ASIAN JOURNAL OF PHARMACEUTICAL AND CLINICAL RESEARCH

INNOVARE ACADEMIC SCIENCES
Knowledge to Innovation

Online - 2455-3891
Print - 0974-2441

Special Issue (April)

Full Proceeding Paper

A SURVEY ON MACHINE LEARNING APPROACH TO MAINFRAME ANALYSIS

PRIYANKA P, DEIVANAI K

School of Computing Science and Engineering, Vellore Institute of Technology Chennai Campus, Chennai, Tamil Nadu, India.
Email: deivanai.kathiresan@vit.ac.in

Received: 03 August 2016, Revised and Accepted: 30 January 2017

ABSTRACT

Mainframe system processing includes a “Batch Cycle” that approximately spans in regular interval on a daily basis. The core part of the cycle completes in the middle of the regular interval with key client deliverables associated with the end times of certain jobs are tracked by service delivery. There are single and multi-client batch streams, a QA stream which includes all clients, and about huge batch jobs per day that execute. Despite a sophisticated job scheduling software and automated system workload management, operator intervention is required. The outcome of our proposed work is to bring out the high priority job first. According to our method, the jobs are re-prioritized the schedules so that prioritized jobs can get the available system resources. Furthermore, the characterization, analysis, and visualization of the reasons for a manual change in the schedule are to be considered. This work requires extensive data preprocessing and building machine learning models for the causal relationship between various system variables and the time of manual changes.

Keywords: Mainframe, Data analysis, Commands.

DOI: http://dx.doi.org/10.22159/ajpcr.2017.v10s1.19542

INTRODUCTION

A centralized server is the thing that organizations use to have their business databases, exchange servers, and applications that require a more noteworthy level of security and accessibility than is usually found on littler scale machines. Centralized computer dependably contains around 70% of corporate information from operations (bookkeeping, finance, charging, etc.). Often the “database server” in web-empowered database applications.

Incorporated PCs will be PCs used chiefly by immense relationship for essential applications, conventionally mass data planning. Present day centralized server PCs have capacities less characterized by their single assignment computational speed (tumbles or clock rate) as by their repetitive interior designing and coming about high dependability and security, broad info yield offices, strict in reverse similarity for more seasoned programing, and high use rates to bolster enormous throughput. These machines regularly keep running for a considerable length of time without intrusion, with repairs and even programming and equipment updates occurring amid ordinary operation. For instance, ENIAC stayed in consistent operation from 1947 to 1955. All the more as of late, there are a few IBM centralized server establishments that have conveyed over 10 years of ceaseless business benefit starting 2007, with redesigns not intruding on administration. Centralized servers are characterized by high accessibility, one of the primary purposes behind their lifespan, as they are utilized as a part of uses where downtime would be expensive or disastrous. The term reliability, availability, and serviceability is a characterizing normal for centralized computer PCs.

This kind of approach requires extensive data preprocessing and building machine learning models for the causal relationship between various system variables and the time of manual changes.

PROBLEM STATEMENT

To characterize the state of the system using both business exceptions and system workload artefacts to determine if there are patterns in operator’s response, capturing this processing knowledge and if the type of manual intervention is [1] predictable and [2] can be automated. Develop a real-time decision support application [3] based on learning how the system state changes and is related to. So that reliance on operator experience to meet business goals can be reduced while continuing to maximize the use of available resources.

PROPOSED WORK

First, this phase consists of extracting the relevant data from the text files and storing them into database on which various analyses would be performed. Generalizing CPU health data for the whole data by linearly extending the data based on fixing the intervals. Also by standardizing the entries into one format and store them as tables in a database. From those files, we track the job entry subsystem (JES) commands given by the operator which can be found from the references. Finally, in this phase tables are created based on these commands, and the relevant information is stored. The database contains all the definitions which have made for planning objects. It likewise holds insights of employment and occupation stream execution and in addition the data as the client ID who has made a protest and with that it indicates when the last question was modified. Upon this, we build a machine learning model on top of this so that we can segment the jobs based on the priority and plan it accordingly which has to be executed.

METHOD OF APPROACH

Mainframe approach

Centralized server preparing incorporates a “group cycle” that roughly traverses 8 PM to 8 AM, every week from Monday night to Saturday morning. The center part of the cycle finishes around 2 AM with key customer deliverables connected with the final days of specific employments, followed by administration conveyance. There are single and multi-customer clump streams, a QA stream which incorporates all customers and around 200,000 bunch occupations for every day that executes. In addition to the daytime business transactions, there is also client and securities information vendor files receive as input into the streams [4]. There is a relative job priority classification scheme. There is a job scheduling application to manage the submission of jobs, based on time and other job or file input delivery dependencies [5].

Job scheduling

During the batch cycle, the mainframe system runs at or near 100% of capacity. Despite a sophisticated job scheduling software and automated system workload management, operation intervention is required or believed to re-prioritize when and what jobs get available system resources to ensure tracked deliverables and business expectations for the night are met. Operator experience with workloads and nightly business expectation variables results in changes to how the schedule executes. These changes are captured by manual operator interventions. Ideally, the job schedule should execute on a daily basis without operator intervention, leaving the maximization of the system resources to the system workload manager, with all deliverables being met according to their relative priority for the business as defined in the job schedule. If that equilibrium condition is not being met, the signature of their departure from equilibrium is certain operator commands captured in the system log and certain types of messages in the scheduler log, signifying a state change.

Tivoli workload scheduler (TWS [OPCA])

TWS1 is a fully automated batch job scheduling system that improves job throughout and greatly reduces operations. TWS helps you arrange and sort out each period of cluster employment execution. Amid the handling day TWS generation control programs deal with the creation control programs deal with the creation environment and computerize most administrator exercises it readies your employments for execution, resolves interdependencies and dispatches and tracks every occupation. Since your occupations start when their conditions are satisfied, idle time is minimized and throughput enhances essentially. Employments never come up short on arrangement and if a vocation falls flat, TWS handler the recuperation procedure with almost no administrator intercession.

JES2

MAVENS or Z/OC which is the working framework for IBM centralized servers [6] utilizes a vocation passage subsystem (JES) to get job1 into the working framework, plan occupations for preparing by MVS and control work yield processing. JES2 is plunged from Houston programmed spooling need (HASP) which is characterized as a PC program that gives supplementary employment administration capacities, for example, Scheduling, control of employment stream and spooling. JES2 is a utilitarian expansion of the HASP program that gets occupations into the framework and process all yield information created by the occupation. JES2 is the part of MVS that gives the essential capacity to land positions into and yield out of the MS framework [7]. It is intended to give effective spooling, planning and administration offices for the MVS working framework.

MCP commands are for the scheduler (TWS/OPCA), and JES commands are for OS (IBM Z/OS). MCP commands can get the jobs into the queue including changing its priority and service class. You can even remove the job from the queue using MCP. However, one initiator pick up jobs from the queue using MCP, if the initiator pick up a job then MCP commands cannot reach them only JES commands can reach.

ALGORITHM

Decision tree

Decision tree [8] can be developed generally quick contrasted with different techniques for characterization explanations can be built from tree that can be utilized to get to databases effectively. Decision tree classifiers acquire comparative or better precision when contrasted and other grouping strategies [9] (Fig. 1).

Various information mining methods have as of now been done on instructive information mining to enhance the execution of understudies such as regression, genetic calculation, Bays order, k-implies grouping, relate rules, expectation, and so on. Information mining methods can be utilized as a part of instructive field to improve our comprehension of learning procedure to concentrate on recognizing, extricating and assessing factors identified with the learning procedure of understudies [10]. Grouping is a standout among the most as often as possible. The C4.5, ID3, Classification and Regression Trees (CART) decision tree are connected on the information of understudies to foresee their execution.

CART algorithm

CART is defined as Classification and Regression Trees [11]. The order tree development via CART depends on paired part of the traits. CART additionally in light of Hunt's calculation and can be actualized serially. Gini list is utilized as part measure as a part of selecting the part quality. CART is unique in relation to other Hunt's based calculation since it is additionally used for relapse examination with the assistance of the relapse trees. The relapse investigation highlight is utilized as a part of estimating a needy variable given an arrangement of indicator factors over a given timeframe. CARTs bolsters constant and ostensible property information and have normal speed of handling.

K-nearest neighbor algorithm

Assume that a question is inspected with an arrangement of various characteristics, yet the gathering to which the protest has a place is obscure. Expecting its gathering can be resolved from its qualities; diverse calculations can be utilized to mechanize the grouping procedure [12]. A nearest neighbor classifier is a framework for describing segments in perspective of the course of action of the segments in the arrangement set that are most similar to the experiment. Using this kind of technique we can get the closest nearing neighbors (Fig. 2).

Naive Bayes

If the inputs are independent, we will be using Naive Bayes technique [13] to solve the problem. Given a game plan of things, each of which has a place with a known class, and each of which has a known

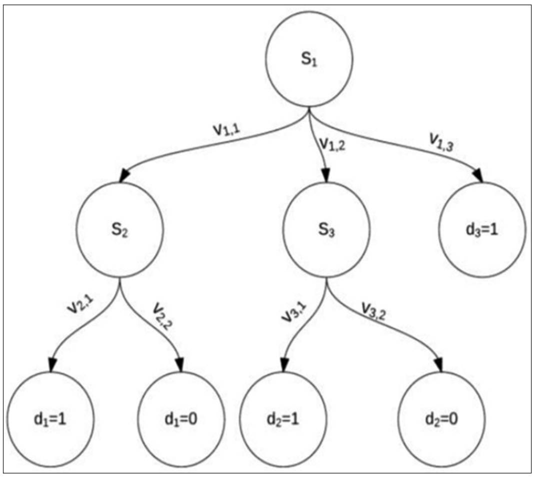


Fig. 1: Decision tree

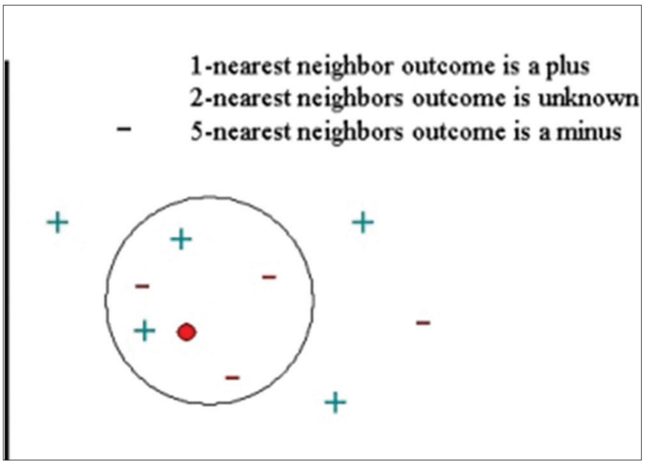
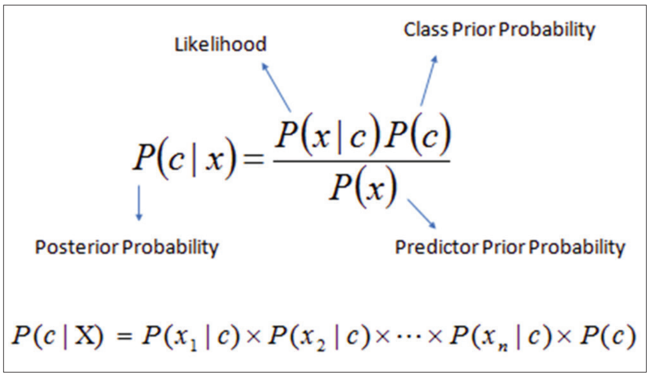

1-nearest neighbor outcome is a plus
2-nearest neighbors outcome is unknown
5-nearest neighbors outcome is a minus

Fig. 2: K-nearest neighbor

vector of components, our indicate is build up a lead which will allow us to dole out future articles to class, given only the vectors of elements portraying the future things. Issues of this kind, called issues of direct request, are ubiquitous, and various techniques for building such standards have been delivered. One crucial one is the guileless Bayes procedure — in like manner called Nitwit's Bayes, essential Bayes, and self-rule Bayes. This method is basic for a couple reasons. It is definitely not hard to assemble, not requiring any convoluted iterative parameter estimation arranges. This infers it may be expeditiously associated with gigantic data sets. It is definitely not hard to interpret, so customers clumsy in classifier development can grasp why it is making the portrayal it makes (Fig. 3).



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Fig. 3: Naive Bayes

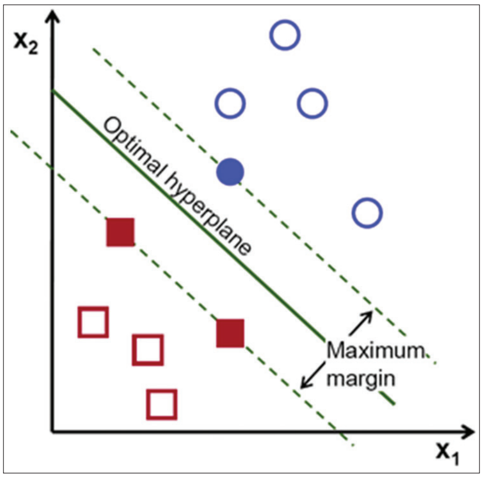
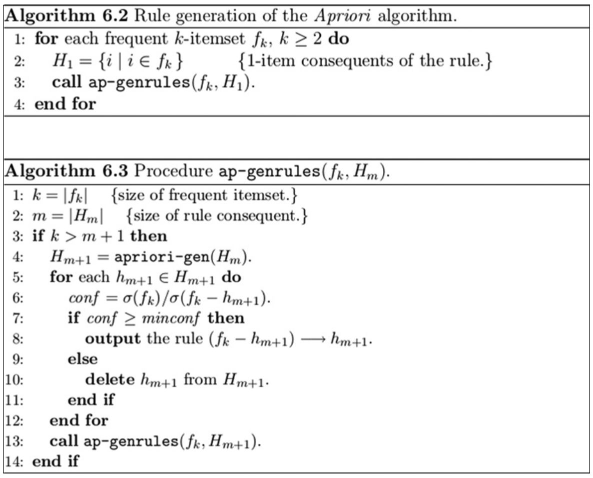


Fig. 4: Support vector machines



```
Algorithm 6.2 Rule generation of the Apriori algorithm.
for each frequent k-itemset $f_k$, $k \geq 2$ do
  $H_1 = \{i \mid i \in f_k\}$        {1-item consequents of the rule.}
  call ap-genrules($f_k$, $H_1$).
end for

Algorithm 6.3 Procedure ap-genrules($f_k$, $H_m$).
$k = |f_k|$    {size of frequent itemset.}
$m = |H_m|$    {size of rule consequent.}
if $k > m + 1$ then
  $H_{m+1}$ = apriori-gen($H_m$).
  for each $h_{m+1} \in H_{m+1}$ do
    $conf = \sigma(f_k)/\sigma(f_k - h_{m+1})$.
    if $conf \geq minconf$ then
      output the rule $(f_k - h_{m+1}) \longrightarrow h_{m+1}$.
    else
      delete $h_{m+1}$ from $H_{m+1}$.
    end if
  end for
  call ap-genrules($f_k$, $H_{m+1}$).
end if
```

Fig. 5: Apriori algorithm

Support vector machines (SVM)

In SVM the data in plotted in the n-dimensional space [14]. After plotting the data in the n-dimensional space, the data are splitted separating the different classes which involve method of supervised learning classification in the n-dimensional space. Based on that, we will be drawing a line which is called as hyperplane since we are drawing the line in the n-dimensional space [15]. After drawing the hyperplane, we will see the classes which are having the highest margin. The classes which are best suited that is nearer to the hyperplane that is having more distance between nearest data point and the hyperplane (Fig. 4).

The apriori algorithm

In apriori algorithm, if we consider n itemset then n set of rules are generated. Based on that among the n rules, we need to find the rule which is having more support and confidence [16,17]. For this, we will using a best aglorithm called Apirori algorithm. First, we need to generate frequent itemsets which are having more support, and we need to change the rules with having more confidence based on the splitting of items (Fig. 5).

LITERATURE SURVEY

Machine learning gives capacity for projects to learn without being expressly customized for a specific dataset. Edmondson's insight is that ML is part of a software engineering thread known as model-driven engineering. ML introduces a new category of model-building activities that can transform the software development life cycle. ML is coming to a mainframe near you but it may be cloaked in predictive analytics. Last year Zementis, whose products leverage the predictive model markup language (PMML), announced availability for z/OS. Zementis models can be used to embed predictive models in z/OS CICS or Web Sphere settings. The models are "write-once," meaning they can be deployed to z/OS SPSS, R, Python, or SAS. In a post on IBM Developer Works, Ravi Kumar outlines how z/OS users can now enable ML on OLTP applications, such as by embedding predictive models in DB2. One technique embeds the z/OS SPSS Scoring Adapter for DB2. Another approach combines a PMML model with business rules to make real-time decisions in DB2 or use Zementis-generated PMML to inject in-app scoring for CICS or Java apps. The IBM DB2 Analytics Accelerator for z/OS supports several major predictive analytics algorithms: K-means, Naive Bayes, decision tree, regression tree, and two-step.

CONCLUSION

By identifying the high priority jobs which are having higher wait time, making them to allocate first so that the higher priority jobs will get executed first. Therefore, we can characterize analyze and visualize the reasons for a manual change in the schedule.

REFERENCES

1. Dumitru D. Prediction of recurrent events in breast cancer using the Naive Bayesian classification. Ann Univ Craiova Math Comput Sci 2009;36(2):92-6.
2. Wu X, Kumar V, Quinlan JS, Ghosh J, Yang Q, Motoda H, *et al*. Top 10 algorithms in data mining. Knowl Inf Syst 2008;14:1-37.
3. Ubeyli ED. Comparison of different classification algorithms in clinical decision making. Expert Syst 2007;24(1):17-31.
4. The IBM Archives, Which Contain a Wealth of History of the Mainframe. Available from: http://www.ibm.com/ibm/history/exhibits/mainframe/mainframe_intro.html.
5. Pugh EW, Johnson LR, Palmer JH. The definitive history of the development of the system/360. IBM's 360 and Early 370 Systems. Cambridge: MIT Press; 1991.
6. The IBM Publication Web Site for Z/OS. Available from: http://www.ibm.com/servers/eserver/zseries/zos/bkserv.
7. JES2 Commands-Version 2, Release1 of z/OS (5650-ZOS). IBM Corporation; 1997, 2013.
8. Delen D, Walker G, Kadam A. Predicting breast cancer survivability: A comparison of three data mining methods. Artif Intell Med 2005;34(2):113-27.
9. Chen MS, Han J, Yu PS. Data mining: A overview from a database

perspective. IEEE Trans Knowl Data Eng 2002;8(6):866-83.

10. Quinlan JR. Programs for Machine Learning. Amsterdam: Morgan Kaufmann; 1993.
11. Schwarzer G, Vach W, Schumacher M. On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. Stat Med 2000;19(4):541-61.
12. Kaur H, Wasan SK. Empirical study on applications of data mining techniques in healthcare. J Comput Sci 2006;2(2):194-200.
13. Han J, Kamber M. Data Mining Concepts and Techniques. Amsterdam: Elsevier; 1992.
14. Hammerstrom D. Neural networks at work. IEEE Spectr 1993;30(6):26-32.
15. Pujari AK. Data Mining Techniques. Hyderabad: Universities Press (India) Ltd.; 2001.
16. Klosgen W, Zytkow JM, Zyt J, editors. Handbook of Data Mining and Knowledge Discovery. 1st ed. New York: Oxford University Press; 2002.
17. Nurnberger A, Pedrycz W, Kruse R. Neural network approaches. In: Klosgen W, Zytkow JM, editors. Handbook of Data Mining and Knowledge Discovery. New York: Oxford University Press; 1990.