanwu, M., and Shiva, S. Application of Enhanced Decision Tree Algorithm to Churn Analysis.2009 International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09), Orlando Florida.

[3] Tan, P., Steinbach, M. et.al. Introduction to Data Mining

[4] Michie, D., Spiegelhalter, D.J. et.al. (1994). Machine Learning Neural and Statistical Classification. Ellis Horwood.

[5] Shafer, J., Agrawal, R., et.al. SPRINT: A scalable parallel Classifier for Data Mining [C]. Proceedings of 22ndJ International Conference on Very Large Database, Mumbai(Bombay), India

[6] Anyanwu, M., and Shiva, S. Comparative Analysis of Serial Decision Tree Classification Algorithms. International Journal of Computer Science and Security, (IJCSS) Volume(3):Issue(3)

[7] Haider, A.A., and Asghar, S. A Survey of Logic Based Classifiers. International Journal of Future Computer and Communication, Vol. 2, No.2, April 2013

[8] Maribao, B. et al. (2016) Enhanced SPRINT Algorithm for Attribute Classification Problem.2016 International Conference on Electronics, Computer and Information Technology (ECIT-16), ISBN 978-93-84468-66-8. August 2016

[9] Xingquan Zhu, Ian Davidson. “Knowledge Discovery and Data Mining: Challenges and Realities”, ISBN 978159904-252,Hershey,NewYork,2007.

[10] Romero C., Ventura S., Garcia E. (2008) Data mining in course management systems: Moodle case study and tutorial, Computers & Education, Vol. 51, No. 1, pp.368-384, 2008 https://doi.org/10.1016/j.compedu.2007.05.016

[11] Varun Kumar and Anupama Chadha, “An Empirical Study of the Applications of Data Mining Techniques in Higher Education”, International Journal of Advanced Computer Science and Applications, Vol. 2, No.3,March 2011

[12] Chin Chia Hsu and Tao Huang. The use of Data Mining Technology to Evaluate Student’s Academic Achievement via multiple Channels of Enrolment. An empirical analysis of St. John’s University of Technology.

[13] Osofisan A.O. and Olamiti A.O. Academic Background of Students and Performance in Computer Science Programme in a Nigerian University. *European Journal of Social Science. Vol. 33 Issues 4. 2009.*

[14] MardikyanS., and Badur B. Analyzing teaching Performance of Instructors Using Data Mining techniques. Informatics in Education, 2011, Vol. 10, No.2, pp 245 – 257.

[15] Hemaid and El-Halees. Improving Teacher Performance using Data Mining International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 2, February 2015.

[16] Surjeet K.Y and Saurabh P. Data Mining: A Prediction for Performance Improvement of Engineering Students using Classification World of Computer Science and Information Technology Journal (WCSIT)ISSN: 2221-0741 Vol. 2, No., 2012

[17] Pal K. and Pal S. Evaluation of Teacher’s Performance: A Data Mining Approach. International Journal of Computer Science and Mobile Computing .IJCSMC, Vol 2, Issue. 12, Dec., 2013.

Benchie L. Maribao is currently taking up her Doctor in Information Technologyand now on dissertation writing at Technological Institute of the Philippines, Quezon City, Philippines and finished her MIT degree at University of La Salette, Santiago City, Philippines in the year 2009. She obtained the Bachelor of Science in Computer Science degree at the University of East, C.M. Recto Avenue, Manila, Philippines in the year 2000. She started her teaching profession at Northeastern College, Santiago City,Philippines (2001-2004). In 2004, she transferred to the Institute of Information and Communication Technology at Isabela State University, Echague, Isabela, Philippines. Currently, she has been designated as the Department and BSIT Program Chairman, and has been extensively involved in Research and Extension of the Institute. She has been tapped by various agencies to perform job relevant to her field of specialization. She is a member

of the Philippine Society of IT Educators (PSITE). Her field of interest includes information system and data mining.

Dr. Bobby D. Gerardo is currently the Vice President for Administration and Finance and holds a rank as professor VI of West Visayas State University, Iloilo City, Philippines. His dissertation was about Discovering Driving Patterns using Rule-based intelligent Data Mining Agent (RiDAMA) in Distributed Insurance Telematic Systems. He has published more than 65 research papers in national and international journals and conferences. He is a referee to international conferences and journal publications such as IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Knowledge and Data Engineering and Elsevier Journal on Pervasive and Mobile Computing, and Ecological Informatics. His research interests lie in the area of distributed systems, telematics systems, CORBA, data mining, we services, ubiquitous computing and mobile communications.

Dr. Ruji P. Medina is Dean of the Graduate Programs and concurrent Chair of the Environmental and Sanitary Engineering Program of the Technological Institute of the Philippines in Quezon City. He holds a Ph.D. in Environmental Engineering from the University of the Philippines with sandwich program at the University of Houston, Texas. He counts among his expertise environmental modeling and mathematical modeling using multivariate analysis.