Applying Machine Learning to Conflict Management in Software Requirement

Pratap Pal
B.tech, CSE
Amity School Of Engineering & Technology
Amity University, Haryana

Geet Sandhu
Assistant professor
Amity School Of Engineering & Technology
Amity University, Haryana

Atish
B.tech, CSE
Amity School Of Engineering & Technology
Amity University, Haryana

Shally Pal
B.tech, CSE
Amity School Of Engineering & Technology
Amity University, Haryana

ABSTRACT
Software requirements is a field within software engineering that deals with the establishing the needs of the stakeholders [1]. In this paper, the concern is about the new approach and techniques for incorporating precision and consistency in requirements specifications. Besides these software requirements should be ambiguity free and complete by all means. This paper also reviews the existing work about how the ambiguity can be resolved through machine learning algorithms that can learn from the data stored through especially through data analytics. Since Machine learning deals with the issue of how to build the programs that improve their performances at some task through experience and Machine learning algorithms has proven to be of great practical value in variety of application domains. This paper focuses on approach of existing applying machine learning algorithms and their methods to specify software requirements.

General terms
Artificial intelligence, Data analytics , Pattern recognition, Associative learning, ambiguity .

Keywords
Software requirement, machine learning, Requirement Engineering (RE).

1. INTRODUCTION
Specifying software requirement is a description of software system to be developed laying out functional and non-functional requirements. The software requirement arrangement enlists enough and necessary requirements that

are required for project development.[2]. The software requirement process is sometime full of ambiguity , complexity, and invisibility. There are certain sort of learning like associative learning, regression(linear one variable/linear multi variable), and reinforcement that can be helpful in resolving the human errors which are cost effecting. The paper focuses on the discussion of existing algorithms and how it can be implemented in specifying software requirement engineering (RE)[3]. Refers to the process of defining, documenting and maintaining requirements[4][5] and to the subfield of system engineering and software engineering concerned processes. Section 1 is the introduction to the research topic, 2nd section discusses the main challenge faced in this area, 3rd section gives the condition of practice in machine learning is discussed in section 4 which talks about the existing and related work done by other researchers in this area, section 5 gives observations on the current work done , section 6 and 7 concludes the research survey .

2. THE CHALLENGE
Learning data is quite expensive, but large number of unlabelled data can be used which is available in low cost.. This difficulty of making and maintaining and developing large software systems has been brought out in Brooks' classic paper, No Silver Bullet [5]. Software Requirements are difficult to reveal sometimes in the initial stages. It sometimes becomes complicated to foresee or anticipate the expected software solution before hand. Another challenge is stability of requirements because software requirements are vulnerable to change overtime. Clients might modify requirements or change completely. Fluidity in software requirements by the client becomes major problem. Communication barrier between the client and the requirement engineer on different grounds such as language, professional background etc is another major challenge.

3. PRESENT STATE-OF--ART IN MACHINE LEARNING & SOFTWARE REQUIREMENTS
Several areas in software development have already witnessed the use of machine algorithms. We will first take a visual at reported results and offer a classification towards the existing methodology, then analyze the current state of the methods in this niche area, and finally discuss some general issues in ML&SE.

4. RELATED WORK
We have come across during the course of our study that it becomes very vital to know what is the methodology that can be implemented, what characteristics do they posses, circumstances in which it shall be implemented and what could be the theoretical obstacles that can come its way. Various factors are to be considered in software solution. The table discusses those factors or attributes and also specifies corresponding learning technique used by other researchers to respond towards it.

Table 1- Activity and corresponding approach

Activity	Types of ML Algorithm/Approach
1.Software Quality Prediction	GP(Gradient Progression), NN(Neural networks), CBR(Case based reasoning)
2.Fault Prediction	NN(Neural Networks), ILP(Inductive Logic Programming)
3.Risk Management Prediction	DT(Decision Tree)
4.Software Size Estimation	NN(Neural Networks), GP(Gradient Progression)
5.Cost Prediction	DT (Decision Tree), CBR (Case based reasoning)
6.Task effort Prediction	IBL(Instance based Learning), CBR(Case Based Reasoning), DT(Decision Tree), NN(Neural Network), [GA(Genetic Algorithm)+NN(Neural Network)], Regression
7.Sowtware maintenance	ILP(Instance Learning programming)
8.Software Resource Analysis	DT(Decision tree)
9.Softwae Reliability	NN(Neural Network), DT(Decision Tree)
10.Release Timing	NN(Neural Network)

Many learning algorithms for generating association rules were presented over time that can be used to specify the requirements of a complex system which is using a branches of sophisticated software which are linked with each

4.1 Apriori algorithm

Agrawal, Rakesh and Srikant, Ramakrishnan et al proposed Apriori[6] which is the best-known algorithm to association rules. It uses a -stretch first search technology to count the support of itemsets and uses a candidate generation function which exploits the downward closure property of support. Apriori(T,ϵ) $C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \in \cup L_{k-1} \wedge b \notin a\}$

$count[c] \leftarrow count[c]+1$

$C_t \leftarrow \{c \mid c \in c_k \wedge c \subseteq t\}$, **for** candidates $c \in C_t$,**for** transaction $t \in T$, $k \leftarrow 2$,$k \leftarrow k+1$

$L_1 \leftarrow \{\text{large } 1 - \text{itemsets}\}$

$L_k \leftarrow \{c \mid c \in C_k \wedge count[c] \geq \epsilon\}$

return $U_k L_k$,**while** $L_{k-1} \neq \emptyset$

4.2 Eclat algorithm

Zaki, M. J et al proposed Eclat[7] (alt. ECLAT, stands for equivalence Class transformation) is a depth-first search algorithm using set intersection.

Initialisation: count 2-itemsets(count-Distribution) , (1) Partition L_2,using 1-item prefixes (2) Transform-dbase to vertical layout, **Async:** for each $e \in E_2$

Kathryn l.Heninger [8], proposed a learning algorithm for a A-7 aircraft that specify the software requirements in two respected fields.

Hardware interfaces: Concise description of information received or transmitted by the computer.

Software interfaces: what the software must do to meet its , requirements ,in various situations and in response to various events.

Un-supervised learning is used in these systems which is a branch of machine learning, constraints like: timing constraints, accuracy constraints, Responds to undesired events, subsets , fundamentals Output Values as Functions of Conditions and Events Originally we thought we would describe each output as a mathematical function of input values. This turned out to be a ingenuous approach. we had to define intermediate values that the current program calculated, but that did not correspond to any output values. These in turn had to be described in terms of other intermediate values. By the time we reached input values, we would have described an implementation. Instead, we expressed requirements by giving output values as functions of aircraft operating conditions.

Du. ZHang et al [9] imposed that Machine learning deals with the issue of how to build programs that improve their performance at some task through experience. Machine learning algorithms have prove to be of great practical value in a variety of application domains. The field of software engineering turns out to be a fertile ground where many software development and maintenance tasks could be formulated in terms of learning algorithms.

Alrajeh D,Kramer J, Russoa[10] , et al approach the idea of of requirements engineering includes many activities, from goal elicitation to require specification. The systematic approach for generating a set of operational requirements that are complete with respect to given aim. We show how the amalgamation of model examining and inductive learning can be effectively used to do this. The model corroborating forms verifies the satisfaction of the goals and produces counterexamples when incompleteness in the operational requirements is detected. Kiri L.wagstaff [11], Machine Learning for Machine Learning's Sake This section highlights aspects of the way ML research is conducted today that limit its impact on the larger world. Our goal is not to point fingers or critique individuals, but instead to initiate a critical self-inspection and constructive, creative changes.. The discussion here is not about "theory versus applications. The criticisms here focus instead on the limitations of work that lies between theory and meaningful applications: algorithmic advances accompanied by empirical studies that are divorced from true impact

5. OBSERVATIONS

The table below is the representation of the outcomes of the two algorithm ie., Apriori and Eclat .

Key Issue address	Apriori Algorithm	Eclat Algorithm	(Pros/Cons)
Requirement change during the development time	Less	average	-
Unreasonable timelines	Average	less	-
Cost Optimization	Above average	-	-
complexity	Good	Below average	-

The above table shows that how the Apriori and Eclat algorithm can be used in getting the desirable outcome of various regimes. The Apriori and Eclat in the category of "requirement change during the development time" gives less and average scale, Average and less in the category of "Unreasonable timeless", while in the category of "Cost Optimization " only the Apriori showed up as above average.

Lastly in the "Complexity section" both algorithms showed as good and below average.

6. CONCLUSION

Machine learning approaches applied in software requirements reviews of complex research fields such as quality improvement may assist in the title and abstract inclusion screening process. Machine learning approaches are of particular interest considering steadily increasing search outputs and accessibility of the existing evidence is a particular challenge of the research field quality improvement. Increased reviewer agreement appeared to be associated with improved predictive performance.

7. FUTURE SCOPE

An extra attention is required more specifically in the parts of requirement elicitation, specification and validation. A new approach is necessary for requirement elicitation process based on stakeholder suggestion and mutual filtering, to overcome inconvenient and infrequent during interaction of developers and end-users in different organizations or enterprise. In order to address the problems, Stakeholder suggested model is required that comprise Identification of large project, Analysis of requirements, Identify and prioritize (which) Predict requirements, itemize requirements definite software specification is in need as a short declaration of the requirements that the software must assure. Good requirements are only guaranteed by the correct grouping of three scopes such as community, administration and expertise. In addition, above requirement elicitation and specification are to be validated with a novel technique in terms of quality accuracy and security. Determine challenges in aligning requirements engineering and verification in a operational requirements is detected.

8. ACKNOWLEDGMENT

I would like to thank to my teacher Mrs Geet Sandhu and My co-authors Mr Atish & Ms Shally Pal for their extreme support and guidance without which this work would not have been possible.

9. REFERENCES

[1] IEEE computer society (1990). "IEEE standard glossary of software engineering terminology".

[2] Pressman roger(2010) software engineering : a practitioner's Approach Boston: Mc Graw Hill:P 123 ISBN 9780073375977.

[3] Nuseibeh.B; Easter brook,S(2000), "Requirements engineering :a roadmap"ICSE'00. Proceedings of the conference on the future of software engineering.

[4] Kotonya, Gerald ; Sommerville Ian(September 1998).Requirements engineering :process and techniques.john wiley & sons,ISBN0-97197208-8

[5] F.Brook,"No silver Bullet-essence and accidents of software engineering",IEEE computer, vol-20,no.4, 1987, PP 10-19

[6] Agrawal, Rakesh; and Srikant, Ramakrishnan;Fast algorithms for mining association rules in large databases, in Bocca, Jorge B.; Jarke, Matthias; and Zaniolo, Carlo; editors, Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, September 1994, pages 487-499.

[7] Zaki, M. J. (2000), "Scalable algorithms for association mining". IEEE Transactions on Knowledge and Data Engineering **12** (3): 372–390.doi:10.1109/69.846291.

[8] Kathryn l.heninger , published in IEEE transaction, vol SE-6

[9] Du Zhang, Tools with artificial intelligence, 2002(ICTAI) Proceedings 14th IEEE

[10] Alrajeh D,Kramer J, Russoa ,et al Elaborating requirements using modeL checking and inductive learning ,2013,(IEEE transactions on software engineering).

[11] Kiri L. Wagstaff(California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109 USA)