Machine Learning Applications in Software Engineering: Recent Advances and Future Research Directions

Subhendu Kumar Pani
Computer Science & Engineering
Odisha Engineering College
Bhubaneswar, India

Anil Kumar Mishra
Computer Science & Engineering
Einstein Academy of Technology & Management
Bhubaneswar, India

***Abstract*-Machine learning is the analysis of building computer programs that develop their performance through experience. To assemble the challenge of developing and managing large and complex software systems in a dynamic and changing environment, machine learning techniques have been playing a progressively more important role in much software development and maintenance tasks. Machine learning techniques have proven to be of huge practical value in a diversity of application domains. Not amazingly, the field of software engineering emerging to be a fertile area where many software development and maintenance tasks could be invented as learning problems and approached in terms of learning algorithms. The history of two decades has witnessed a rising interest, and some heartening results and publications in machine learning application to software engineering. As a consequence, a crosscutting niche area emerges. Presently, there are some efforts to raise the awareness and profile of this crosscutting, emerging area, and to systematically study various issues in it. Some of the latest advances in this emerging niche area is presented in this paper.**

Keywords—Machine learning, Software engineering, analytical learning, supervised learning.

I. INTRODUCTION

Machine learning methods fall into the following broad categories: supervised learning, unsupervised learning, semi-supervised learning, analytical learning, and reinforcement learning. Supervised learning deals with learning a target function from labeled examples. Unsupervised learning attempts to learn patterns and associations from a set of objects that do not have attached class labels. Semi-supervised learning is learning from a combination of labeled and unlabeled examples. Analytical learning relies on domain theory or background knowledge, instead of labeled examples, to learn a target function. Reinforcement learning is concerned with learning a control policy through reinforcement from an environment

Data and information have become major assets for most of the organizations [1]. The success of any organization depends largely on the extent to which the data acquired from business operations is utilized. In other words, the data serves as an input into a strategic decision making process, which could put the business ahead of its competitors. Also, in this era, where businesses are driven by the customers,

Having a customer database would enable management in any organization to determine customer behavior and preference in order to offer better services and to prevent losing them resulting better business [2,3]. Data mining is considered to be an emerging technology that has made revolutionary change in the information world. The term `data mining'(often called as knowledge discovery) refers to The process of analyzing data from different perspectives and summarizing it into useful information by means of a number of analytical tools and techniques, which in turn may be useful to increase the performance of a system. Technically, ―data mining is the process of finding correlations or patterns [4, 5].Among dozens of fields in large relational databases‖. Therefore, data mining consists of major functional elements that transform data onto data warehouse, manage data in a multidimensional database, facilitates data access to information professionals or analysts, analyze data using application tools and techniques, and meaningfully present data to provide useful information.

II. TECHNIQUES AND ALGORITHMS

Researchers find two important goals of data mining: prediction and description. First, the Prediction is possible by use of existing variables in the database in order to predict unknown or future values of interest. Second the description mainly focuses on finding patterns describing the data the subsequent presentation for user interpretation. The relative emphasis of both prediction and description differs with respect to the underlying application and techniques.

A. CLASSIFICATION

Classification is the most commonly applied data mining technique, which employs a set of pre-classified examples to develop a model that can classify the population Of records at large. Fraud detection and credit risk applications are particularly well suited to this type of analysis. This approach frequently employs decision tree or neural network-based classification algorithms. The data classification process involves learning and classification . In Learning the training data are analyzed by classification algorithm. In classification test data are used to estimate the accuracy of the classification rules. If the accuracy is acceptable the rules can be applied to the new data tuples. For a fraud detection application, this would include complete records of both fraudulent and valid activities

determined on a record-by-record basis. The classifier-training algorithm uses these preclassified examples to determine the set of parameters required for proper discrimination. The algorithm then encodes these parameters into a model called a classifier. Some well-known classification models are:

a) Classification by decision tree induction
b) Bayesian Classification
c) Neural Networks
d) Support Vector Machines (SVM)

B. DECISION TREE LEARNING

A target function is defined as a decision tree. Search in decision tree learning is often guided by an entropy-based information gain measure that indicates how much information a test on an attribute yields. Learning algorithms often have a bias for small trees. It is an eager, supervised, and unstable learning method, and is susceptible to noisy data, a cause for over fitting. It cannot accommodate prior knowledge during the learning process. However, it scales up well with large data in several different ways[6].

C. NEURAL NETWORK LEARNING

A fixed network structure, learning a target function amounts to finding weights for the network such that the network outputs are the same as (or within an acceptable range of) the expected outcomes as specified in the training data. A vector of weights in essence defines a target function. This makes the target function very difficult for human to read and interpret. This is an eager, supervised, and unstable learning approach and cannot accommodate prior knowledge. A popular algorithm for feed-forward networks is back propagation, which adopts a gradient descent search and sanctions an inductive bias of smooth interpolation between data points [7].

D. BAYESIAN LEARNING

It offers a probabilistic approach to inference, which is based on the assumption that the quantities of interest are dictated by probability distributions, and that optimal decisions or classifications can be reached by reasoning about these probabilities along with observed data. Bayesian learning methods can be divided into two groups based on the outcome of the learner: the ones that produce the most probable hypothesis given the training data, and the ones that produce the most probable classification of a new instance given the training data. A target function is thus explicitly represented in the first group, but implicitly defined in the second group. One of the main advantages is that it accommodates prior knowledge (in the form of Bayesian belief networks, prior probabilities for candidate hypotheses, or a probability distribution over observed data for a possible hypothesis). The classification of an unseen case is obtained through combined predictions of multiple hypotheses. It also scales up well with large data. It is an eager and supervised learning method and does not require search during learning process. Though it has no problem with noisy data, Bayesian learning has difficulty with small data sets. Bayesian learning adopts a bias that is based on the minimum description length principle.

E. CONCEPT LEARNING

A target function is represented as a conjunction of constraints on attributes. The hypothesis space H consists of a lattice of possible conjunctions of attribute constraints for a given problem domain. A least-commitment search strategy is adopted to eliminate hypotheses in H that are not consistent with the training set D. This will result in a structure called the version space, the subset of hypotheses that are consistent with the training data. The algorithm, called the candidate elimination, utilizes the generalization and specialization operations to produce the version space with regard to H and D. It relies on a language (or restriction) bias that states that the target function is contained in H. This is an eager and supervised learning method. It is not robust to noise in data and does not have support for prior knowledge accommodation[8].

F. CLUSTERING

Clustering is a technique for identification of similar classes of objects. By using clustering techniques we can further identify dense and sparse regions in object space and can discover overall distribution pattern and correlations among data attributes. Classification approach can also be used for effective means of
distinguishing groups or classes of object but it becomes costly so clustering can be used as preprocessing approach for attribute subset selection and classification. For example, to form group of customers based on purchasing patterns, to categories genes with similar functionality. Some commonly used clustering
Methods are: Partitioning Methods

a) Hierarchical Agglomerative (divisive) methods
b) Density based methods
c) Grid-based methods
d) Model-based methods

G. ASSOCIATION RULES

An Association Rule is a rule of the form milk and bread =>butter, where 'milk and bread' is called the rule body and butter the head of the rule. It associates the rule body with its head. In context of retail sales data, our example expresses the fact that people who are buying milk and bread are likely to buy butter too. This association rule makes no assertion about people who are not buying milk or bread. We now

Define an association rule: Let D be a database consisting of one table over n attributes {a1, a2,. . . , an}. Let this table contain k instances. The attributes values of each ai are nominal1. In many real world applications (such as the retail sales data) the attribute values are even binary (presence or absence of one item in a particular market basket). In the following an attribute-value-pair will be called an item.

An item set is a set of distinct attribute-value pairs. Let d be a database record. d satisfies an item set X _ {a1, a2, . . . , an} if X _ d.An association rule is an implication X) Y where X,Y _ {a1, a2, . . . , an}, Y 6= ; and X \ Y = ;.The

support s(X) of an item set X is the number of database records d which satisfy X.Therefore the support s(X) Y) of an association rule is the number of database records that satisfy both the rule body X and the rule head Y . Note that we define the support as the number of database records satisfying X \ Y, in many papers the support is defined as s (X\Y) k. They refer to our definition of support as support count. The confidence ˆc(X) Y of an association rule X) Y is the fraction ˆc(X) Y) = s(X\Y) s(X) .From a logical point of view the body X is a conjunction of distinct attribute-value-pair and the head Y is a disjunction of attribute value-pairs where X \ Y = ;.Coming back to the example a possible association rule with high support and high confidence would be i1) i2 whereas the rule i1) i3 would have a much lower support value.

H. CLASS ASSOCIATION RULES

The use of association rules for classification is restricted to problems where the instances can only belong to a discrete number of classes. The reason is that association rule mining is only possible for nominal attributes. However, association rules in their general form cannot be used directly. We have to restrict their definition. The head Y of an arbitrary association rule X) Y is a disjunction of items. Every item which is not present in the rule body may occur in the head of the rule. When we want to use rules for classification, we are interested in rules that are capable of assigning a class membership. Therefore we restrict the head Y of a class association rule X) Y to one item. The attribute of this attribute-value-pair has to be the class attribute. According to this, a class association rule is of the form X) ai where ai is the class attribute and X _ {a1, . . . , ai−1, ai+1, .. . , an}.The Apriori algorithm has become the standard approach to mine association rules. We have adapted it to mine class association rules in the way explained by Liu et al. .The second algorithm, Predictive Apriori, has been recently proposed by Schaffer .Both algorithms have their first step in common. They generate frequent item sets in the same way. An item set is called frequent when its support is above a predefined minimum support.

III. MACHINE LEARNING APPLICATIONS IN SOFTWARE ENGINEERING

In software engineering, there are three categories of entities: processes, products and resources. Processes are collections of software related activities, such as constructing specification, detailed design, or testing. Products refer to artifacts, deliverables, documents that result from a process activity, such as a specification document, a design document, or a segment of code. Resources are entities required by a process activity, such as personnel, software tools, or hardware. The aforementioned entities have internal and external attributes. Internal attributes describe an entity itself, whereas external attributes characterize the behavior of an entity (how the entity relates to its environment). Machine learning methods have been utilized to develop better software products, to be part of software products, and to make software development process more efficient and effective[9,10]. The following is

A partial list of software engineering areas where machine learning applications have found their way into:

- Predicting or estimating measurements for either internal or external attributes of processes, products, or resources. These include: software quality, software size, software development cost, project or software effort, maintenance task effort, software resource, correction cost, software reliability, software defect, reusability, software release timing, productivity, execution times, and testability of program modules.
- Discovering either internal or external properties of processes, products, or resources. These include: loop invariants, objects in programs, boundary of normal operations, equivalent mutants, process models, and aspects in aspect-oriented programming.
- Transforming products to accomplish some desirable or improved external attributes. These include: transforming serial programs to parallel ones, improving software modularity, and Mapping OO applications to heterogeneous distributed environments.
- Synthesizing or generating various products. These include: test data, test resource, project management rules, software agents, design repair knowledge, design schemas, data structures, programs/scripts, project management schedule, and information graphics.
- Reusing products or processes. These include: similarity computing, active browsing, cost of rework, knowledge representation, locating and adopting software to specifications, generalizing program abstractions, and clustering of components.
- Enhancing processes. These include: deriving specifications of system goals and requirements, extracting specifications from software, acquiring knowledge for specification refinement and augmentation, and acquiring and maintaining specification consistent with scenarios.
- Managing products. These include: collecting and managing software development knowledge, and maintaining software process knowledge.

IV. CONCLUSION

The main contribution of this review is to discuss the various Machine-Learning Techniques employed in the field of Software Engineering. The paper also gives relative techniques based on their applications, advantages and limitations. After analysis of all the techniques, we cannot

state as any one technique being the best. Each technique has different application areas and is useful in different domains based on its advantages. Thus, keeping in mind the limitations of each of the techniques and also the prime focus being the improvement in performance and efficiency we should use that technique, which best suits a particular application. Besides reviewing the techniques we have also discussed various research areas where machine learning techniques can be used in software engineering.

REFERENCES

[1] Klosgen W and Zytkow J M (eds.), Handbook of data mining and knowledge discovery, OUP, Oxford, 2002.
[2] Provost, F., & Fawcett, T., Robust Classification for Imprecise Environments. Machine Learning, Vol. 42,No.3, pp.203-231, 2001.
[3] Larose D T, Discovering knowledge in data: an introduction to data mining, John Wiley, New York,2005.
[4] Kantardzic M, Data mining: concepts, models,methods, and algorithms, John Wiley, New Jersey,2003.
[5] Goldschmidt P S, Compliance monitoring for anomaly detection, Patent no. US 6983266 B1, issue date January 3, 2006, Available at:www.freepatentsonline.com/6983266.html
[6] Bace R, Intrusion Detection, Macmillan Technical Publishing, 2000.. Agnar Aamodt, Enric Plaza. "Foundational Issues, Methodological Variations, System approaches." AlCom - Artificial Intelligence Communications, IOS Press Vol. 7: 1, pp. 39-59.
[7] Al Globus. "Towards 100,000 CPU Cycle-Scavenging by Genetic Algorithms." CSC at NASA Ames Research Center, September 2001.
[8] Chris Bozzuto. "Machine Learning: Genetic Programming." February 2002.
[9] Dr. Bonnie Morris, West Virginia University "Case Based Reasoning" AI/ES Update vol. 5 no. 1 Fall 1995.
[10] Eleazar Eskin and Eric Siegel. "Genetic Programming Applied to Othello: Introducing Students to Machine Learning Research" available at http://www.cs.columbia.edu/~evs/papers/sigcsepaper.ps.