International Journal of Engineering & Technology, 7 (4.19) (2018) 394-396

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Predictive Model for Successful Product Mix in Trade Outlets using Genetic Algorithm and Association Rule Mining

V.V.Ramalingam[1*], Viplav Vijay Jha[2], A.Pandian[3]

[1, 2, 3]Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur
**Corresponding Author Email: [1]ramabi1976@gmail.com*

Abstract

Supermarket chains seem to be have a humongous amount of data. Aprioi algorithm is considered to be the classic way to create associations in the data and create power combination of products that would occupy a particular shelf and area. This algorithm employs the greedy method to create meaningful associations between data. This mechanism, however , has a very high time complexity. Its time complexity is linearly proportional to the product of number of transactions and the average number of products bought per transaction, thus creating computations that can only be solved in polynomial time when compared to the logarithmic-time complexity algorithm which we employ in this paper.

Keywords: *Association-mining, Apriori, Binary-Encoding, Genetic-Algorithms*

1. Introduction

The project aims to mine the billing and retail data of a big supermarket chain using Association rule mining. Every potential association will be given a relative fitness which would then be compared with the fitness of the all the other associations. The more powerful association will be selected for reproduction creating the next generation of associations. The businesses and supermarket that have very wide range of products, with frequent new introduction of new products as well as new groups of customers, find it cumbersome to spend so much processing power in order to mine their humongous transactional data. Furthermore ,the consistent introduction of new customers reduces the important of exact precision, and instead demands for an approach, that is both quicker and little more uncertain(to better the chances with new customer bases) . Genetic algorithm would be a powerful tool to solve this problem, because the nature of the optimization demanded in this case has a global nature. The time complexity of the whole computation would drastically be reduced by genetic algorithm, and it would also maintain a level of favorable chaos in its outcome. This would allow the business to make frequent computations, thus turning the market into a more dynamically attractive system for the customers. The associations gained as the output of these computation tends to align well with psychology of the customer.

2. Association Mining

An association can be defined as a form of implication i.e the presence of one factor implies the presence of another. For the purpose of our computation, we need to verify and validate all such implicative relations between the various product present in a trade outlet. The aim is to verify the associations that are comparatively more frequent throughout the given database.

2.1 Set Construction

Associations can be easily extended by identifying all the bridge building associations between two different set of associations i.e if two or more different association sets are made to go an intersection operation, and the result is not null, then the union of both sets would be a new association set with a cardinality that is atleast one more than the biggest of the two or more association set that participated in the union operation. This could be done greedily by dataset pruning, but such an operation will have linear time complexity, whereas a more heuristic approach will get us closer towards a logarithmic complexity.
It should be kept in mind that any given frequent set which is maximal in nature ,i.e that addition of any more element to it will make the new association set useless for our computations, can be broken in any manner to get a subset which will always be a successful association.
It is also possible to come across border sets, sets which themselves aren't a successful piece of association ,but when decomposed in whatsoever manner, the resultant set is always a successful association. The Apriori algorithm generates and evaluates candidate set by greedily generating maximal frequent sets from the border set. Each transaction is refined one by one in order to get a border set. The same functionality can also be performed using a FP tree. the transactions once traversed, are sorted in a format which starts with the bit that signals the presence of the most bought product(the most favoured product) and ends with one which was bought the least number of times(the least favoured product).Each transaction in sorted in this format in order reduce the processing power that would have been spent for the construction of the FP-tree.

2.2 FP-Tree

The FP-tree always has null as it root node. the root can have any number of children but the upward bound is total number of

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

products in the product-mix. The FP-tree , unlike binary trees, allows duplication of nodes, because each path from root to any one of the leaf nodes represents a transaction(atleast one customer's bill) .When customers choose a product(node) that is on one of the higher levels of a particular traversal path(from root to a leaf) as their first product and ignore its parents in their transaction, the tree respond by creating a duplicate of that particular node , which is then inserted as a new child of the root(null) node. Every node has an integral counter associated with it. This counter is initially set at zero for the Node class. A particular instance , when first created increments the value of this counter by one, and this procedure is repeated every time a customer chooses to traverse a path that has already been established , thereby assigning the respective counters the required frequencies that could further be compared with each other in order to deduce the paths that been traverses by more number of customer(support constant :a value which is premptively determined to generate a specific minimum profit) .The most frequent paths are the successful associations that were supposed to be mined from the transactional input. The counter associated with each node present in one of the paths can further be used to determine all the associations present in the transactional input and their relative success with the customer base.

3. Genetic Algorithms

3.1 Binary Encoding

A form of presenting information using binary bits. The binary bits are used here for encoding transactions. Each product is assigned a definite location ,where its presence is signalled using a 1(live bit) and its absence is signalled using a Zero(the dead bit)All the transactions need to traversed in order to make a survey of relative preference towards particular products by using integral counters.

3.2 Selection

Selection deals with the probabilistic survival of the fittest, in that, more fit chromosomes are chosen to survive. Where fitness is a comparable measure of how well a chromosome solves the problem at hand.

3.3 Crossover

This operation is performed by selecting a random gene along the length of the chromosomes and swapping all the genes after that point.

3.4 Mutation

Alters the new solutions so as to add positive entropy to the computation in the search for better solutions. This is the chance that a bit within a chromosome will be flipped (0 becomes 1, 1 becomes 0)

4. Implementation and Discussion

GAs, as the name suggest follow the evolutionary model for generation and development of a solution. It is like a multiple layered mechanism, where each successive layer is superior to its ancestor. An initial population of potential solutions is taken. The idea is to generate a powerful solution by breaking and recombining these solution with each other in multiple iterations that dominantly follow the same mathematical structure .These operations are made possible by the having each potential solution, also called chromosome, in the form of an encoded string. Binary encoding is the most preferred mechanism for handling problem related to creation, combination and reshuffling of a given set of elements.

The whole computational process will be analogous to evolution. Every new generation of solutions will be created solely from its ancestors using a more probabilistic approach .Every solution will have its own fitness factor, which is basically a function of number of time a product(or a particular association of products) appears in the transactional database. The chromosomes, whose fitness factor exceeds the computational requirement, will the only ones that are allowed to survive and produce offspring for the next generation. The fitness is calculated by doing certain mathematical operations on the solution string. First, the frequency of each product is taken in a frequency table. These frequencies are divided by the total number of transactions for each case in order to bring frequency where they can be easily compared in relative manner. The fitness of a given binary string(chromosome) is found by taking the average of relative frequencies of all the live bits in a given chromosome .The dead bits are ignored, because our aim is not only to find the bigger associations , but the smaller ones too. This process will be re-iterated for each generation ,returning a new population every time. The size and mathematical nature of these chromosomes in the most recent generation depends heavily on the cutoff ration specified by the user. This cutoff ratio is number which species the percentage of transaction. that a particular association has to be part in order to qualify as fit solution.

Every iteration of fitness evaluation of the population member is followed by a selection procedure. The chromosomes whose fitness exceeds the cutoff ration qualified to participate in the reproduction events of the current iteration (generation). Reproduction, in the context of Genetic Algorithms, refers to concatenation of string in a selective manner, with option shuffling to produce a progeny string which has the same length as that of the parents. The progeny will have a concatenation of partial traits from both its parents. This can be achieved by mechanisms like one point crossover and uniform crossover. Any kind of rotation operation should be avoid, as the problem in our case will loose meaning if they are rotated even by a single position. This is due the fact that each string is a unique transaction, and each allele are boolean in nature indicating the presence of a certain product in the given transaction. Thus any rotation can fundamentally change the knowledge that binary chromosomes convey

A coefficient of mutation also comes into play. This mutation refers to a random flipping of a bit(a given set of bits) in chromosome string, which is lucky enough to undergo mutation. The coefficient allocated for mutation is always a very small number. It could be around a decimal number like 0.001, i.e only one in 1000 solution string that participate in the whole procedure is likely to get mutated. The number is kept low in order to avoid introducing too much entropy into our mathematical system, because such random entropy might make the next generation of solutions absurd. Nevertheless, mutations are very important in Genetic Algorithms in order to avoid getting stuck near a local minima in the solution search space. The mutation ensures that the algorithm is pushed further whenever a strong local minima tries to halt the evolution process. Therefore, mutation is required in order to maintain the speed of evolution(improvement that get incorporated in solutions across multiple generation),and the quality of solutions.

Each iteration is analogical to generation in the evolutionary model. The quality and quantity of the solution reduces with each generation. This algorithm will function well with large input thats need to be processed with a reasonably less amount of time with possible restrictions on the processing power that can be allocated to the computation. The solution might be a little inferior for some cases when compared to a similar computation done by a greedy approach, but it is evident that time and power are key to stay ahead in today's market places and a little amount of approximation might actually produce better results in a longer because it needs less resources, and furthermore, it leaves space for the uncertainty that human psychology often experiences when making choices at the market place. The greedy counterparts don't have the capacity to account for such approximation.

5. Conclusion

Genetic algorithms produce solution that are as powerful as the ones which are found The solution is calculated in logarithmic time making this approach the fast possible way to find association s in the transactional database .The future work will include an comparative analysis of the standard greedy methods of association rule mining against the Genetic Algorithm model.

References

[1] Analysis and implementation of association rule mining, R Karthiya Banu. ; R Ravanan. ; J Gopal., IEEE

[2] Bandit problems and the exploration/exploitation tradeoff, W.G. Macready ; D.H. Wolpert.IEEEE

[3] Mining Frequent Itemsets Using Genetic Algorithm Soumadip Ghosh+ , Sushanta Biswas* , Debasree Sarkar* , Partha Pratim Sarkar

[4] Generalized Association Rule Mining Using Genetic Algorithms Peter P. Wakabi-Waiswa, Venansius Baryamureeba and K. Sarukesi

[5] A Survey of Association Rule Mining Using Genetic Algorithm 1Shruti Aggarwal, 2Babita Rani 1,2SGGSWU, Fatehgarh Sahib, Punjab, India

[6] Optimized association rule mining using genetic algorithm Anandhavalli M.*, Suraj Kumar Sudhanshu, Ayush Kumar and Ghose M.K.

[7] A GENERAL SURVEY ON FREQUENT PATTERN MINING USING GENETIC ALGORITHM K. Poornamala1 and R. Lawrance2

[8] Optimization of Association Rule Mining using Improved Genetic Algorithms *

[9] Agrawal R, Imielinski T and Swami A, "Mining Association Rules between Sets of Items in Large Databases", Proceedings of the ACM SIGMOID International Conference on Management of data

[10] Ghosh S., Biswas S., Sarkar Dand Sarkar P.P., "Mining Frequent Itemsets Using Genetic Algorithm", International

[11] Wilson Soto, Amparo Olaya-Benavides,"A Genetic Algorithm for Discovery of Association Rules",

[12] References [1] Agrawal R., Imielinksi T. and Swami A. (1993) Database mining: a performance perspective, IEEE Transactions on Knowledge and Data Engineering.