Scalable Action Mining Hybrid Method for Enhanced User Emotions in Education and Business Domain

Sanchari Chatterjee [1], Angelina A. Tzacheva [2], and Zbigniew Ras [1]

[1]Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC, 28223, USA
[2] Computer Science and Information Technology
College of Computing and Engineering
WestCliff University
Irvine, CA 92614

Abstract. Education sector, Business field ,Medical domain and Social Media, huge amounts of data in a single day . Mining this data can provide a lot of meaningful insights on how to improve user experience in social media, users engage in these domains collect and cherish the data as they hope to find patterns and trends and the golden nuggets that help them to accomplish their goal. For example: How to improve student learning; how to increase business profitability; how to improve user experience in social media; and how to heal patients and assists hospital administrators. Action Rule Mining mines actionable patterns which are hidden in various datasets. Action Rules provide actionable suggestions on how to change the state of an object from an existing state to a desired state for the benefit of the user. There are two major frameworks in the literature of Action Rule mining namely Rule-Based method where the extraction of Action Rules is dependent on the pre-processing step of classification rule discovery and Object-Based method where it extracts the Action Rules directly from the database without the use of classification rules. Hybrid Action rule mining approach combines both these frameworks and generates complete set of Action Rules. The hybrid approach shows significant improvement in terms computational performance over the Rule-Based and Object-Based approach. In this work we propose a novel Modified Hybrid Action rule method with Partition Threshold Rho, which further improves the computational performance with large datasets.

Keywords: Actionable Patterns, Action Rules, Emotion Detection, Data Mining, Rule-Based, Object-Based.

1 Introduction

Data science emphasises on various techniques to extract some surprising,very interesting,and unknown knowledge patterns from massive data.These techniques embrace the relationship of data objects with other objects (Clustering) or classes (Classification) to unwrap useful patterns in the data. One of the simple data mining method is the rule based learning that identifies, learns,or develops 'rules' to store, operate or apply. Association Rules and Decision are the few fragments of rule-based methods that actually generates rules to associate patterns and classify data respectively. In general, we constitute rules as given in Equation 1, where the *antecedent* is a conjunction of conditions and the *consequent* is the resulting pattern in the given data for the given conditions in antecedent.

$$condition(s) \rightarrow result(s) \tag{1}$$

Action rule is the knowledge extraction technique developed in context to advocate possible transitions for an individual to move from one state to another. For example, recommending the business to improve customer satisfaction [1] and sentiment analysis on Twitter [2]. Action rules follow the representation, similar to Equation 1, as given in Equation 2, where Ψ represents a conjunction of stable features, $(\alpha \rightarrow \beta)$ represents a conjunction of changes in values of flexible features and $(\theta \rightarrow \phi)$ represents desired change in decision action which is beneficial to the user.

$$[(\Psi) \wedge (\alpha \rightarrow \beta)] \rightarrow (\theta \rightarrow \phi) \tag{2}$$

Action Rules recommending Actionable pattern are prone to acquire definite form of cost to the user [3], [4]. Cost for actions in Action Rules include time, energy, money, or human resources. Actions being recommended can cause both positive(*benefits*) and negative(*loses*) effects for users [5]. Thus, Action Rules recommendations system should take on low cost to the users to make them plausible actions. The existing approaches [6–9] do not consider the cost effectiveness for recommendations. In [3] [10], the concept of cost of the Action Rules is introduced and refined. Searching for the low cost Action Rules from huge dataset can really be very time consuming and will require a distributed and scalable approach for extracting them in a practicable timeframe.

Distributed Processing frameworks like Hadoop [11] and Spark [12] have been introduced to make data mining and big data processing faster and easier. These frameworks distribute the data among nodes in a cluster of computers. The data processing work is distributed among the multiple nodes, each of which on their part of the data performs computations . Finally, when all nodes finish executing their own tasks, the results are merged together to present the final result. In this work, we use Apache Spark [12] framework for implementing a scalable solution to the proposed Action Graph method, and make it suitable for big data. Spark provides APIs such as GraphX [13] for a productive parallel processing in large graphs.

In this paper we focus on hierarchically structured recommender system to improve the efficiency of a company's growth engine. The NPS dataset used for this research contains answers to a set of questionnaire sent to a randomly chosen groups of customers. It covers 34 companies called clients. The purpose of the questionnaire is to check customer satisfaction in using services of these companies which have repair shops all involved in a similar type of business (fixing heavy equipment). These shops are located in 29 states in the US and Canada. Some of the companies have their shops located in more than one state. They can compete with each other only if they target the same group of customers. The performance of a company is evaluated using the Net Promoter System (NPS). For that purpose, the data from the completed questionnaires are stored in NPS datasets. They focus on 34 such datasets, one for each company. Knowledge extracted from them, especially action rules and their triggers, can be used to build recommender systems giving hints to companies how to improve their NPS ratings.Author believes larger the datasets ,higher is the knowledge extracted from them. Authors [14] present the concept of semantic similarity between companies. More semantically similar the companies are, the knowledge extracted from their joined NPS datasets has higher accuracy and coverage. The hierarchically structured recommender system is a collection of recommender systems organized as a tree. Lower the nodes in the tree, more specialized the recommender systems are and the same the classifiers and action rules used to build their recommendation engines have higher precision and accuracy.

In this paper, we propose an extension to our previous work on distributed actionable pattern mining with Spark [15]. We extract actions rules from the business and survey datasets, that help to obtain better , desirable outcome for future.

In this work, we focus on Opinion Mining from Text to suggest Actionable Recommendations. The Actionable Patterns may suggest ways to alter the user's sentiment or emotion to a more positive or desirable state. We extract action rules from business data and student survey data. Action Rule Mining literature consists of two major frameworks namely: Rule-Based approach and Object-Based approach. In this work we focus on Hybrid Action Rule mining method, which combines the above two frameworks with the advantage of scalability with large datasets. In this work we propose a new Modified Hybrid Action Rule [16] mining approach that improves the computational performance. We propose a new Threshold Rho - which allows the user to choose the number of data partitions . This yields Faster Scalable processing. We are applying the method to Student Survey Data, however this method can be used for Improving Customer Satisfaction as well. We also aim to suggest ways to improve the Teaching Methods and Student Learning and also how to change detractors(Customers with Negative Emotions) to promoters(Customers with Positive Emotions) in business . We implement and test our system in Scalable Environment with BigData using the Apache Spark platform.

2 Related Work

Multiple techniques are proposed to discover action-rules, and actions rules are very important in modeling expert and domain knowledge, but there are few drawbacks, like the problem of triggering those rules which is left exclusively to domain experts and Knowledge. To trigger a specific rule on objects with meta-actions , they might as well trigger transitions outside of the target action rule scope. Those additional transitions caused by meta-actions are called side effects, which could be positive or negative. Negative side effects could be ruinous in some domains such as healthcare. Authors in this paper [17], try to reduce those negative side effects by extracting personalized action rules. They propose three object-grouping schemes with regards to same negative side effects to extract personalized action rules for each object group. The authors indicate trusting personalization is a very important aspect in filtering noise that skilled experts face when making decisions.

The authors of the paper [18] provides an overview of a user-friendly NPS based Recommender System for driving business revenue. This technique hierarchically designed recommender system for improving NPS of clients that is driven mainly by action rules and meta-actions. The paper presents main techniques used to build the data-driven system, including data mining and machine learning techniques, such as action rules and meta actions, hierarchical clustering, as well as visualization design. The system implements domain-specific sentiment analysis performed on comments collected within telephone surveys with end customers. Advanced natural language processing techniques are used including visualization, dependency analysis, aspect-based sentiment analysis, text summarization and text parsing.

The Authors Kuang and et.al in their paper [19] proposes a new strategy to improve NPS (Net Promoter Score) of certain companies called HAMIS. Those companies are involved in heavy equipment repair in the US and Canada. HAMIS is based on the semantic dendrogram built by using agglomerative clustering strategy and semantic distance between clients. More similar is the knowledge extracted from two clients,more close these clients semantically are to each other. There dataset involves 34 clients located in different areas across the United States as well as some parts of Canada. These clients provide similar services to over 25,000 customers. The dataset consists of three categories of values which are collected from the questionnaire answered by randomly selected customers during 2011 and 2012.Each company is represented by a dataset which is built from answers to the questionnaire sent to a number of randomly chosen customers using services offered by this company. Before knowledge is extracted from these datasets, each one is extended by merging it with datasets which are close to it in the semantic dendrogram, have higher NPS, and if classifiers extracted from them have higher FS-score. The authors explain that by expanding datasets assigned to nodes of the dendrogram, recommender systems can give clients more promising suggestion for improving their NPS score for their business.

The authors of paper [14] present preliminary results of a flexible hierarchically structured recommender system for improving NPS of a company in a global competitive market. Clients are compare in terms of the similarity of their knowledge concerning the meaning of three concepts: promoter, passive, and detractor. The recommendations are based on action rules which are extracted from the datasets assigned to all nodes of the dendrogram. The questionnaire sent to the customers allows them to enter statements in the text format explaining their ratings. Information included in these statements helps us to find triggers for action rules. The triggers are also called meta-actions [20], [21].

Kuang and Ras talk about building a recommender system in their paper [22] which is driven by action rules and meta-actions for providing proper suggestions to improve revenue of a group of clients (companies) involved with similar businesses. They collect feedback from customers and use them as their dataset. The paper proposes a strategy to classify and organize meta-actions in such a way that they can be applied most efficiently to achieve desired goal. In their previous work, [14] they propose and implement the method of mining meta-actions from customers' reviews in text format. However they discover action rules need more than one meta-action to be triggered. The way and the order of executing triggers causes new problems due to the commonness, differential benefit and applicability among sets of meta-actions.

Recently various domains like medicine [23], education [24], and business [25] started adopting data science research in their respective problems. Many research studies have focused on using the copious real world datasets for healthcare applications and decision making using such knowledge extraction and data mining techniques [26]. For example, in context to hospital readmission, researchers and scientists created a machine learning model to predict patient readmissions by considering some basic patient admission characteristics and their billing codes [27]. Some emphasis on predicting the liklihood of patient readmitting to the hospital, modelled as risk prediction, using Support Vector Machines,Neural Networks, and Random Forests [28]. Similarly, there is a study on using logistic regression to measure the relationship between early readmission and diabetes [29], and a study on using a classic data mining technique like Support Vector Machine to predict readmission [30] using other features such as patient demographics,admission type, disease type, and clinical procedures undertaken. There is an interesting study that came into focus in the recent years related to designing a personalized procedure graphs,that gives a probability on patient's future procedure and recommend hospitals in making decisions for a patient [31, 32].

Nowadays a systematic study has been conducted on developing different types of machine learning models, including both deep and non-deep ones, for business analytics, where they aim to build the machine learning models upon include both knowledge-driven ones, and data-driven features. Businesses also want to keep a good reputation and maintain public trust.A new topic was being introduced related to Interpretability. For a relevant example, one of the social media like Facebook uses model for maximizing digital ad revenue, which has inadvertently shown users offensive content or disinformation in recent years. The solution would be for Facebook to look at why their model shows this content so often, then commit to reducing it. Interpretability plays a crucial role here. These concepts add value and practical benefits when businesses apply them. For starters, interpretability can lead to better decision-making because when a model is tested in the real world, those who developed it can observe its strengths and weaknesses.

In this work, we prefer to use rule based systems inorder to recommend various steps to improve User's emotions including Student Surveys and Customer's Net Promoter Score (NPS) for businesses . Rule based systems are one of the most commonly used machine learning methods like regression, classification and association [33] because it is simple to understand and easy to use. Action rules are such rule based systems that designed to recommend actionable insights, for example recommendations for businesses to gain profit by finding interesting actionable patterns in the data [34]. In the literature, action rules are extracted using two different methods. First method is a rule based approach, in which first the intermediate classification rules are extracted using efficient rule generation algorithms such as LERS or ERID. From these extracted rules, action rules are generated with systems like DEAR [6], which extracts Action Rules from two classification rules, or ARAS [7], which extracts Action Rules using a single classification rule. Second method is object-based approaches, in which the Action Rules are extracted directly from the given decision table without involving any intermediary steps. Systems ARED [9] and Association Action Rules [8] works in the object-based approach. Algorithms, except association action rules, runs much faster with the aim of extracting rules that provides maximum benefits to the user and extracts only limited recommendations.

Ras and Tzacheva [3] introduced the concept of cost and feasibility of Action Rules as an interesting measure. They proposed a graph based method for extracting plausible and low cost Action Rules. Ras and Tzacheva [3] proposed a heuristic search of new low cost Action Rules, where objects supporting the new set of rules also supports the existing rule set but the cost of reclassifying them is much lower for the new rules. Later, Tzacheva and Tsay [10] proposed a tree based method for extracting low cost Action Rules.

Some research , apart from Action Rules has been done on extracting Actionable knowledge. For example, Yang, et.al [35] considered *Customer Attrition* in Customer Relationship Management (CRM) in telecommunications industry and the cost complexities involved in gaining profit to all customers. They proposed a method that extract low cost Actionable patterns for converting undesired customers to loyal ones while improving the net profit of all customers. Karim and Rahman [36] proposed another method to extract cost effective actionable patterns for customer attrition problem in post processing steps of Decision Tree and Naive Bayes classifiers. Su, et.al [4] proposed a method to consider positive benefits that occurs by following an Action Rule apart from all costs that incur from the same rule. Cui, et.al [37] proposed to extract optimal actionable plans during post processes of Additive Tree Model (ATM) classifier. These actionable patterns can actually change the given input to a desired one with a minimum cost. Hu, et.al [38] proposed an integrated framework to gather the cost minimal actions sets to provide support for social projects stakeholders inorder to control risks involved in risk analysis and project planning phases. Lately, Hu, et.al [39] developed an ensemble framework and cost sensitive method to predict software project risk predictions and conducted large scale analysis over 60 models 327 real world project samples.

Due to the advent of big data, some research [21], [15], [40] started applying distributed computing frameworks like MapReduce [11] and Spark [12], recently have been done to extract actionable recommendation completely in a clustered setup. Bagavathi [21] pro-

Table 1. Example Decision System $\mathbb{T}$

X	A	B	C	D
x_1	Y	N	N	D_1
x_2	Y	H	Y	D_2
x_3	Y	H	Y	D_1
x_4	N	N	N	D_2
x_5	N	H	N	D_1
x_6	N	N	Y	D_2
x_7	N	H	Y	D_2
x_8	N	H	N	D_1

posed a method to distribute the data in random to multiple sites, combining results from all sites and taking average on parameters like Support and Confidence. Bagavathi [15] handle the load balancing by uniformly distributing the data into partitions based on the decision attribute. Authors [40] introduces a new method of projecting the database into smaller chunks, for handling data with large number of attributes, and extract action rules from them effectively.

In this work we propose a Modified Hybrid Action Rule mining approach with Additional Threshold Rho - for the Number of Partitions which further improves the computational performance from our previous method that has only one threshold [16]. This allows for Faster and more Scalable processing. We will apply our method to the Student Survey Data , and NPS business data however this method can be used for as well. We are focusing on our work to suggest ways to improve the Teaching and Student Learning methods and also how to improve Customer Satisfaction , like the status change from detractors(Customers with Negative Emotions) to promoters(Customers with Positive Emotions) in business . We implement and test our system in Scalable Environment with BigData using the Apache Spark platform.

3 Background

In this section, we give some basic idea about Decision system, Action Rules, Spark and GraphX frameworks to understand out methodology.

3.1 Decision System

Consider a decision system given in Table 1

Information System can be represented as $\mathbb{T} = (\mathbb{X}, \mathbb{A}, \mathbb{V})$ where,

X is a nonempty, finite set of objects: $\mathbb{X} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$

A is a nonempty, finite set of attributes: $\mathbb{A} = A, B, C, D$ and

V_i is the *domain* of attribute a which represents a set of values for attribute $i | i \in \mathbb{A}$. For example, $V_B = N, H$.

An information system becomes Decision system, if $A = A_{St} \cup A_{Fl} \cup d$, where D is a *decision attribute*. The user chooses the attribute d if they wants to extract desired action from $d_i : i \in V_d$. A_{St} is a set of *Stable Attributes* and A_{Fl} is a set of *Flexible Attributes*. For example, *ZIPCODE* is a Stable Attribute and *User Ratings* can be a Flexible Attribute.

Let us assume from Table 1 that $C \in A_{St}$. A, B $\in A_{Fl}$ and $D \in d$. and the decision maker desires Action Rules that triggers the decision attribute change from D_1 to D_2 throughout this paper for examples.

3.2 Information System

Consider a information system given in table 2. Information system can be represented as $Z = (X, M, V)$where,

X is set of objects $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ in the system;

M is non-empty finite set of attributes $\{A, B, C, E, F, G, D\}$;

V is the domain of attributes in M, for instance the domain of attribute B in the system Z is $\{B_1, B_2, B_3\}$.

Table 2. Information System Z

X	A	B	C	E	F	G	D
x_1	A_1	B_1	C_1	E_1	F_2	G_1	D_1
x_2	A_2	B_1	C_2	E_2	F_2	G_2	D_3
x_3	A_3	B_1	C_1	E_2	F_2	G_3	D_2
x_4	A_1	B_1	C_2	E_2	F_2	G_1	D_2
x_5	A_1	B_2	C_1	E_3	F_2	G_1	D_2
x_6	A_2	B_1	C_1	E_2	F_3	G_1	D_2
x_7	A_2	B_3	C_2	E_2	F_2	G_2	D_2
x_8	A_2	B_1	C_1	E_3	F_2	G_3	D_2

The information system in table 2 becomes a Decision System if the attributes M are classified into flexible attributes M_{fl} , stable attributes M_{st} and decision attributes d, $M = (M_{st}, M_{fl}, \{d\})$.

From table 2 $M_{st} = \{A, B, C\}$, $M_{fl} = \{E, F, G\}$, and $d = D$.

3.3 Action Rules

In this subsection, we give definitions of action terms, action rules and the properties of action rules [34]

Let $\mathbb{T} = (\mathbb{X}, \mathbb{A} \cup d, \mathbb{V})$ be a decision system, where d is a decision attribute and $\mathbb{V} = \cup V_i : i \in \mathbb{A}$. Action terms can be given by the expression of $(m, m_1 \rightarrow m_2)$, where $m \in A$ and $m_1, m_2 \in V_m$. $m_1 = m_2$ if $m \in A_{St}$. In that case, we can simplify the expression as (m, m_1) or $(m = m_1)$. Whereas, $m_1 \neq m_2$ if $m \in A_{Fl}$

Action Rules can take the form of $t_1 \cap t_2 \cap .... \cap t_n$, where t_i is an atomic action or action term and the Action Rule is a conjunction of action terms to achieve the desired action based on attribute D. Example Action Rule is given below: $(a, a_1 \rightarrow a_2).(b, b_1 \rightarrow b_2) \longrightarrow (D, D_1 \rightarrow D_2)$

Properties of Action Rules Action Rules are considered interesting based on the metrics such as Support, Confidence, Coverage and Utility. Higher these values, more interesting they are to the end user.

Consider an action rule $\mathcal{R}$ of form:

$(Y_1 \rightarrow Y_2) \longrightarrow (Z_1 \rightarrow Z_2)$ where,

Y is the condition part of $\mathcal{R}$

Z is the decision part of $\mathcal{R}$

Y_1 is a set of all left side action terms in the condition part of $\mathcal{R}$

Y_2 is a set of all right side action terms in the condition part of $\mathcal{R}$

Z_1 is the decision attribute value on left side

Z_2 is the decision attribute value on right side

In [34], the support and confidence of an action rule $\mathcal{R}$ is given as
$Support(\mathcal{R}) = min\{card(Y_1 \cap Z_1), card(Y_2 \cap Z_2)\}$

$Confidence(\mathcal{R}) = [\frac{card(Y_1 \cap Z_1)}{card(Y_1)}] \cdot [\frac{card(Y_2 \cap Z_2)}{card(Y_2)}]$

Later, Tzacheva et.al [10] proposed a new set of formula for the calculation of Support and Confidence of Action Rules. Their idea is to reduce the complexities in searching data several times for Support and Confidence of an Action Rule. The new formula are given below.
$Support(\mathcal{R}) = \{card(Y_2 \cap Z_2)$

$Confidence(\mathcal{R}) = [\frac{card(Y_2 \cap Z_2)}{card(Y_2)}]$

Tzacheva et. al [10] also introduced a concept of utility for Action Rules. Utility of Action Rules takes a following form. For most of cases Utility of Action Rules equals the Old Confidence of the same Action Rule.
$Utility(\mathcal{R}) = [\frac{card(Y_1 \cap Z_1)}{card(Y_1)}]$

Coverage of an Action Rule means that how many decision from values, from the entire decision system S, are being fully covered by all extracted Action Rules. In other words, using the extracted Action Rules, *Coverage* defines how many data records in the decision system can successfully transfers from Z_1 to Z_2

3.4 Cost of Action Rules

Generally, there is a cost associated with changing an attribute value from one class to another class- the more desirable one. The cost is a subjective measure, in a sense that domain knowledge from the experts or user in the field is necessary in order to determine the costs associated with taking the actions. Costs could be moral,monetary, or a combination of the two. For example,changing the marital status from 'married' to 'divorced' has a moral cost; whereas ,lowering the interest percent rate for a customer is a monetary cost for the bank; in addition to any monetary costs which may be incurred in the process. Feasibility is an objective measure, i.e. domain independent.

According to the cost of actions associated with the classification part of the action rules, a business user may be unable or unwilling to proceed with them.

The definition of cost was introduced by Tzacheva and Ras [3] as follows:

Assume that $S = (X, A, V)$ is an information system. Let $Y \subseteq X$, $b \in A$ is a *flexible* attribute in S and $v_1, v_2 \in V_b$ are its two values. By $\wp_S(b, v_1 \to v_2)$ we mean a number from $(0, \omega]$ which describes the average cost of changing the attribute value v_1 to v_2 for any of the qualifying objects in Y. These numbers are provided by experts. Object $x \in Y$ qualifies for the change from v_1 to v_2, if $b(x) = v_1$. If the above change is not feasible, then we write $\wp_S(b, v_1 \to v_2) = \omega$. Also, if $\wp_S(b, v_1 \to v_2) < \wp_S(b, v_3 \to v_4)$, then we say that the change of values from v_1 to v_2 is more feasible than the change from v_3 to v_4. Assume an action rule r of the form:

$$(b1, v_1 \to w_1) \wedge (b2, v_2 \to w_2) \wedge \ldots \wedge (bp, v_p \to w_p) \Rightarrow (d, k_1 \to k_2)$$

If the sum of the costs of the terms on the left hand side of the action rule is smaller than the cost on the right hand side, then we say that the rule r is *feasible*.

3.5 Meta Action

As an action rule can be seen as a set of atomic actions that need to be made happen for achieving the expected result, meta-actions are the actual solutions that should be

Table 3. Meta-actions Influence Matrix for $\mathbb{S}$

	a	**b**	**d**
M_1, M_2, M_3		$(b_1 \rightarrow b_2)$	$(d_1 \rightarrow d_2)$
M_1, M_3, M_4	(a_2)	$(b_2 \rightarrow b_3)$	
M_5	(a_1)	$(b_2 \rightarrow b_1)$	$(d_2 \rightarrow d_1)$
M_2, M_4		$(b_2 \rightarrow b_3)$	$(d_1 \rightarrow d_2)$
M_1, M_5, M_6		$(b_1 \rightarrow b_3)$	$(d_1 \rightarrow d_2)$

executed to trigger the corresponding atomic actions, Table 3 below shows an example of influence matrix which describes the relationships between the meta-actions and atomic actions influenced by them.

3.6 Spark

Spark [12] is a framework that is quite similar to MapReduce [11] to process large quantity of data in a parallel fashion. Spark introduces a distributed memory abstraction strategy called Resilient Distributed Datasets(RDD) that can perform in-memory computations on nodes distributed in a cluster. Results of each operation are then stored in memory itself, which can be accessed for future processes and analyses, which in-turn creates another RDD. Thus, Spark cuts-off the larger number of disk accesses for storing intermediate outputs like in Hadoop MapReduce. Spark functions in two stages: 1. *Transformation*, 2. *Action*. During the *Transformation* stage, computations are made on data splits and results are stored in the worker nodes memory as RDD. While the *Action* stage on an RDD collect results from all the workers and send it to the driver node or save the results to a storage unit. With RDDs Spark helps machine learning algorithms to skip innumerable disk access during iterations.

4 Dataset Description

To test our methods, we use two datasets: *Student Survey Data* [16], and the Net Promoter Score dataset data [18].

Student survey data aims to evaluate student emotions and overall satisfaction with course teaching methods and group work experience. The survey is designed to get meaningful insights on students' feelings towards the Active Learning methods and other factors that can help students in their learning process. The data is collected in the courses which implement the Active Learning methods and teaching style. This survey dataset contains 50 attributes. The original data contains 549 instances and 59 attributes. Data is collected in classes employing Active Learning methods to assess student opinions about their learning experience in the years 2019, 2020. The data size on disk is 59 Kilobytes. For scalability purpose to test the performance of our proposed method with BigData, we replicate the original Student Survey Data 100 times. The replicated dataset has a total of 54900 instances. Size on disk is 5.815 Megabytes.

We also used a sample of Net Promoter Score dataset [18] for our experiments. The NPS (Net Promoter Score) dataset is collected customer feedback data related to heavy equipment repair. The entire dataset consists of 38 companies, located in multiple sites across the whole United States as well as several parts of Canada. Overall, there are about 340,000 customers surveyed in the database over time span of 2011-2015. Customers were randomly selected to answer a questionnaire which was specifically designed to collect information relevant to NPS (structured into so-called "benchmarks"). All the responses from customers were saved into database with each question (benchmark) as one feature in the dataset. Benchmarks include numerical scores (0-10) on certain aspects of service: e.g. if job done correctly, are you satisfied with the job, likelihood to refer, etc. The dataset also contains customer details (name, contact, etc.) and service details (company, invoice, type of equipment repaired, etc.). The decision attribute in the dataset is $PromoterStatus$ which labels each customer as either $promoter, passive$ or $detractor$. The decision problem here is to improve customer satisfaction / loyalty as measured by Net Promoter Score. The goal of applying action rules to solve the problem is to find minimal sets of actions so that to "reclassify" customer from "Detractor" to "Promoter" and the same improve NPS.

For our experiments, we used survey given by customers for 2 companies over the year of 2015. We have used 17-california and 30-35 datasets for our method. Each of NPS data consists of around 1500 unique surveys from multiple customers with around 25 unique questions. The original data for 17-california contains 547 instances and 23 attributes and the dataset for company 30-35 contains 3335 instances and 23 attributes.

5 Methodology

In our paper, we propose Action Rule extraction techniques to generate action rules. graph-based method to search for optimal low cost Action Rules.In this section, the algorithm for Action Rules are described wisely.

5.1 Distributed Action rules extraction algorithm

In this work,for the extraction technique we focused on distributed Association Action Rules [40] in order to extract the actionable knowledge from big data using Spark framework. Association Action Rules method is not appropriate for big data due to high dimensional data and lacks efficiency in run time. By using the vertical data partitioning technique as proposed in [40], we create partitions of data sets by splitting the data according by attributes in a high dimension data. We perform Association Action Rule extraction algorithm on each partitions of data in parallel, which allows much faster computational time for Association Action Rules extraction in Cloud platforms.

Association Action Rules algorithm is quite similar to Association Rules extraction algorithm with the A-priori method [41]. Association Rules find patterns that occur most frequently together in the given data set. The most popular algorithm for extracting Association Rules is Apriori algorithm [42]. Apriori algorithm starts with 2 element pattern and continues n iterations until it finds the n element patterns, where n is the number of attribute in the given data set. Sample Association rule, that means when a pattern $a_1 \cap b_2$ occur together in the data, pattern $c_1 \cap d_2$ also occurs in the same data, are given below.

Figure 1 presents an example vertical data partitioning with the sample Decision system in Table 1. The actionable knowledge extraction algorithm runs separately on each data partition, does transformations like *map()*, *flatmap()* functions and combine results with *join()* and *groupBy()* operations. We later combine action rules from different partitions to get the final set of action rules.

5.2 Vertical Split - Data Distribution for Scalable Association Action Rules

Authors Bagavathi et al. [40] in method 2 propose the extraction of Action Rules basically by splitting the data in vertical order, which is in contrast to traditional horizontal split, which is performed by parallel processing systems. This method follow Association Action

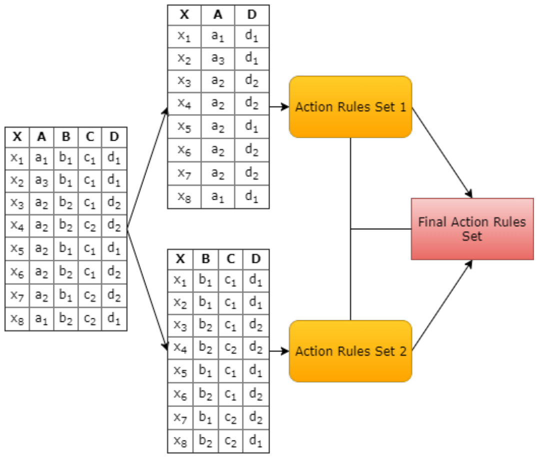

Fig. 1. Example Vertical Data Distribution for Table 1

Rules [8] which is based on iterative method to extract all the possible action rules.To overcome the expense and computational complexity , the authors in [40] proposed vertical data split method for parallel processing along with faster computation. In this method, the data is split in vertical order into 2 or more partitions, with each partition having only a small subset of larger attributes. Fig. 1explains the example of Data partitioning using Vertical Data Distribution in *Distributed Action rules extraction algorithm*,the first section of methodology.

5.3 Hybrid Action Rule Mining

There is a disadvantage of computing preexisting decision rules in generating the Action Rule by Rule-Based method using LERS [43] .The process requires complete set of attributes which is difficult to implement in distributed cloud environment.

We can implement the Object-Based method in distributed cloud environment by splitting the data vertically[40], where subsets of the attributes are taken for scalability. However, since this method is iterative it takes longer time to process huge datasets.

This approach-Hybrid Action Rule mining [44] combines the Rule-Based and Object-Based methods to generate complete set of Action Rules. It provides better performance and scalability for large datasets,in compare to Iterative Association Action Rule approach. The pseudocode of the algorithm is given bellow in the Fig. 2.

The Algorithm approaches with the Information System as follows. The information system in table 2 contains the following attributes: flexible P_{fl} , stable P_{st} and decision d, $P = (P_{st}, P_{fl}, \{d\})$. From table 2 $P_{st} = \{A, B, C\}$, $P_{fl} = \{E, F, G\}$, and $d = D$.

The following example re-directs the decision attribute D from $d_2 \rightarrow d_1$. The algorithm Fig. 2. to extract the classification rules that are certain initially uses the LERS method and then generates Action Rule schema as given in the following equations " 3" ," 4".

$$[\mathrm{B}_1 \wedge \mathrm{C}_1 \wedge (\mathrm{F}, \rightarrow \mathrm{F}_1) \wedge (\mathrm{G}, \rightarrow \mathrm{G}_1)] \rightarrow (\mathrm{D}, \mathrm{D}_2 \rightarrow \mathrm{D}_1). \tag{3}$$

$$[(\mathrm{E}, \rightarrow \mathrm{E}_1)] \rightarrow (\mathrm{D}, \mathrm{D}_2 \rightarrow \mathrm{D}_1). \tag{4}$$

The algorithm then creates sub-table for each of the Action Schema. For example " 3", generates the following sub-table shown in table 4.

```
Algorithm(certainRules, decisionFrom, decisionTo, support, confidence)
    (where certainRules are provided by algorithm LERS)
    for each rule r in certainRules
        if consequent(r) equals decisionTo
            Form ActionRuleSchema(r)
            ARS ← ActionRuleSchema(r)
        end if
    end for
    for each schema in ARS
        Identify objects satisfying schema
        Form subtable
        Generate frequent action sets using Apriori
        Combine frequent action set to form Action Rules
        (Such that the frequent action sets satisfy the decisionFrom → decisionTo)
        Output ← Action Rules
    end for
```

Fig. 2. Hybrid Action Rule Mining Algorithm.

The Hybrid Action Rule Mining Algorithm involves the Association Action Rule extraction algorithm in parallel on each of the sub-tables. The algorithm generates the following Action Rules Equation " 5" based on the sub-table shown in table 4.

Table 4. Subtable for Action Rule Schema

X	B	C	F	G	D
x_1	B_1	C_1	F_2	G_1	D_1
x_3	B_1	C_1	F_2	G_3	D_2
x_6	B_1	C_1	F_3	G_1	D_2
x_8	B_1	C_1	F_2	G_3	D_2

$$[\mathrm{B}_1 \wedge \mathrm{C}_1 \wedge (\mathrm{F}, \rightarrow \mathrm{F}_1) \wedge (\mathrm{G}, \mathrm{G}_3 \rightarrow \mathrm{G}_1)] \rightarrow (\mathrm{D}, \mathrm{D}_2 \rightarrow \mathrm{D}_1). \tag{5}$$

This Hybrid Action Rule algorithm is implemented in Spark [45] and runs separately on each of the sub-table and performs the transformations like map(), flatmap(), join(). The method of this algorithm is shown in Fig. 3. Our new Threshold algorithm method Fig. 4

5.4 Modified Hybrid Action Rule Mining with Partition Threshold Rho

We propose a Modified Hybrid Action Rule Mining with Partition Threshold Rho which provides scalability with big data. It presents a significant improvement over the previous method - Hybrid Action Rule Mining, which has a major disadvantage. If the Size of the Intermediate Table becomes very large it affects the performance and the scalability of

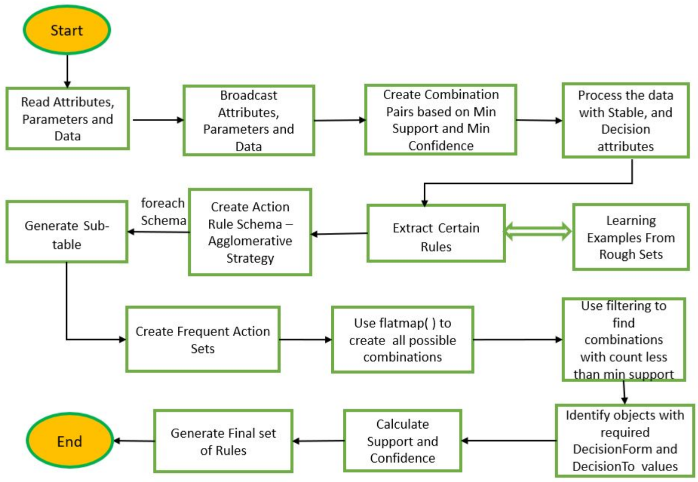

Fig. 3. Hybrid Action Rule Mining Algorithm - Flowchart.

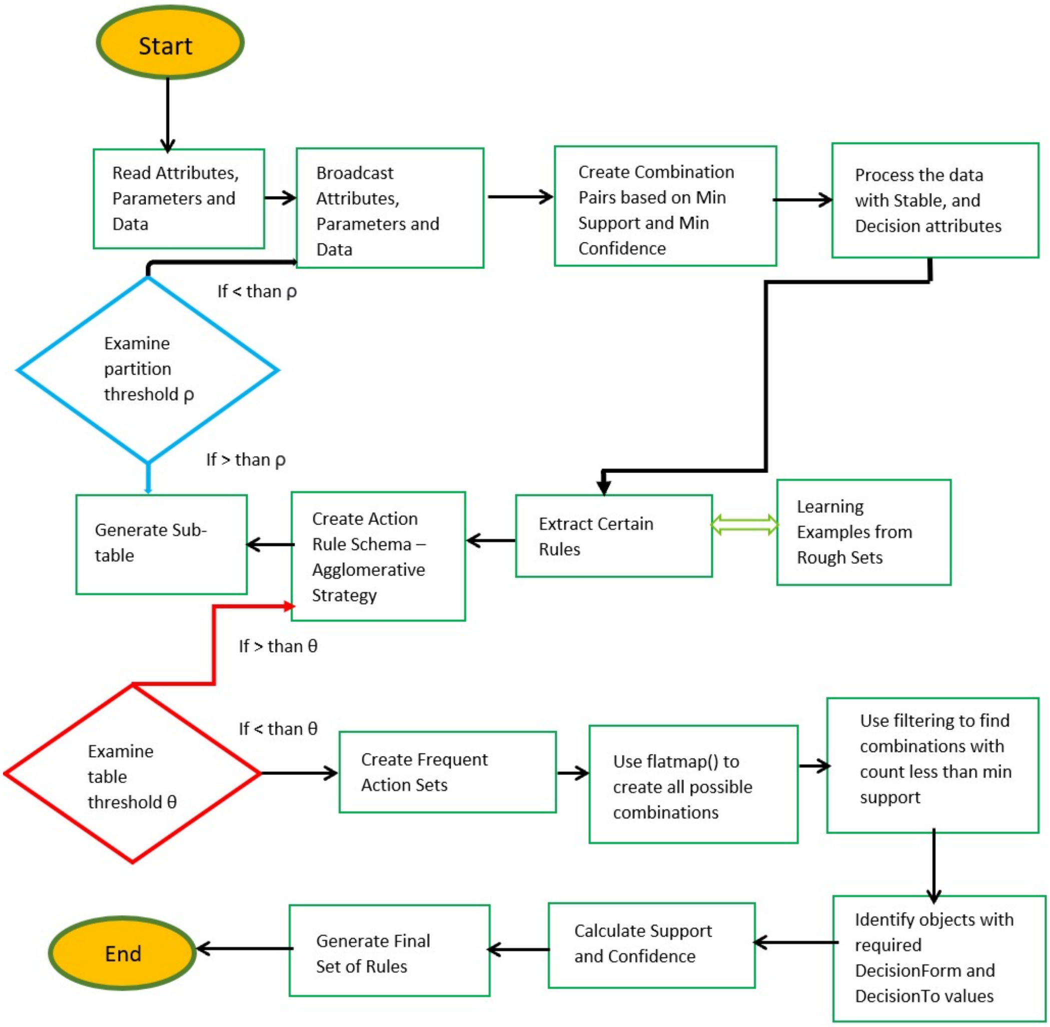

Fig. 4. Hybrid Action Rule Mining Algorithm(New Threshold) - Flowchart.

```
Algorithm(certainRules, decisionFrom, decisionTo, support, confidence)
    (where certainRules are provided by algorithm LERS)
    for each rule r in certainRules
        if consequent (r) equals desisionTo
                Form ActionRuleSchema (r)
                ARS <- ActionRuleSchema (r)
        end if
    end for
    for each schema in ARS
            Identify objects satisfying schema
            Form partition
            While partition size > Rho ρ
            Form subtable
            While subtable size > Theta θ
                Divide subtable until subtable < Theta θ
            Generate frequent action sets using Apriori
            Combine frequent action set to form Action Rules
            (such that the frequent action sets satisfy the
                decisionFrom -> decisonTo)
            Output <- Action Rules
    end for
```

Fig. 5. Hybrid Action Rule Mining with Threshold Algorithm.

this method. Our proposed new method solves this problem, as the Threshold ρ allows the user to control the size of the table and it increases the computational speed.

Our proposed method - Modified Hybrid Action Rule Mining with Partition Threshold Rho - is presented in the Fig. 5 and the proposed methodology is depicted in the Fig. 4.

6 Experiments and Results

In this work we use, student survey data which focus on student emotions. We applied the data in all the three experiments and compared the Computational time. We also used a sample of NPS (Net Promoter Score) data [18] for our experiments that aims to evaluate Promoter Status. We applied NPS (Net Promoter Score) data for Vertical Split - Data Distribution method only. See section IV. Dataset Description.

We compare our proposed method with the Vertical Split - Data Distribution for Scalable Association Action Rules method and Hybrid Action Rule mining method. We achieve faster computational time through our new proposed method for Student Survey method.

6.1 Experiment 1 - Vertical Data Split Method Implementation in Spark AWS Cluster

We perform this experiment on the Student Survey Data - using Amazon Web Services (AWS) cluster with two nodes, 4 vCore and 16GiB memory and EBS Storage 64GiB. For very large data this method requires additional resources. We find we must provide extra 32 Gigabytes of memory to complete computation on the replicated data in 2400 seconds. Otherwise, the method receives OutOfMemory Exception with our replicated Student Survey Data. This occurs because of iterative nature of the algorithm with large data that causes computational overhead and requires extra hardware memory resources to work successfully. This method only works for Association Action Rules because it considers only subset of the attributes.

Selected Action Rules generated by this experiment are shown in table 5.

The Action Rule 1 says that when GroupAssignmentBenefit changes from SharedKnowledge to SocialLearning and LikeTeamWork changes from 1Don't to 5VeryMuch and TeamMemberResponsibility changes from HelpfulMembers to ResponsibleMembers then the StudentEmotion changes from Sadness to Joy. This shows that when the Student likes TeamWork and the group contains Responsible TeamMembers and benefits from GroupAssignment then it enhances the Student's Emotion from Sadness to Joy.

We experiment with NPS (Net Promoter Score) Business data and extract Action Rules from it using the Verticle Data Split Method. Sample results are shown in table 6. The Action Rule 1 says that when BenchmarkPartsOrderAccuracy changes from 3 to 10 then the PromoterStatus changes from Detractor to Promoter. This shows that when a Customer enjoys good BenchmarkPartsOrderAccuracy then his/her Status is enhanced from Detractor to Promoter. We have 62 percentage confidence in this rule. We plan to continue this experiment with NPS (Net Promoter Score) Business data by applying the Hybrid Method and Modified Hybrid Method with Partition Threshold Rho.

6.2 Experiment 2 - Hybrid Method Implementation in Spark AWS Cluster

We perform this experiment on the Student Survey Data with Hybrid Action Rule Mining Method - using Amazon Web Services (AWS) cluster with two nodes, 4 vCore and 16GiB memory and EBS Storage 64GiB. This method takes 5088 seconds to complete computation on our replicated Student Survey Data.

Selected Action Rules generated by this experiment are shown in 7 .

The Action Rule 1 in table 7 says when TeamSenseOfBelonging changes from 2BelowAverageSenseofBelongingtotheTeam to 3AverageSenseofBelongingtotheTeam and the NumberofTeamMembers changes from 5to7 to 10orMore then the StudentEmotion changes from Sadness to Joy. This rule has support of 20 and confidence of 59%.This shows that when the Student has an average sense of belonging to the Team and the team contains 10orMore members then it enhances the Student's Emotion from Sadness to Joy.

6.3 Experiment 3 - Modified Hybrid Action Rule Mining with Partition Threshold Rho Implementation in Spark AWS Cluster

We perform this experiment on the Student Survey Data with our proposed Modified Hybrid Action Rule Mining Method - using Amazon Web Services (AWS) cluster with two nodes, 4 vCore and 16GiB memory and EBS Storage 64GiB. Our proposed method takes 3900 seconds to complete computation on the replicated Student Survey Data. We experiment with 3 different Threshold values of ρ ::: 5, 10 and 15 and θ ::: 5, 10 and 15.

The runtime comparison for different Threshold values for two different thresholds θ and ρ implemented on Student Survey data is shown in the below table 9

For Student Survey data Threshold value of $\rho = 5$ and $\theta = 15$ provides optimum performance.

Selected Action Rules generated by this method are shown in 8 .

The Action Rule 1 in table 8 says when TeamFormation changes from 2BelowAverage to 4Perfect and the NumberofTeamMembers changes from 5to7 to 8to10 then the StudentEmotion changes from Sadness to Joy. This rule has support of 21 and confidence of 62%. This shows how having a good team and increased number of Team Members enhances a Student's Emotion from Sadness to Joy.

Enhance Student Emotion - Sadness → Joy
1. $AR1SadnesstoJoy : (GroupAssignmentBenefit, SharedKnowledge \rightarrow SocialLearning) \wedge (LikeTeamWork, 1Don't \rightarrow 5VeryMuch) \wedge (TeamMemberResponsibility, HelpfulMembers \rightarrow ResponsibleMembers) \Longrightarrow (StudentEmotion, Sadness \rightarrow Joy)[Support : 100.0, Confidence : 75.0\%]$ 2. $AR2SadnesstoJoy : (GroupAssignmentBenefit, None \rightarrow None) \wedge (LikeTeamWork, 3Somewhat \rightarrow 5VeryMuch) \wedge (TeamMemberResponsibility, TechnicallyIneffectiveMembers \rightarrow FriendlyMembers) \Longrightarrow (StudentEmotion, Sadness \rightarrow Joy)[Support : 100.0, Confidence : 50.0\%]$ 3. $AR3SadnesstoJoy : (NumberofTeamMembers, 8to10 \rightarrow 10orMore) \wedge (LikeTeamWork, 3Somewhat \rightarrow 5VeryMuch) \wedge (GroupAssignmentBenefit, None \rightarrow SocialLearning) \Longrightarrow (StudentEmotion, Sadness \rightarrow Joy)[Support : 100.0, Confidence : 16.6\%]$

Table 5. Sample Action Rules ::: Sadness to Joy ::: - Student Survey Data - Vertical Data Split Method.

Enhance Customer Emotion - Detractor → Promoter
1. $AR1DetractortoPromoter : (BenchmarkPartsOrderAccuracy, 3 \rightarrow 10) \Longrightarrow (PromoterStatus, Detractor \rightarrow Promoter)[Support : 2.0, Confidence : 62.19\%]$ 2. $AR2DetractortoPromoter : (BenchmarkPartsHowOrdersArePlaced, 2 \rightarrow 4) \wedge (BenchmarkPartsOrderAccuracy, 3 \rightarrow 10) \Longrightarrow (PromoterStatus, Detractor \rightarrow Promoter)[Support : 2.0, Confidence : 100.00\%]$ 3. $AR3DetractortoPromoter : (BenchmarkPartsPartsAvailability, 4 \rightarrow 9) \wedge (Division, WagnerHeavyEquipment - Parts \rightarrow WagnerHeavyEquipment - Parts) \Longrightarrow (PromoterStatus, Detractor \rightarrow Promoter)[Support : 2.0, Confidence : 79.48\%]$

Table 6. Sample Action Rules ::: Detractor to Promoter ::: - NPS (Net Promoter Score) Business data - Vertical Data Split Method.

Enhance Student Emotion - Sadness → Joy
1. $AR1SadnesstoJoy : (TeamSenseofBelonging, 2BelowAverageSenseofBelongingtotheTeam \rightarrow 3AverageSenseofBelongingtotheTeam) \wedge (NumberofTeamMembers, 5to7 \rightarrow 10orMore) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 20.0, Confidence : 59.0\%]$ 2. $AR2SadnesstoJoy : (NumberofTeamMembers, 5to7 \rightarrow 8to10) \wedge (TeamWorkHelpedDiversity, 2Occasionally \rightarrow 3Often) \wedge (GroupAssignmentBenefit, None \rightarrow AllofThem) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 20.0, Confidence : 100\%]$ 3. $AR3SadnesstoJoy : (NumberofTeamMembers, 5to7 \rightarrow 8to10) \wedge (GroupAssignmentBenefit, None \rightarrow SharedKnowledge) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 34.0, Confidence : 85.0\%]$

Table 7. Sample Action Rules ::: Sadness to Joy ::: - Student Survey Data - Hybrid Method.

Enhance Student Emotion - Sadness → Joy
1. $AR1SadnesstoJoy : (TeamFormation, 2BelowAverage \rightarrow 4Perfect) \wedge (NumberofTeamMembers, 5to7 \rightarrow 8to10) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 21.0, Confidence : 62.0\%]$ 2. $AR2SadnesstoJoy : (LikeTeamWork, 1Don't \rightarrow 3Somewhat) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 21.0, Confidence : 91.0\%]$ 3. $AR3SadnesstoJoy : (NumberofTeamMembers, 5to7 \rightarrow 8to10) \wedge (GroupAssignmentBenefit, None \rightarrow SocialLearning) \implies (StudentEmotion, Sadness \rightarrow Joy)[Support : 34.0, Confidence : 85.0\%]$

Table 8. Sample Action Rules ::: Sadness to Joy ::: - Student Survey Data - Hybrid Method with Threshold.

Table 9. Threshold values - ρ and θ Run Time for Student Survey Data

Threshold	Threshold	Time in Second
5	5	447
5	10	355
5	15	352
10	5	452
10	10	643
10	15	455
15	5	354
15	10	354
15	15	353

6.4 Runtime Comparison of the above 3 implementations with respect to Student Survey Data in Spark AWS Cluster

We compare the execution runtime of the above described implementations:Vertical Data Split Method in Spark AWS Cluster, Hybrid Method Implementation in Spark AWS Cluster and Hybrid Method with Threshold Implementation in Spark AWS Cluster. The runtimes are given in below table 10 .

Our proposed Hybrid Method with Threshold (Modified Hybrid Method with threshold rho) shows improved performance over the previous Hybrid Method, and shows the best performance with standard memory.

7 Conclusion

The ultra-connected world is generating massive volumes of data stored in a computer database and cloud environment. These huge large datasets need to be analyzed in order to extract useful knowledge and present it to decision makers for further use. Data mining techniques and extracting patterns from large data plays a vital role in knowledge discovery. Most of the decision makers encounter a large number of decision rules resulted from action rules mining. Moreover, the volume of datasets brings a new challenges to extract patterns - such as high cost of computing; or unreasonable time to extract the relevant rules. However emotion analysis has been attracting researcher's attention. Emotions play a very important role in the lives of people all over the world. Today we have multiple platforms available for electronic communication. The expansion of social media, online surveys, customer surveys, blogs, industrial and educational data generates large amounts of data. Hidden in the data are valuable insights on people's opinions and their emotions. We are searching for emotions in data - this can applied to Student Surveys as well Customer Satisfactions opinions such as the NPS (Net Promoter Score) data.

Discovering emotions in text data through Action Rule Mining can benefit industries [46], including Healtcare, Business, Social Media and Education. In this work we apply our proposed method - Modified Hybrid Action Rule Mining with Partition Threshold Rho to Student Survey Dataset and NPS (Net Promoter Score) business Dataset. In our results we suggest ways for improving Customer Emotions that may be a Student or may be a Business person. The Student Survey data contains student opinions regarding the use of Active Learning methods, Teamwork and class experiences. The NPS data contains the customer opinion regarding their service experience with the business. The discovered Action Rules help to enhance the user Emotion from Negative to Positive and from Neutral to Positive.

In general, there are many industrial solutions developed in recent years that are based on aspect-based sentiment analysis and text analytics. However, we recognize the problem remains un-solved. Also, the research focus has been mainly on electronic products, hotels and restaurants. There are still novel ideas needed to study specific domains. Domain and context dependent sentiments remain to be highly challenging. Today, a completely automated and accurate solution is yet to be found. At the same time, there is still a great demand in industry domains for such systems, because every business wants to know how customers perceive their products and services.

8 Future Work

Our proposed method improves the processing time. However, the quality of rules may decrease. In the future, we plan to use Correlation of Attributes and run classical Clustering Algorithm. This obtains optimal Vertical Partitioning which is flexible. We plan to apply Agglomerative strategy to change levels of vertical partitions. We also plan to examine the Quality of the Action Rules using F-Score.

References

1. J. Kuang, A. Daniel, J. Johnston, and Z. W. Ras, "Hierarchically structured recommender system for improving nps of a company," in *International Conference on Rough Sets and Current Trends in Computing.* Springer, 2014, pp. 347–357.
2. J. Ranganathan, A. S. Irudayaraj, A. Bagavathi, and A. A. Tzacheva, "Actionable pattern discovery for sentiment analysis on twitter data in clustered environment," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–15, 2018.
3. Z. W. Raś and A. A. Tzacheva, "In search for action rules of the lowest cost," in *Monitoring, Security, and Rescue Techniques in Multiagent Systems.* Springer, 2005, pp. 261–272.
4. P. Su, D. Li, and K. Su, "An expected utility-based approach for mining action rules," in *Proceedings of the ACM SIGKDD Workshop on Intelligence and Security Informatics*, ser. ISI-KDD '12. New York, NY, USA: ACM, 2012, pp. 9:1–9:4. [Online]. Available: http://doi.acm.org/10.1145/2331791.2331800
5. A. Tzacheva and Z. Ras, "Action rules mining." *Int. J. Intell. Syst.*, vol. 20, pp. 719–736, 01 2005.
6. L.-S. Tsay* and Z. W. Raś, "Action rules discovery: system dear2, method and experiments," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 17, no. 1-2, pp. 119–128, 2005.
7. Z. W. Raś, E. Wyrzykowska, and H. Wasyluk, "Aras: Action rules discovery based on agglomerative strategy," in *International Workshop on Mining Complex Data.* Springer, 2007, pp. 196–208.
8. Z. W. Ras, A. Dardzinska, L.-S. Tsay, and H. Wasyluk, "Association action rules," in *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on.* IEEE, 2008, pp. 283–290.
9. S. Im and Z. W. Raś, "Action rule extraction from a decision table: Ared," in *International Symposium on Methodologies for Intelligent Systems.* Springer, 2008, pp. 160–168.
10. S. R. A.A. Tzacheva, C.C. Sankar and R. Shankar, "Support confidence and utility of action rules triggered by meta-actions," in *proceedings of 2016 IEEE International Conference on Knowledge Engineering and Applications*, ser. ICKEA 2016. IEEE Computer Society, 2016.
11. J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.
12. M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 2–2.
13. J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica, "Graphx: Graph processing in a distributed dataflow framework." in *OSDI*, vol. 14, 2014, pp. 599–613.
14. J. Kuang, A. Daniel, J. Johnston, and Z. W. Raś, "Hierarchically structured recommender system for improving nps of a company," in *Rough Sets and Current Trends in Computing*, C. Cornelis, M. Kryszkiewicz, D. Ślzak, E. M. Ruiz, R. Bello, and L. Shang, Eds. Cham: Springer International Publishing, 2014, pp. 347–357.
15. A. Bagavathi, P. Mummoju, K. Tarnowska, A. Tzacheva, and Z. Ras, ""sargs method for distributed actionable pattern mining using spark"," 12 2017, pp. 4272–4281.

16. A. A. Tzacheva and A. Easwaran, “Modified hybrid scalable action rule mining for emotion detection in student survey data,” in *International Conference on Data Mining, Big Data, Database and Data Technologies*, 2022, pp. 13–19.
17. T. Hakim, K. Jieyan, H. Ayman, and Z. W. Ras, “Personalized action rules for side effects object grouping,” in *International Journal of Intelligence Science.* Scientific Research, 2013, pp. 24–33.
18. Z. W. Ras, K. A. Tarnowska, J. Kuang, L. Daniel, and D. Fowler, “”user friendly nps-based recommender system for driving business revenue”,” in *”Rough Sets”*, 07 2017, pp. 34–48.
19. J. Kuang, Z. W. Ras, and A. Daniel, “Hierarchical agglomerative method for improving nps,” in *Pattern Recognition and Machine Intelligence*, M. Kryszkiewicz, S. Bandyopadhyay, H. Rybinski, and S. K. Pal, Eds. Cham: Springer International Publishing, 2015, pp. 54–64.
20. Z. W. Ras and A. Dardzinska, “Action rules discovery based on tree classifiers and meta-actions,” in *Foundations of Intelligent Systems*, J. Rauch, Z. W. Ras, P. Berka, and T. Elomaa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 66–75.
21. A. A. Tzacheva and Z. W. Ras, “Association action rules and action paths triggered by meta-actions,” in *2010 IEEE International Conference on Granular Computing*, 2010, pp. 772–776.
22. J. Kuang, Z. W. Ras, and A. Daniel, “Personalized meta-action mining for nps improvement,” in *Foundations of Intelligent Systems*, F. Esposito, O. Pivert, M.-S. Hacid, Z. W. Ras, and S. Ferilli, Eds. Cham: Springer International Publishing, 2015, pp. 79–87.
23. N. Esfandiari, M. R. Babavalian, A.-M. E. Moghadam, and V. K. Tabar, “Knowledge discovery in medicine: Current issue and future trend,” *Expert Systems with Applications*, vol. 41, no. 9, pp. 4434–4463, 2014.
24. R. S. Baker and P. S. Inventado, “Educational data mining and learning analytics,” in *Learning analytics.* Springer, 2014, pp. 61–75.
25. G. Shmueli, P. C. Bruce, I. Yahav, N. R. Patel, and K. C. Lichtendahl Jr, *Data mining for business analytics: concepts, techniques, and applications in R.* John Wiley & Sons, 2017.
26. H. C. Koh, G. Tan *et al.*, “Data mining applications in healthcare,” *Journal of healthcare information management*, vol. 19, no. 2, p. 65, 2011.
27. D. He, S. C. Mathews, A. N. Kalloo, and S. Hutfless, “Mining high-dimensional administrative claims data to predict early hospital readmissions,” *Journal of the American Medical Informatics Association*, vol. 21, no. 2, pp. 272–279, 2014.
28. B. Zheng, J. Zhang, S. W. Yoon, S. S. Lam, M. Khasawneh, and S. Poranki, “Predictive modeling of hospital readmissions using metaheuristics and data mining,” *Expert Systems with Applications*, vol. 42, no. 20, pp. 7110–7120, 2015.
29. B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore, “Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records,” *BioMed research international*, vol. 2014, 2014.
30. P. Braga, F. Portela, M. F. Santos, and F. Rua, “Data mining models to predict patient’s readmission in intensive care units,” in *ICAART 2014-Proceedings of the 6th International Conference on Agents and Artificial Intelligence*, vol. 1. SCITEPRESS, 2014, pp. 604–610.
31. M. Al-Mardini, A. Hajja, L. Clover, D. Olaleye, Y. Park, J. Paulson, and Y. Xiao, “Reduction of hospital readmissions through clustering based actionable knowledge mining,” in *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on.* IEEE, 2016, pp. 444–448.
32. M. Almardini, A. Hajja, Z. W. Raś, L. Clover, D. Olaleye, Y. Park, J. Paulson, and Y. Xiao, “Reduction of readmissions to hospitals based on actionable knowledge discovery and personalization,” in *Beyond Databases, Architectures and Structures. Advanced Technologies for Data Mining and Knowledge Discovery.* Springer, 2015, pp. 39–55.
33. H. Liu, A. Gegov, and F. Stahl, “Categorization and construction of rule based systems,” in *International Conference on Engineering Applications of Neural Networks.* Springer, 2014, pp. 183–194.
34. Z. W. Ras and A. Wieczorkowska, “Action-rules: How to increase profit of a company,” in *European Conference on Principles of Data Mining and Knowledge Discovery.* Springer, 2000, pp. 587–592.
35. Q. Yang, J. Yin, C. Ling, and R. Pan, “Extracting actionable knowledge from decision trees,” *IEEE Transactions on Knowledge and data Engineering*, vol. 19, no. 1, pp. 43–56, 2007.
36. M. Karim and R. M. Rahman, “Decision tree and naive bayes algorithm for classification and generation of actionable knowledge for direct marketing,” *Journal of Software Engineering and Applications*, vol. 6, no. 04, p. 196, 2013.
37. Z. Cui, W. Chen, Y. He, and Y. Chen, “Optimal action extraction for random forests and boosted trees,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2015, pp. 179–188.
38. Y. Hu, J. Du, X. Zhang, X. Hao, E. Ngai, M. Fan, and M. Liu, “An integrative framework for intelligent software project risk planning,” *Decision Support Systems*, vol. 55, no. 4, pp. 927–937, 2013.

39. Y. Hu, B. Feng, X. Mo, X. Zhang, E. Ngai, M. Fan, and M. Liu, "Cost-sensitive and ensemble-based prediction model for outsourced software project risk prediction," *Decision Support Systems*, vol. 72, pp. 11–23, 2015.
40. A. Bagavathi, V. Rao, and A. A. Tzacheva, "Data distribution method for scalable actionable pattern mining," in *Proceedings of the First International Conference on Data Science, E-learning and Information Systems*. ACM, 2018, p. 3.
41. M. Hahsler and R. Karpienko, "Visualizing association rules in hierarchical groups," *Journal of Business Economics*, vol. 87, no. 3, pp. 317–335, 2017.
42. S. Rathee, M. Kaul, and A. Kashyap, "R-apriori: an efficient apriori based algorithm on spark," in *Proceedings of the 8th Workshop on Ph. D. Workshop in Information and Knowledge Management*. ACM, 2015, pp. 27–34.
43. J. W. Grzymala-Busse, "A new version of the rule induction system lers," *Fundamenta Informaticae*, vol. 31, no. 1, pp. 27–39, 1997.
44. A. Tzacheva and J. Ranganathan, ""pattern discovery from student feedback: Identifying factors to improve student emotions in learning"," vol. 14, no. 12. World Academy of Science, Engineering and Technology, 2020, pp. 1260 – 1265. [Online]. Available: https://publications.waset.org/vol/168
45. M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 10–10. [Online]. Available: http://dl.acm.org/citation.cfm?id=1863103.1863113
46. A. F. A. Nasir, E. S. Nee, C. S. Choong, A. S. A. Ghani, A. P. P. A. Majeed, A. Adam, and M. Furqan, "Text-based emotion prediction system using machine learning approach," *IOP Conference Series: Materials Science and Engineering*, vol. 769, p. 012022, jun 2020. [Online]. Available: https://doi.org/10.1088%2F1757-899x%2F769%2F1%2F012022

Authors

Sanchari Chatterjee received MS degree from University of North Carolina, Charlotte, in computer science. Currently, she is pursuing her PhD in Computer Science from the University of University of North Carolina, Charlotte. Her research interests include Data Mining,knowledge discovery in databases, Machine Learning, Artificial Intelligence.

Angelina A. Tzacheva is currently a Professor, and Ph.D. Program Director,Department of Computer Science and CyberSecurity School of Technology and Engineering,National University, San Diego. Her research interests include data mining,knowledge discovery in databases, distributed knowledge systems, medical imaging, multimedia databases, bioinformatics.

Zbigniew Ras Dr. Zbigniew Ras is a professor of computer science at the University of North Carolina, Charlotte. He is the director of KDD(knowledge discovery in databases) Laboratory at University of North Carolina, Charlotte. He also holds professorship position in the Institute of Computer Science at the Polish-Japanese Academy of Information Technology in Warsaw, Poland. His PhD degree is from University of Warsaw and DSc Degree from the Polish Academy of Sciences. In 2012, he was awarded National Professorship Title by the President of Poland. Dr. Ras' areas of specialization include Knowledge Discovery and Data Mining, Recommender Systems, Health Informatics, Business Analytics, Flexible Query Answering, Music Information Retrieval, and Art.

Method	Time Taken
Vertical Data Split Method _* with additional resources: 32 GB cluster memory _* with standard memory	 2400 seconds OutOfMemoryException
Hybrid Method	5088 seconds
Modified Hybrid Action Rule Mining with Partition Threshold Rho	3900 seconds

Table 10. Runtime Comparison of the above 3 implementations with respect to Student Survey Data.

Method	Time Taken
Vertical Data Split Method _* with additional resources: 32 GB cluster memory _* with standard memory	 2400 seconds OutOfMemoryException
Hybrid Method	5088 seconds
Modified Hybrid Method	4002 seconds

Table 11. Runtime Comparison of the above 3 implementations with respect to NPS Data.