

Efficient Frequent Pattern Mining using Auto-Associative Memory Neural Network

ABSTRACT

Aims: Frequent pattern mining is one of the imperative tasks in the data mining. The soft computing techniques such as neural network, fuzzy logic have potential to be used in frequent pattern mining since these powerful tools efficiently model data which is also an essential part of mining. The proposed paper aims to provide efficient mining solution using auto-associative memory neural network to efficiently traverse and reduce the search space, and to reduce the I/O computations. It also aims to keep balance in computational and resource efficiency.

Methodology: This paper proposes an efficient algorithm for mining frequent patterns using auto-associative memory. Auto-associative memory is best suitable artificial neural network (ANN) approach for association rule mining as it stores associations among the patterns. In the proposed system auto-associative memory based on the correlation matrix memory (CMM) is used to find the frequent patterns. The proposed work introduces novel learning and recall algorithms using CMM for mining frequent patterns efficiently. The proposed learning algorithm reduces the search space tremendously for recall mechanism. The proposed recall algorithm uses only frequent 1-patterns and frequent 2-patterns for determining all other frequent patterns, reducing the number of I/O computations and speeding up the mining process. This approach keeps well balance among computational and resource efficiency.

Conclusion: The performance of the proposed system is compared with traditional algorithms like Apriori, Frequent pattern growth (FP-growth), Compressed FP-tree based algorithm (CT-PRO) and Linear time Closed itemset Miner (LCM). The experimental results show order of magnitude improvement in execution time and storage space optimization to accumulate frequent patterns. Proposed work proves milestone approach in the field of frequent pattern mining using artificial neural network.

Keywords: Association Rule Mining, Frequent Pattern Mining, Artificial Neural Network, Auto-associative Memory, Correlation Matrix Memory

1. INTRODUCTION

The data mining techniques can be broadly classified as predictive and descriptive techniques into association rule mining, classification, clustering, and regression and prediction analysis. Data mining is defined as a multi disciplinary research area which integrates various techniques from soft computing field such as machine learning, statistics, ANNs, etc.

Among the descriptive data mining techniques, association rule mining is a two step process. The first step is frequent pattern mining where all frequent patterns are mined using predefined minimum support threshold. In the second step strong association rules are generated from the frequent patterns using minimum confidence threshold. As the second step

25 is less costly in terms of number of computation and is based on the first step, the overall
26 performance of association rule mining is determined mainly by frequent pattern mining.

27 Introduced by Agrawal et al. [1] association rule mining is a technique to form rules defining
28 the associativity among the attributes of the datasets. Similarly, in the ANNs, the concept of
29 associativity defines an explicit relationship between an input and target patterns offered to
30 the ANN. Earlier less research work appeared using ANN for rule mining may be because of
31 its complex architecture and learning algorithms. Later, rule mining researchers recognized
32 ANN as an important technique.

33 The rule mining algorithms have been continuously developed by both ANN and non-neural
34 network researchers. In non-neural network based algorithms the Apriori algorithm [1] is a
35 level-wise algorithm where it first process frequent 1-itemsets then frequent 2-itemsets and
36 so on till maximum frequent n-itemsets. Another characteristic of this algorithm is generate-
37 and-test for finding frequent patterns. It requires multiple database scans equal to maximum
38 length of frequent pattern in worst case. It also requires huge memory space to generate and
39 store all these candidate sets.

40 In neural network based algorithms Craven and Shavilk proposed rule mining as learning for
41 if-then-else rule mining using a trained ANN [2, 3]. One approach to tackle the drawback of
42 not gaining accurate interpretations of the nonlinear systems has been developed using meth-
43 ods of rule extraction [4, 5, 6, 7, 8, 9]. This approach turns the conversion of the real-
44 valued knowledge formed in the weight matrix into symbolic rules.

45 The rule extraction mechanisms already exist for different neural architectures such as multi-
46 layer perceptron [10, 11, 15, 18, 25], recurrent networks [13], self-organising networks [9,
47 14], associative memory networks [19], genetic algorithms [23,28], Fuzzy rules [21], and
48 even for hybrid architectures [12]. They have focused on the formation of rules, in particular
49 for classification tasks. A Hopfield network for rule mining has already been proposed by
50 Gaber et al. [16], generating association rules from an ANN. Employing incremental training
51 for maintaining rules throughout time, is proposed by Eom and Zhang [17]. Multilevel associ-
52 ation rule mining using Multi Level Feed Forward (MLFM) neural network is proposed as a
53 specialized area of rule mining [20, 26]. A novel four phase data mining algorithm using
54 ANN, referred as ESRNN (Extraction of Symbolic Rules from ANNs), is proposed for ex-
55 tracting symbolic rules [25]. The algorithm uses backpropagation learning.

56 ANNs have the capability to interpret meaning from complicated data and hence can be
57 used for extracting patterns and detecting trends in various applications like e-commerce
58 applications [27], credit card fraud detection [22], and biomedicine [24].

59 Other non-neural based algorithms tried to overcome the drawbacks of Apriori algorithm are
60 Frequent pattern growth (FP-growth) [31, 32], Compressed FP-tree based algorithm (CT-
61 PRO) [33] and Linear time Closed itemset Miner (LCM) [30].

62
63 The FP-growth algorithm proposed compact prefix tree based frequent pattern tree structure
64 FP-tree and the FP-growth algorithm to mine conditional trees. The CT-PRO is another pre-
65 fix tree based algorithm which utilizes a more dense data structure, Compressed FP-Tree
66 (CFP-Tree). The algorithm achieves better resource efficiency by reducing number of nodes
67 in a CFP-Tree to a half less than in the corresponding FP-Tree, resulting in memory optimi-
68 zation. In LCM it proposed various algorithms like LCM, LCMfreq and LCMmax, for enume-
69 rating closed, all and maximal frequent itemsets. It tried to solve the problems of dense da-
70 tasetes. The second version of LCMmax includes a pruning method, thus the computation
71 time is reduced when the number of maximal frequent itemsets is small.

The challenge of the generation of association rules from a dataset is substantial, since its complexity, which is mainly defined by the complexity of frequent pattern mining, is exponential in search space. Although many approaches as discussed above like multilayer perceptron, recurrent networks, Hopfield network and genetic algorithms have claimed reducing the complexity of the problem, the real causes that produce a good performance in the algorithms are still uncertain. Hence, it is important to continue research in this area in order to find answers to the current challenges.

The objective of the proposed work is to implement efficient strategies to traverse and reduce the search space, and to reduce the I/O computations. The proposed auto-associative memory based CMM is best suitable neural network approach for association rule mining as it stores associations among patterns and uses CMM as it provides excellent speed and scalability.

After observing the drawbacks of alternative options available for frequent pattern mining using other neural networks and associative memory approach as discussed in literature review the proposed work based on the auto-associative memory using CMM has been carried out. All the type of associative memories uses CMM structure as it deals with binary and non-binary data and has fast training and recall stages as defined by Austin and Stonham in 1987 and Haykin in 1999 and uses hebbian learning rule to train the CMM as it provides excellent speed and scalability advantage as stated by Haykin.

An auto-associative memory using CMM structure is able to learn, store and recall itemset support after it have been trained with data defining associations. Especially, we analyze and decode the training process and the CMM weight matrix of an auto-associative memory to propose itemset support extraction mechanisms through which they are able to recall itemset support when an itemset is presented as stimulus to the trained networks.

The novel learning and recall algorithms are proposed for performing these tasks efficiently. The proposed learning algorithm reduces the search space tremendously for recall mechanism. The proposed recall algorithm uses only frequent 1-patterns and frequent 2-patterns for determining all other frequent patterns, reducing the number of I/O computations and speeding up the mining process. The input key parameters are min_sup and min_conf and output key parameter is mining time.

The remainder of this paper is organized as follows: Section 2 reviews work related to frequent pattern mining using ANN approaches. Section 3 describes preliminaries for auto-associative memory. Section 4 introduces a novel learning and recall algorithm using CMM for mining frequent patterns efficiently. Section 5 shows experimental results and comparison of proposed implementation with Apriori, FP-growth, CT-PRO and LCM algorithms. Finally, Section 6 draws conclusions from the proposed work and discuss about future scope.

2. RELATED WORK

This section represents the problem statement and related work done in the area of frequent pattern mining using ANN.

2.1 Problem Definition

Let any element i from a set $I = \{i_1, i_2, \dots, i_m\}$ be called an *item* and a grouping or formation of items X such that $X \subseteq I$ be called an *itemset*. In particular, an itemset X with $k = |X|$ is called a k -itemset. Let database D be a set of n transactions, representing an environment. Each transaction t_i is defined by a unique identifier id together with an itemset Y , $ing Y \subseteq I$. A t_i is said to support an itemset X if $X \subseteq t_i.Y$.

121 A basic association rule is an implication defined by $X \rightarrow Y$ in which the itemsets $X \wedge Y \subset I$,
122 but, $X \cap Y = \emptyset$.

123 The support $supp(X)$ of an itemset X with respect to D is defined by the fraction of transac-
124 tions supporting it. This can be defined as follows:

$$125 \quad supp(X) = P(X) = \frac{|\{t_i | t_i \in D \wedge t_i.Y \supseteq X\}|}{|D|} = \frac{freq(X)}{n} \quad (1)$$

126 Since the $supp(X)$ defines the occurrence frequency of X in D , it can also be understood as
127 the probability of X , $P(X)$. In the case of a rule defined by $X \rightarrow Y$, its support is given by the
128 support of $X \cup Y$. The strength of a rule in D is defined by its confidence $conf(X \cup Y)$ as
129 follows:

$$130 \quad conf(X \cup Y) = \frac{supp(X \cup Y)}{supp(X)} \quad (2)$$

131 Since it is impractical and not desirable to mine and generate the total space of itemsets and
132 rules, the challenge of rule mining has been to discover only those rules which are interest-
133 ing. The interestingness of an itemset and a rule is determined by evaluating the properties
134 of support and confidence against the thresholds of minimum support threshold, min_sup ξ
135 and minimum confidence threshold, min_conf respectively. For instance, an itemset X is
136 frequent iff $supp(X) \geq min_sup \xi$.

137
138 Most association rule mining algorithms employ a support-confidence framework. This
139 framework exclude the good number of uninteresting rules but the bottleneck of this ap-
140 proach is filtering uninteresting rules especially when mining at low support thresholds or
141 mining for the long patterns. To tackle this weakness, a correlation measure can be used like
142 lift, χ^2 , all_confidence, max_confidence, Kulczynski and cosine measures.

144 2.2 Literature review

145 Frequent Pattern Mining using ANN is less researched area. The techniques like associative
146 memories, feedforward neural network, and competitive neural network like self organizing
147 map are already explored.

148 Dong et al. [18] proposed competitive network based FP-growth (CNFP) method for mining
149 frequent patterns. This method combines competitive neural network with FP-growth. In
150 competitive learning grouping of similar patterns is carried out by the network and
151 represented by a single neuron. Grouping of similar patterns is done automatically based on
152 data correlations and large dataset is divided into sets of similar patterns. After competitive
153 learning, to construct FP-sub-trees neurons in competitive layer are considered as root and
154 FP-sub-trees contain transactions similar to each other. Frequent patterns are mined using
155 FP-sub-tree which in turn reduces the search space, whereas for large datasets competition
156 phase may take more time for generating neurons.

157 The paper [25] proposed a novel four-phase data mining algorithm using ANNs, referred as
158 Extraction of Symbolic Rules from ANNs (ESRNN), for extracting symbolic rules. The algo-
159 rithm uses backpropagation learning. By using heuristic clustering algorithm, the nodes in
160 the hidden layers are discretized. Then symbolic rules are extracted from frequent patterns
161 using extraction algorithm. An important feature of the proposed rule extraction algorithm is
162 its recursive nature.

163 The Vicente Oswaldo [19] proposed algorithms for frequent pattern mining using **auto-**
164 **associative** neural network and the self-organizing map. In this paper the learning and recall
165 algorithms are introduced to mine frequent patterns. The neural-based proposed framework
166 involves the building of artificial memories that are able to learn, store and recall itemset
167 support. In learning process the weight matrix of an itemset support is constructed based on
168 the associations among patterns. In recall phase frequent patterns are mined. It utilizes inhe-
169 rent feature of **auto-associative** memory to store associations among patterns. However, the
170 case of finding frequent patterns for an itemsets larger than two items remains an open prob-
171 lem.

172 Various soft computing tools are proposed using real life applications like Long-term predic-
173 tion of discharges in Manwan Reservoir, analysis of outcomes of construction claims, Swarm
174 Optimization [34, 35, 36, 37, 38, 39]. The proposed work can be extended further for using
175 real life applications in the field of web mining, text mining and bioinformatics.

176 The scope of the research work is to propose a novel and efficient algorithm to overcome the
177 drawbacks of previously proposed approaches and utilize inherent features of **auto-**
178 **associative** memory more efficiently.

179 **3. AUTO-ASSOCIATIVE MEMORY BASICS**

180
181 The concept of an **auto-associative** memory is mainly exploited by neural architectures
182 which imitate the concept of associative memory present in our brain. In general terms, the
183 explicit association among the inputs is the target to be learnt by some networks to form a
184 knowledge which can be used for pattern association or recognition tasks.

185 The frequent pattern association involves associating a new pattern with a stored pattern.
186 The associative memory is considered as a special case of the neural computing approach
187 for pattern recognition. Types of associative memory are the Hetero-Associative memory, an
188 Auto-Associative memory, the Hopfield Network and the Bidirectional Associative Memory
189 (BAM).

190 The Hopfield network for association rule mining has been proposed by Gaber et al. [16] but
191 is used to determine maximal itemsets so the main drawback of this work is that to calculate
192 support for all itemsets derived from the maximal discovered itemsets an extra pass over the
193 training data is needed.

194
195 After observing the drawbacks of alternative options available for frequent pattern mining
196 using associative memory the proposed work based on the auto-associative memory using
197 CMM has been proposed.

198
199 The **auto-associative** memory is the best suited neural network approach for association rule
200 mining as it stores associations among patterns. In the proposed system **auto-associative**
201 memory based on CMM is used to find frequent patterns. Associative memory can be feed-
202 forward or recurrent. This is supervised single-layer network.

203 *Architecture of Auto-Associative Memory*

204 The architecture of an Auto-Associative Memory is shown in the Fig. 1. It is based on the
205 concept of association.

206 The auto-associative memory neural network is first trained to store a set of patterns in the
207 form $X : Y$ where X represents the input vector and Y is the corresponding output associa-

tive vector. The input patterns are represented as unipolar vectors $\in \{0,1\}$. It has the property to remember the associativity expressed between input patterns under the supervised training.

Since it has been stated by Haykin in 1999 that the target y_i pattern will be defined by the corresponding key pattern x_i of each input pair, proposed study could be confined to the study of the suitability of an auto-associative memory for frequent pattern mining since this neural network holds the characteristic of $y_i = x_i \forall x_i$ in D in its inputs.

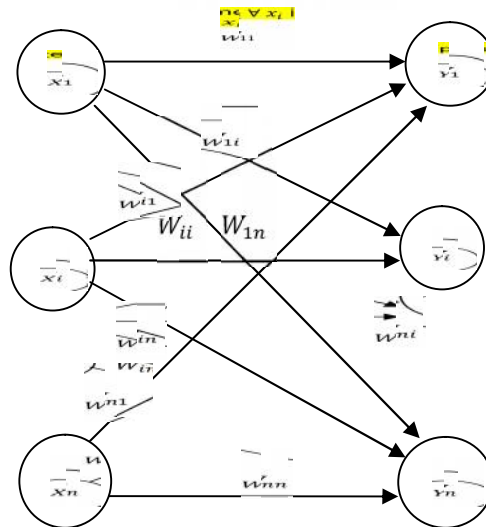


Fig. 1. Auto-Associative Memory

It uses the CMM to store the associations among patterns. The CMM is a single-layer memory structure defined as a square matrix whose dimension is the $m \times m$ elements. The CMM structure is used to store frequencies or pattern occurrences in the form of weights.

The CMM Models are binary weighted and non-binary weightless and has fast training and recall stages. The CMM resulting from the supervised training is represented as $M \in B^{m \times n}$ for binary weighted model. The CMM is trained using a hebbian learning rule which offers excellent speed and scalability advantages. The work utilizes inherent features of CMM and proposes an efficient algorithm for frequent pattern mining.

4. FREQUENT PATTERN MINING USING AUTO-ASSOCIATIVE MEMORY

In order to perform frequent pattern mining with an auto-associative memory using CMM, focus is on how to perform the counting of patterns or itemsets. It is normally produced by looking for the patterns by scanning the high dimensional space defined by the original data environment.

In the work learning algorithm is proposed first using weighted CMM matrix, where it accomplishes the task of training the neural network. The outcome of a learning algorithm is the trained weighted CMM matrix where weights represent the occurrence frequency of associations among patterns. The efficiency for this algorithm is achieved by optimizing the search space as explained in section 4.1.

242 Then the proposed recall algorithm applies a novel approach to recall itemset support from
 243 the trained CMM matrix. The proposed recall algorithm mine frequent patterns efficiently using
 244 weighted CMM matrix, hence reduces the I/O computations as explained in section 4.2.

245 The following sections describe a two step novel frequent pattern mining algorithm using
 246 CMM matrix memory.

247 4.1 Learning for itemset support using CMM

248 The task of itemset support counting is mainly governed by the learning algorithm responsible
 249 for modifying the neural-network architecture (nodes or weight matrix). It accumulates
 250 the knowledge presented in terms of support counts in the environment throughout the time.
 251 Therefore, the learning algorithm is primarily determined by the type of neural network used
 252 to learn the data coming from the environment. In our particular case, the learning algorithm
 253 of an auto-associative memory will be evaluated to fulfill this framework task.

254 The binary weighted CMM matrix M represented as $M \in B^{m \times m}$ is used to store frequencies
 255 or pattern occurrences in the form of weights. Learning is done using the natural ability of the
 256 CMM to learn about occurrence frequency of the patterns based on input patterns.

257 Input to this step is database D represented as a group of n unipolar vectors $\in \{0,1\}$, where
 258 each input is an m -vector defining a particular association among items from the set
 259 $I = \{i_1, i_2, \dots, i_m\}$ where m represents the total number of items and transactions from the set
 260 $T = \{t_1, t_2, \dots, t_n\}$ where n is total number of transactions in database D .

261 Training requires pairs of patterns representing associations with a form $X \rightarrow Y$ in which an
 262 input or key pattern is associated with an output or memorized pattern. Corresponding associations
 263 among patterns may look as, $\{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$, where both the patterns,
 264 the input and the output, will obtain their values from the n transactions in D . In particular,
 265 each pattern of each pair will take the same value defined by a transaction t_k . Hence pair k
 266 satisfies $X_k \rightarrow Y_k = (X_k, Y_k)$ and represented as $X_k = Y_k = t_k \cdot x$. The learning algorithm is
 267 proposed as described below which takes input as pairs of patterns and output is weighted
 268 CMM M .

269 Learning Algorithm:

- 270 • To train M , the equation defined (Haykin, 1999; Ham and Kostanic, 2001) is as in (3).

$$271 \quad \quad \quad 272 \quad \quad \quad M = \sum_{k=1}^m Y_k X_k^T \quad (3)$$

- 273 • The term $Y_k X_k^T$ is an estimation of the weight matrix $W(k)$ of the neural network functioning as a linear associator. This matrix $W(k)$, which associates X_k to Y_k , forms a mapping representing the association described by the k^{th} input pair in turn.

- 277 • The resulting matrix M is grouping of the m weighted matrices W .

- 278 • Individual weight values of the network, whose update resembles a generalization of the Hebbian rule learning, can be expressed by (4).

$$280 \quad \quad \quad 281 \quad \quad \quad w_{ij} = \sum_{k=1}^m y_{ik} x_{jk} \quad (4)$$

- As a consequence of using unipolar elements as inputs, the product $y_i x_j$ of some k^{th} pattern will be in one of the two states as shown in (5).

$$y_{ik}x_{jk} = \begin{cases} 1 & \text{existence of association } ij \text{ in } k \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

The output of symmetric M , from which itemset support values will be estimated, is represented in (6)

$$M = \begin{bmatrix} w_{11} & & & 0 \\ w_{21} & w_{22} & & \\ \vdots & \vdots & \ddots & \\ w_{m1} & w_{m2} & \dots & w_{mm} \end{bmatrix} \quad (6)$$

In the weighted CMM M , w_{ij} not only means the existence of an association between elements i and j , but also the number of times that such association has occurred in the environment. Therefore, we can asseverate that a weighted symmetric CMM naturally builds a frequency matrix which is used in recall algorithm to mine frequent patterns efficiently.

Advantage of learning algorithm is it optimizes space required to store frequent 1-patterns and frequent 2-patterns in weighted CMM matrix. It also reduces search space for recall mechanism to focus on the knowledge of only $n(n+1)/2$ nodes rather than the n^2 elements of the complete matrix.

4.2 RECALL FOR FINDING FREQUENT PATTERNS USING CMM

As a result of learning, pattern frequency is embedded in the knowledge of a neural network; the focus of this section is on giving the right interpretation to the weight matrix in order to recall the frequency (support) of patterns, which can be composed by different items of the learnt environment.

To use the embedded information in triangular M matrix, the right interpretation needs to be drawn to produce accurate support values from it. Therefore, to achieve accurate itemset support recalls from M , when a stimulus is presented, a novel recall algorithm is proposed. The output of recall algorithm is mined frequent 1-itemsets, 2-itemsets, ..., n -itemsets respectively.

Algorithm Recall (weighted CMM M , int min_sup ξ)

- Determine the support of all items $I = \{i_1, i_2, \dots, i_m\}$ stored in M , applying equation (7), where $freq(w_{ii})$ is occurrence frequency of pattern i stored in M at index ii . Support for pattern i is determined as $P(w_{ii})$ which is the probability of the i^{th} item in the n transactions defining the training dataset.

$$supp(i) = P(w_{ii}) = \frac{freq(w_{ii})}{n} \quad (7)$$

If $supp(i) \geq \xi$ then that pattern is stored as frequent 1-itemset or frequent 1-pattern.

- 315 • Determine the frequent 2-itemsets (ij) applying formula as in (8) to all m items
316 stored in M .

317
$$supp(ij) = P(w_{ij}) = \frac{freq(w_{ij})}{n} \text{ for all } i \neq j \quad (8)$$

318 If $supp(ij) \geq \xi$, then pattern (ij) is stored as frequent 2-itemset.

- 319 • Determine support for k -itemsets ($2 > k \leq m$) using following steps-
- 320 a. Only frequent 1-itemsets and frequent 2-itemsets are considered as per an-
321 timonotone property (if subset of any pattern is infrequent then its superset
322 are also infrequent)
- 323 b. Determine frequent k -itemsets applying equation as in (9) to combinations of
324 only frequent 1-itemsets and frequent 2-itemsets stored in M .

325
$$P(X) = \frac{\sum_{i=1}^k \sum_{j=i}^k P(w_{ij})}{n} \quad (9)$$

326 Where, all unique pairs of X_i i.e., $n = {}^k C_2$ and k is length of itemset.

327 Proposed novel recall algorithm is more efficient as it finds k -itemsets ($2 > k \leq m$) using
328 frequent 1-itemsets and frequent 2-itemsets and ignores all infrequent patterns. Another ad-
329 vantage is it reduces I/O cost as there is no need to scan original database various times to
330 find frequent patterns as recall is carried out using knowledge embedded in weighted CMM
331 matrix M .

332 As an example consider database DB as shown in Table 1, with 5 transactions. Let mini-
333 mum absolute support threshold ξ is 3 and relative support threshold on the scale of 10 to
334 100 is 60% and on the scale of 0 to 1 is 0.6.

335 **Table 1. The transaction database**

Transaction ID	Items Input
1001	3, 6, 1, 4, 7, 13, 16, 9
1002	3, 6, 1, 2, 13, 12, 15
1003	6, 2, 8, 10, 15
1004	3, 16, 2, 11, 19
1005	3, 6, 1, 13, 16, 5, 12, 14

336

337 The output M of learning algorithm for example is as shown below where CMM M is of size
338 19×19 , as maximum number of items in itemset are 19. The learning algorithm generates
339 symmetric weighted CMM M . It trains matrix M with frequency count of all 1-itemsets and 2-
340 itemsets in one single database scan with supervised training and speed up the mining
341 process.

342 Showing weighted CMM M

30000000000000000000

13000000000000000000

```

32400000000000000000
10110000000000000000
10101000000000000000
32311400000000000000
10110110000000000000
01000101000000000000
10110110100000000000
01000101010000000000
01100000001000000000
21201200000200000000
31311310100230000000
10101100000111000000
12100201010110200000
21311210101121030000
00000000000000000000
00000000000000000000
0110000000100001001

```

343 The recall algorithm uses weighted CMM M to generate frequent 1-itemsets and frequent 2-
344 itemsets efficiently with the advantage of reduced search space. It discards all infrequent 1-
345 itemsets and 2-itemsets. Taking antimonotone property into consideration all k -itemsets
346 ($2 > k \leq m$) are mined using frequent 1-itemsets and frequent 2-itemsets, hence speed up
347 task of mining. The outputs of Recall algorithm for given example for certain cases are as
348 shown below:

349
350 ➤ Case 1: for $k = 3$, therefore ${}^3C_2=3$ so $X = (X_1, X_2, X_3)$. Then, to find out
351 whether pattern (1 2 3) is frequent or not recall equation is used.
352

$$P(1\ 2\ 3) = \frac{P(w_{2\ 1}) + P(w_{3\ 1}) + P(w_{3\ 2})}{3}$$

353

$$P(1\ 2\ 3) = \frac{\frac{1}{5} + \frac{3}{5} + \frac{2}{5}}{3} = 0.4$$

354 As $P(1\ 2\ 3) = 0.4 < 0.6$ (ξ), this pattern is infrequent 3 - itemset.
355

356 ➤ Case 2: for $k = 4$, therefore ${}^4C_2 = 6$ so $X = (X_1, X_2, X_3, X_4)$. Then, to find
 357 out whether pattern (1 3 6 13) is frequent or not recall equation is used.
 358
 359

$$P(1\ 3\ 6\ 13) = \frac{P(w_{3\ 1}) + P(w_{6\ 1}) + P(w_{13\ 1}) + P(w_{6\ 3}) + P(w_{13\ 3}) + P(w_{13\ 6})}{6}$$

$$P(1\ 3\ 6\ 13) = \frac{\frac{3}{5} + \frac{3}{5} + \frac{3}{5} + \frac{3}{5} + \frac{3}{5} + \frac{3}{5}}{6} = 0.6$$

361
 362 As $P(1\ 3\ 6\ 13) = 0.6 = 0.6 (\xi)$, this pattern is frequent 4 - itemset.
 363

364 The final output of recall algorithm for given example is:

365
 366 Generating candidates: 1 - itemset

367 Possibilities: 19

368 {1}: 3.0 : 0.6
 369 {2}: 3.0 : 0.6
 370 {3}: 4.0 : 0.8
 371 {6}: 4.0 : 0.8
 372 {13}: 3.0 : 0.6
 373 {16}: 3.0 : 0.6

374

375 Generating candidates: 2 - itemset

376 Possibilities: 171

377 {1 3}: 3.0 : 0.6
 378 {1 6}: 3.0 : 0.6
 379 {1 13}: 3.0 : 0.6
 380 {3 6}: 3.0 : 0.6
 381 {3 13}: 3.0 : 0.6
 382 {3 16}: 3.0 : 0.6
 383 {6 13}: 3.0 : 0.6

384

385 Generating candidates: 3 - itemset

386 Possibilities: 969

387 {1 3 6}: 3.0 : 0.6
 388 {1 3 13}: 3.0 : 0.6
 389 {1 6 13}: 3.0 : 0.6
 390 {3 6 13}: 3.0 : 0.6

391

392 Generating candidates: 4 - itemset

393 Possibilities: 3876

394 {1 3 6 13}: 3.0 : 0.6

395

396 Generating candidates: 5 - itemset

397 Possibilities: 11628

398

399 Time required finding frequent candidate is: 0.015 seconds

400

401 If the results are compared for a given example, time required to mine frequent patterns is
 402 less than the standard frequent pattern mining algorithms like Apriori, FP-growth, CT-PRO
 403 and LCM algorithms.

5. EXPERIMENTAL RESULTS

This section will provide the experimental results of the proposed system. The key input parameter of proposed algorithm is minimum support threshold ξ and the output parameters considered for comparison are time required for finding frequent 1-itemsets (time to learn the neural network) and total execution time (time to learn and recall). For lower ξ values mining requires more execution time as it generates large number of frequent patterns and higher the ξ range lesser the mining time.

The proposed work at present is implemented based on support-confidence framework and can be further extended using correlation measure like Kulczynski measure, together with the imbalance ratio, to discover the quality of the patterns.

In order to determine the accuracy of the proposed mechanisms for itemset support recalls from a trained CMM, experiments with standard real datasets are performed in this section. The experiments were performed on an Intel Core2 Duo processor 1.66GHz with 2GB memory, running the Microsoft Windows.

The proposed system is compared with non-neural network based algorithms as it is observed that very few neural network based algorithms compared their work with standard and large datasets. The major achievement of our proposed work is in neural network based algorithms, it proves good mining efficiency on standard datasets where other algorithms failed to prove their work. So this algorithm proves a milestone in the research of frequent pattern mining using neural network. This work is also extended for handling huge datasets using Hadoop MapReduce framework and is not included in the scope of this paper.

The standard datasets used in previous studies are downloaded from Frequent Itemset Mining Implementations Repository (FIMI'04) website [29] and used to compare the algorithms performance with non-neural network based algorithms like Apriori, FP-growth, CT-PRO and LCM algorithms. Experiments are performed on various dense and sparse datasets from FIMI Repository. The experimental results of the proposed work are consistent for all the datasets while compared with non-neural network based algorithms.

Mushroom dataset is a sparse dataset with 8124 transactions, 119 items and average transaction length 23, hence it is relatively sparse. Time required for finding frequent 1-itemsets using mushroom dataset is shown in table 2 for Apriori and proposed frequent pattern mining with an auto-associative memory using CMM algorithms. It proves that frequent 1-itemsets generation time efficiency obtained by CMM in real datasets like mushroom is also notable with respect to various minimum support thresholds.

Table 2. Time required for finding frequent 1-itemsets

Algorithm/ Min. Support	10%	20%	30%	40%	50%	60%	70%	80%	90%
Apriori Time Taken (Sec)	0.422	0.422	0.437	0.453	0.422	0.406	0.438	0.421	0.422
CMM Time Taken (Sec)	0.171	0.156	0.156	0.141	0.109	0.093	0.093	0.093	0.094

Time required for training, recall phase and total mining time for CMM using mushroom dataset is shown in table 3 for minimum support threshold range of 50 to 90%. Proposed learning algorithm reduces search space for recall mechanism and achieves resource efficiency by optimizing search space. Proposed recall algorithm uses only frequent 1-patterns and

frequent 2-patterns for calculating $(2 > k \leq m)$ itemsets, improving the computational efficiency.

Table 3. Time required for mining frequent patterns using CMM for mushroom dataset

Min. Support	50%	55%	60%	65%	70%	75%	80%	85%	90%
Time to train(sec)	0.109	0.094	0.093	0.093	0.093	0.094	0.093	0.11	0.094
Time to recall (Sec)	0.235	0.062	0.032	0.016	0.079	0.015	0.016	0.031	0.047
Total Time (Sec)	0.375	0.203	0.172	0.156	0.172	0.156	0.156	0.141	0.141

447

Table 4 shows comparison for mushroom dataset based on mining time for Apriori, FP-growth, LCM, CT-PRO and CMM algorithms. The results of all algorithms are compared for the minimum support threshold range of 50 to 90%.

Table 4. Comparison of Mining time for mushroom dataset

Algorithm/ Min. Support	50%	55%	60%	65%	70%	75%	80%	85%	90%
Apriori Time (Sec)	1.812	1.406	1.094	1.015	0.953	0.938	0.969	0.797	0.641
CMM Time (Sec)	0.375	0.203	0.172	0.156	0.172	0.156	0.156	0.141	0.141
LCM Time (Sec)	0.5	0.4	0.317	0.252	0.23	0.225	0.225	0.187	0.187
CT-PRO Time (Sec)	0.55	0.42	0.4	0.4	0.38	0.35	0.3	0.25	0.2
FP-growth Time (Sec)	0.7	0.6	0.4	0.4	0.4	0.3	0.3	0.3	0.2

As shown in Fig. 2., the proposed CMM algorithm shows better efficiency as compared to Apriori, FP-growth, CT-PRO and LCM algorithms.

460

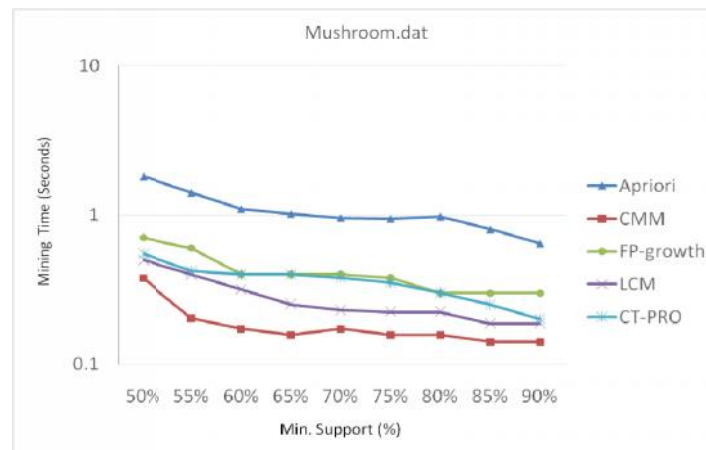


Fig. 2 Mining time for mushroom dataset

For chess dataset proposed CMM algorithm achieves good mining time efficiency. Chess dataset contain 3196 transactions, 75 items and average transaction length is 37, hence it is denser. Although being dense it takes comparatively lesser time for mining than Apriori, FP-growth and CT-PRO algorithms and is slightly comparable with LCM algorithm. Comparative analysis of mining time for chess dataset is as shown in Fig. 3.

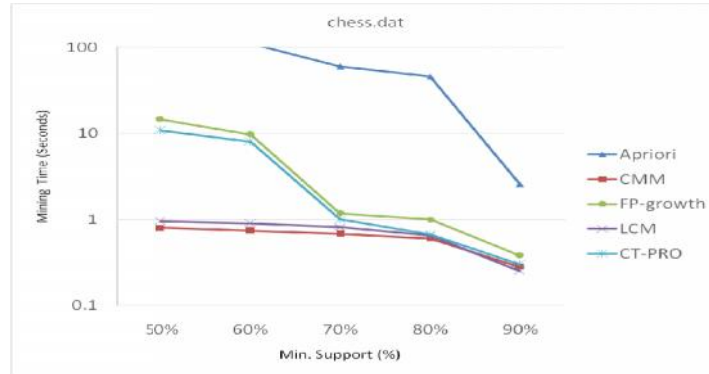


Fig. 3 Mining time for chess dataset

The results obtained using the large dataset, accidents, with respect to the various minimum support thresholds is shown in Fig.4. This dataset is relatively large than the other datasets with 340183 transactions, 468 items and average transaction length 34, hence it is relatively sparse. Results prove the efficiency of mining large datasets as compared to FP-growth and LCM algorithms.

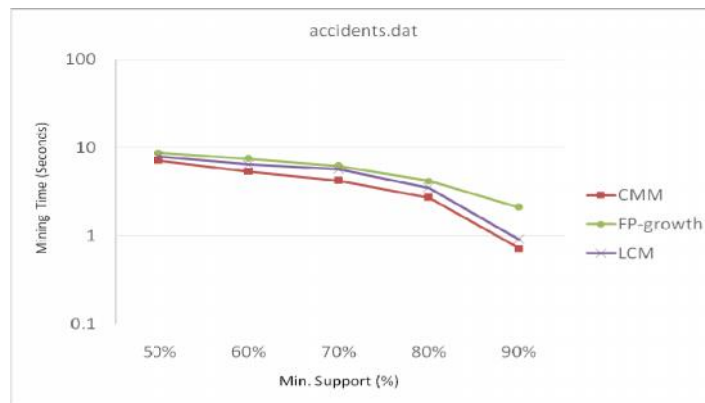


Fig. 4 Mining time for accidents dataset

As shown in Fig. 5, BMS-WebView-2 dataset is tested and compared using CMM, FP-growth and LCM algorithms. The implementation results of CMM shows better computational efficiency as compared to FP-growth and LCM.

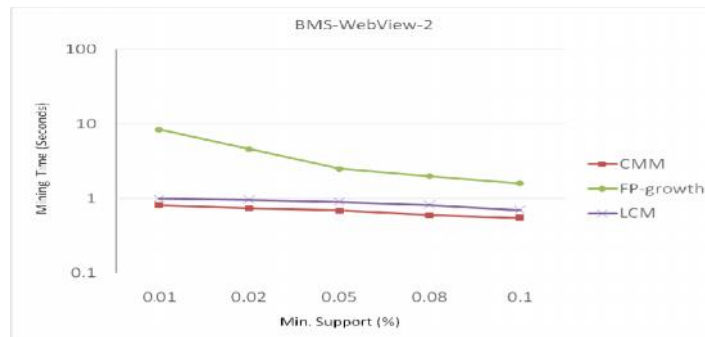


Fig. 5 Mining time for BMS-WebView-2 dataset

530 These experimental results prove efficiency of the proposed CMM algorithm on sparse and
531 dense datasets. The proposed CMM algorithm even work efficiently for low minimum sup-
532 ports. For improving mining efficiency of dense datasets and for mining huge and volumin-
533 ous datasets useful in Big Data analysis, the proposed algorithm is further extended using
534 distributed processing on Hadoop MapReduce environment but is not the scope of this pa-
535 per.

536

537 6. CONCLUSION

538

539 Auto-Associative memory based CMM is best suitable neural network approach for associa-
540 tion rule mining due to its inherent capability to store associations among patterns. Proposed
541 learning algorithm reduces search space for recall mechanism to focus on the knowledge of
542 only $n(n + 1)/2$ nodes rather than the n^2 elements of the complete matrix. Proposed recall
543 algorithm uses only frequent 1-patterns and frequent 2-patterns for calculating ($2 > k \leq m$)
544 itemsets, speeding up mining process.

545

546 This approach keeps well balance among computational and resource efficiency. The major
547 achievement of our proposed work is in neural network based algorithms, it proves good
548 mining efficiency on standard datasets where other neural network based algorithms failed
549 to prove their work. Proposed work proves milestone approach in the field of frequent pat-
550 tern mining using artificial neural network.

551

552 For improving mining efficiency for mining huge and voluminous datasets useful in Big Data
553 analysis, the proposed algorithm is further extended using distributed processing on Hadoop
554 MapReduce environment but is not the scope of this paper. Future scope of the research is
555 in advanced frequent mining like sequential pattern mining, closed and maximal frequent
556 pattern mining and distributed pattern mining.

557

558 REFERENCES

559

- 560 1. Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in
561 large databases. Proc. ACM-SIGMOD international conference on management of data
562 (SIGMOD'93), Washington, DC. 1993;207–216.
- 563 2. Towell GG, Shavlik JW. Extracting refined rules from knowledge-based neural networks.
564 Machine Learning. 1993;13:71–101.
- 565 3. Craven MW, Shavlik JW. Using neural networks for data mining. Future Generation
566 Computer Systems. 1997;13(2–3):211–229.
- 567 4. Benitez JM, Castro JL, Requena I. Are artificial neural networks black boxes? IEEE
568 Transactions on Neural Networks. 1997;8(5):1156– 1164.
- 569 5. Tickle AB, Andrews R, Golea M, Diederich J. The truth will come to light: Directions and
570 challenges in extracting the knowledge embedded within trained artificial neural
571 networks. IEEE-NN. 1998;9(6):1057.
- 572 6. Taha IA, Ghosh J. Symbolic interpretation of artificial neural networks. Knowledge and
573 Data Engineering. 1999;11(3):448–463.
- 574 7. Tsukimoto, H. Extracting rules from trained neural networks. IEEE-NN. 2000;11(2):377.
- 575 8. Browne A, Hudson B. Knowledge extraction from neural networks. Proceedings of the
576 29th Annual Conference of the IEEE Industrial Electronics Society, Roanoke, Virginia,
577 USA. 2003;1909–1913.

- 578 9. Malone J, McGarry K, Wermter S, Bowerman C. Data mining using rule extraction from
579 kohonen self-organising maps. *Neural Computing and Applications*. 2006;15(1):9–17.
- 580 10. Duch W, Adamczak R, Grabczewski K. Extraction of logical rules from training data
581 using backpropagation networks. *First Polish Conference on Theory and Applications of*
582 *Artificial Intelligence*. 1996;210.
- 583 11. McGarry KJ, Wermter S, MacIntyre J. Knowledge extraction from radial basis function
584 networks and multi-layer perceptrons. *IEEE International Conference on Neural*
585 *Networks (IJCNN'99)*, Washington DC. 1999;4:2494–249.
- 586 12. Eggermont J. Rule-extraction and learning in the BP-SOM architecture. **Leiden**
587 **university, Internal Report IR-98-16 (Masters Thesis), August 1998.**
- 588 13. Jacobsson H. Rule Extraction from recurrent neural networks: A taxonomy and review.
589 *Neural Computing*. 2005;17(6):1223–1263.
- 590 14. Hammer B, Rechten, A., Strickert, M, Villmann T. Rule extraction from self-organizing
591 networks. Dorronsoro JR editor, 213 *ICANN, Lecture Notes in Computer Science*. 2002;
592 2415:877– 883.
- 593 15. Setiono R. Extracting M-of-N rules from trained neural networks. *IEEE-NN*. 2000;
594 11(2):512.
- 595 16. Gaber K, Bahi J, El-Ghazawi T. Parallel mining of association rules with a hopfield type
596 neural network. *12th IEEE International Conference on Tools with Artificial Intelligence*
597 *(ICTAI 2000)*, Vancouver, Canada. 2000;90–93.
- 598 17. Eom JH, Zhang BT. Prediction of east protein-protein interactions by neural feature
599 association rule. 2005.
- 600 18. Yihong Dong, Xiaoying Tai, Jieyu Zhao. A Distributed Algorithm Based on Competitive
601 Neural Network for Mining Frequent Patterns. *International Conference on Neural*
602 *Networks and Brain*. 2005;1:499- 503.
- 603 19. Vicente Oswaldo Baez Monroy and Simon O'Keefe. The Identification and Extraction of
604 Itemset Support defined by the Weight Matrix of a Self-Organising Map. *IEEE World*
605 *Congress on Computational Intelligence*. 2006.
- 606 20. Tong-yan Li, Xing-ming Li. Study of the neural network applied to weighted Association
607 Rules Mining. *Proc. International Conference on Wavelet Analysis and Pattern*
608 *Recognition*, Beijing, China. 2-4 Nov. 2007.
- 609 21. Au W, Chan K. Mining Fuzzy Association Rules. **Proc. of the 6th international conference**
610 **on Information and knowledge management, Las Vegas, Nevada, USA. 1997;209-215.**
- 611 22. Baesens B, Setiono R, Mues C, Vanthienen J. Using neural network rule extraction and
612 decision tables for credit risk evaluation. *Management Science*. 2003;49(3):312-329.
- 613 23. Markowska U, Wnuk P. Rule Extraction from Neural Network by Genetic Algorithm with
614 Pareto Optimization. *Springer Lecture Notes in Computer Science*. 2004;3070(1):450-
615 455.
- 616 24. Robardet C, Cremilleux B, Boulicaut J. Characterization of Unsupervised Clusters with
617 the Simplest Association Rules: Application for Child's Meningitis. *Seventh International*
618 *Workshop on Intelligent Data Analysis in Biomedicine and Pharmacology (IDAMAP'02)*,
619 *Lyon, France*. 2002;61-66.
- 620 25. Kamruzzaman SM, Jehad Sarkar AM. A New Data Mining Scheme Using Artificial
621 Neural Networks. *Sensors*. 2011;11:4622-4647, doi:10.3390/s110504622.
- 622 26. Amit Bhagat, Sanjay Sharma, Pardasani KR. Ontological Frequent Patterns Mining by
623 potential use of Neural Network. *International Journal of Computer Applications* (0975-
624 8887). 2011;36(10):44-53.

- 625 27. Pao-Hua Chou, Pi-Hsiang Li, Kuang-Ku Chen, Menq-Jiun Wua. Integrating web mining
626 and neural network for personalized e-commerce automatic service. *Expert Systems*
627 *with Applications*. 2010;37:2898–2910.
- 628 28. Peter Wakabi-Waiswa, Venansius Baryamureeba, K Sarukesi. Generalized Association
629 Rule Mining Using Genetic Algorithms. *Computer Science, Proceedings of Seventh*
630 *International conference in natural computation*, IEEE. 2011;59-69.
- 631 29. Frequent Itemset Mining Implementations Repository (FIMI) Repository. 2004. Accessed
632 on 1 June 2013. Available: <http://fimi.cs.helsinki.fi>.
- 633 30. Takeaki Uno, Masashi Kiyomi, Hiroki Arimura. LCM ver. 2: Efficient Mining Algorithms
634 for Frequent/Closed/Maximal Itemsets. In: *IEEE ICDM Workshop on Frequent Itemset*
635 *Mining Implementation (FIMI)*. 2004;126:1-11.
- 636 31. Han, J., Pei, J. and Yin, 2000, Mining frequent patterns without candidate generation, In:
637 *Proceedings of the ACM-SIGMOD conference management of data*, Vol.29, No.2, pp.1–
638 12.
- 639 32. Han, J., Pei, J., Yin, Y., and Mao, R. , 2004, Mining frequent patterns without candidate
640 generation: A frequent-pattern tree approach, *Data Mining and Knowledge Discovery*,
641 Vol.8, No.1, pp.53–87.
- 642 33. Sucahyo, Yudho Giri and Gopalan, Raj P. (2004), 'CT-PRO: A Bottom-Up Non Recur-
643 sive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure',
644 *IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI)*, Vol.126.
- 645 34. Huang, Z.K., et al., "A New Image Thresholding Method Based on Gaussian Mixture
646 Model," *Applied Mathematics and Computation* 205 (2): 899-907 2008.
- 647 35. Taormina, R., et al., "Artificial Neural Network simulation of hourly groundwater levels in
648 a coastal aquifer system of the Venice lagoon", *Engineering Applications of Artificial In-*
649 *telligence* 25 (8): 1670-1676 2012.
- 650 36. Wu, C.L., et al., "Predicting monthly streamflow using data-driven models coupled with
651 data preprocessing techniques," *Water Resources Research* 45, W08432, doi: 10.1029/
652 2007WR006737, 2009.
- 653 37. Zhang, J., et al., "Multilayer Ensemble Pruning via Novel Multi-sub-swarm Particle
654 Swarm Optimization," *Journal of Universal Computer Science* 15 (4): 840-858 2009.
- 655 38. Cheng, C.T., et al., "Long-term prediction of discharges in Manwan Reservoir using ar-
656 tificial neural network models," *Lecture Notes in Computer Science* 3498: 1040-1045
657 2005.
- 658 39. Chau, K.W., "Application of a PSO-based neural network in analysis of outcomes of
659 construction claims," *Automation in Construction* 16 (5): 642-646 2007.