

Original Research Article

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.

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 frequent patterns. The proposed work introduces novel learning and recall algorithms using CMM for mining frequent patterns efficiently.

Conclusion: The performance of the proposed system is compared with traditional algorithms like Apriori and FP-growth. The experimental results show order of magnitude improvement in execution time and storage space optimization to accumulate frequent patterns.

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

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

The rule mining algorithms have been continuously developed by both ANN and non-neural network researchers. Craven and Shavilk proposed rule mining as learning for if-then-else rule mining using a trained ANN [2, 3]. One approach to tackle the drawback of not gaining accurate interpretations of the nonlinear systems has been to develop methods of rule extraction [4, 5, 6, 7, 8, 9]. This approach turns the conversion of the real-valued knowledge formed in the weight matrix into symbolic rules. The rule extraction mechanisms already exist for different neural architectures such as multilayer perceptron [10, 11, 15, 18, 25], recurrent networks [13], self-organising networks [9, 14], associative memory networks [19], genetic algorithms [23,28], Fuzzy rules [21], and even for hybrid architectures [12]. They have focused on the formation of rules, in particular for classification tasks. A Hopfield network for rule mining has already been proposed by Gaber et al. [16], generating association rules from an ANN and employing incremental training for maintaining rules throughout time, is proposed by Eom and Zhang [17]. Multilevel association rule mining using Multi Level Feed Forward (MLFM) neural network is proposed as a specialize area of rule mining [20, 26]. A novel four phase data mining algorithm using ANN, referred as ESRNN (Extraction of Symbolic Rules from ANNs), is proposed for extracting symbolic rules [25]. The algorithm uses backpropagation learning.

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

This paper uses CMM structure of auto associative memory neural network effectively for storing and mining frequent patterns. Novel learning and recall algorithm is proposed for efficient frequent pattern mining.

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 and FP-growth 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 ij 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 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 , satisfying $Y \subseteq I$. A t_i is said to support an itemset X if $X \subseteq t_i.Y$.

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

The support of an itemset X with respect to D is defined by the fraction of transactions supporting it. This can be defined as follows:

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

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

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

Since it is impractical and not desirable to mine and generate the total space of itemsets and rules, the challenge of rule mining has been to discover only those rules which are interesting. The interestingness of an itemset and a rule is determined by evaluating the properties of support and confidence against the thresholds of *min_sup* and *min_conf* respectively. For instance, an itemset X is *frequent* iff $supp(X) \geq min_sup$.

2.2 Literature review

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

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

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

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

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

3. AUTO-ASSOCIATIVE MEMORY BASICS

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

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

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

Architecture of Auto-Associative Memory

The architecture of Auto-Associative Memory is shown in the Fig. 1. It is based on the concept of association. The neural network is first trained to store a set of patterns in the form $X : Y$ where X represents the input vector and Y is the corresponding output associative vector. Input patterns are represented as unipolar vectors. It has the property to remember the associativity expressed between input patterns under supervised training.

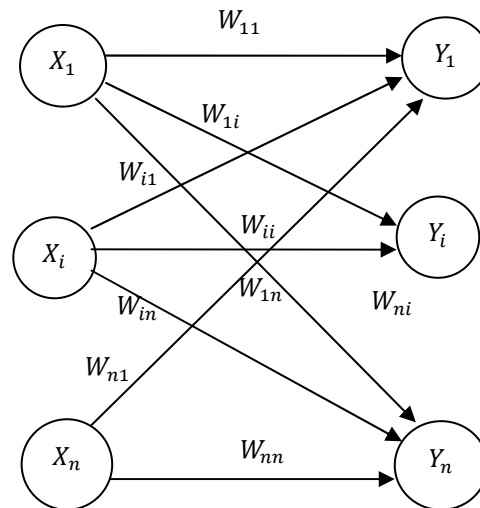


Fig. 1. Auto-Associative Memory

It uses CMM to store the associations among patterns. The CMM is a single-layer memory structure defined as square matrix whose dimension is the m number of elements in the patterns.

The CMM Models are binary weighted and non-binary weightless. The CMM resulting from supervised training is represented as $M \in B^{m \times m}$ for binary weighted model. The CMM is trained using a hebbian learning rule which offers excellent speed and scalability advantage.

es. 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 first learning algorithm is proposed using weighted CMM matrix. Then the proposed recall algorithm utilizes a novel approach to recall itemset support from trained CMM matrix. The following section describes a two step novel frequent pattern mining algorithm using CMM matrix memory.

4.1 Learning for itemset support using CMM

The weighted CMM matrix M is used to store frequencies or pattern occurrences in the form of weights. Learning is done using natural ability of the CMM to learn about occurrence frequency of the patterns based on input patterns.

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

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

Learning Algorithm:

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

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

- 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 onto Y_k , forms a mapping representing the association described by the k^{th} input pair in turn.

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

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

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

- 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).

193

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

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

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

198

199 In the weighted CMM M , w_{ij} also means frequency of pattern. Advantage of learning algo-
200 rithm is it reduces search space for recall mechanism to focus on the knowledge of only
201 $n(n+1)/2$ nodes rather than the n^2 elements of the complete matrix.

202 4.2 RECALL FOR FINDING FREQUENT PATTERNS USING CMM

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

208 Algorithm Recall (weighted CMM M , int min_sup ξ)

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

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

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

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

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

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

- 219 • Determine support for k -itemsets ($2 > k \leq m$) using following steps-

220 a. Only frequent 1-itemsets and frequent 2-itemsets are considered as per an-
 221 timonotone property (if subset of any pattern is infrequent then its superset
 222 are also infrequent)

223 b. Determine frequent k-itemsets applying equation as in (9) to combinations of
 224 only frequent 1-itemsets and frequent 2-itemsets stored in M .

$$225 \quad P(X) = \frac{\sum_{i=1}^k \sum_{j=i}^k P(w_{ij})}{n} \quad (9)$$

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

227 Proposed novel recall algorithm is more efficient as it finds k-itemsets ($2 > k \leq m$) using
 228 frequent 1-itemsets and frequent 2-itemsets and ignores all infrequent patterns.

229 As an example consider database DB as shown in Table 1, with 5 transactions. Let mini-
 230 mum support threshold ξ is 3.

231 **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

232

233 The output M of learning algorithm for example is as shown below:

234 Showing weighted CMM M

```

3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 2 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 2 3 1 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 0 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0
    
```

0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

2 1 2 0 1 2 0 0 0 0 0 2 0 0 0 0 0 0 0

3 1 3 1 1 3 1 0 1 0 0 2 3 0 0 0 0 0 0

1 0 1 0 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0

1 2 1 0 0 2 0 1 0 1 0 1 1 0 2 0 0 0 0

2 1 3 1 1 2 1 0 1 0 1 1 2 1 0 3 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1

235 The outputs of Recall algorithm for given example for certain cases are as shown below:

236

237

238

239

➤ Case 1: for $k = 3$, therefore ${}^3C_2=3$ so $X = (X_1, X_2, X_3)$
Then, for example

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

240

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

241

242

243

➤ Case 2: for $k = 4$, therefore ${}^4C_2 = 6$ so $X = (X_1, X_2, X_3, X_4)$. Then,

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

244

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

245

246

247

248

249

250

251

252

253

254

255

256

257

258

The final output of Recall algorithm for given example is: CMM matrix

Generating candidates: 1 - itemset

Possibilities: 19

{1}: 3.0 : 0.6

{2}: 3.0 : 0.6

{3}: 4.0 : 0.8

{6}: 4.0 : 0.8

{13}: 3.0 : 0.6

{16}: 3.0 : 0.6

Generating candidates: 2 - itemset

Possibilities: 171

259 {1 3}: 3.0 : 0.6
 260 {1 6}: 3.0 : 0.6
 261 {1 13}: 3.0 : 0.6
 262 {3 6}: 3.0 : 0.6
 263 {3 13}: 3.0 : 0.6
 264 {3 16}: 3.0 : 0.6
 265 {6 13}: 3.0 : 0.6
 266
 267 Generating candidates: 3 - itemset
 268 Possibilities: 969
 269 {1 3 6}: 3.0 : 0.6
 270 {1 3 13}: 3.0 : 0.6
 271 {1 6 13}: 3.0 : 0.6
 272 {3 6 13}: 3.0 : 0.6
 273
 274 Generating candidates: 4 - itemset
 275 Possibilities: 3876
 276 {1 3 6 13}: 3.0 : 0.6
 277
 278 Generating candidates: 5 - itemset
 279 Possibilities: 11628
 280 Time required finding frequent candidate is: 0.015 seconds
 281
 282 If the results are compared for a given example, time required to mine frequent patterns is
 283 less than the standard frequent pattern mining algorithms like Apriori and FP-growth algo-
 284 rithms.
 285

286 5. EXPERIMENTAL RESULTS

287
 288 This section will provide the experimental results of proposed system. Various parameters
 289 considered for comparison are time required for finding frequent 1-itemsets and total execu-
 290 tion time. In order to determine the accuracy of the proposed mechanisms for itemset sup-
 291 port recalls from a trained CMM, experiments with standard real datasets are performed in
 292 this section. The experiments were performed on an Intel Core2 Duo processor 1.66GHz
 293 with 2GB memory, running the Microsoft Windows. The standard datasets used in previous
 294 studies are downloaded from FIMI'04 website [29] and used to compare the algorithms per-
 295 formance. Experiments are performed on chess and mushroom dataset, where chess is
 296 denser than mushroom dataset. The experimental results of the proposed work are consis-
 297 tent for all the datasets.

298 Time required for finding frequent 1-itemsets using mushroom dataset is shown in table 2 for
 299 Apriori and proposed frequent pattern mining with an auto-associative memory using CMM
 300 algorithms. It proves that frequent 1-itemsets generation time efficiency obtained by CMM in
 301 real datasets like mushroom is also notable with respect to various minimum support thre-
 302 sholds.

303 **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

304

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%.

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

Table 4 shows comparison based on mining time for Apriori, FP-growth 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
FP Growth Time (Sec)	0.7	0.6	0.4	0.4	0.4	0.3	0.3	0.3	0.3

Mushroom dataset is a sparse dataset and as shown in Fig. 2 frequent pattern mining with an auto-associative memory using CMM shows better efficiency as compared to Apriori and FP-growth algorithms.

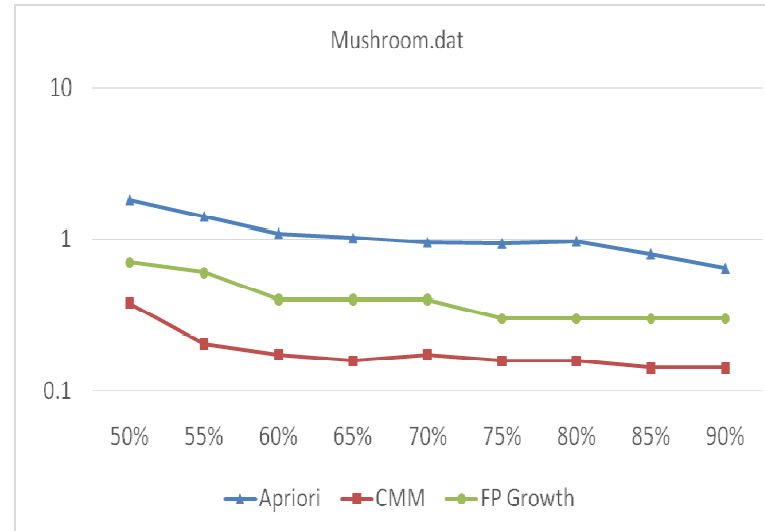


Fig. 2 Mining time for mushroom dataset

For chess dataset frequent pattern mining with an auto-associative memory using CMM achieves good mining time efficiency as shown in Fig. 3. Although being dense it takes comparatively lesser time for mining than Apriori and FP-growth algorithms.

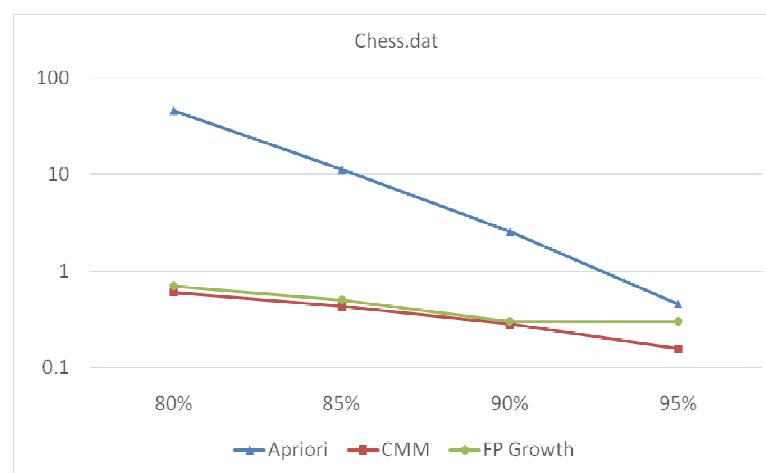


Fig. 3 Mining time for chess dataset

These experimental results prove efficiency of proposed CMM algorithm on dense and sparse datasets. Proposed CMM algorithm even work efficiently for low minimum supports.

6. CONCLUSION

Auto-Associative memory based CMM is best suitable neural network approach for association rule mining as it stores associations among patterns. Proposed learning algorithm 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. Proposed recall algorithm uses only frequent 1-patterns and frequent 2-patterns for calculating $(2 > k \leq m)$ itemsets, speeding up mining process. More scope of research is in advanced frequent mining like sequential pattern mining and distributed pattern mining.

REFERENCES

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

- 377 9. Malone J, McGarry K, Wermter S, Bowerman C. Data mining using rule extraction from
378 kohonen self-organising maps. *Neural Computing and Applications*. 2006;15(1):9–17.
- 379 10. Duch W, Adamczak R, Grabczewski K. Extraction of logical rules from training data
380 using backpropagation networks. *First Polish Conference on Theory and Applications of*
381 *Artificial Intelligence*. 1996;210.
- 382 11. McGarry KJ, Wermter S, MacIntyre J. Knowledge extraction from radial basis function
383 networks and multi-layer perceptrons. *IEEE International Conference on Neural*
384 *Networks (IJCNN'99)*, Washington DC. 1999;4:2494–249.
- 385 12. Eggermont J. Rule-extraction and learning in the BP-SOM architecture. 1998.
- 386 13. Jacobsson H. Rule Extraction from recurrent neural networks: A taxonomy and review.
387 *Neural Computing*. 2005;17(6):1223–1263.
- 388 14. Hammer B, Rechten, A., Strickert, M, Villmann T. Rule extraction from self-organizing
389 networks. Dorronsoro JR editor, 213 *ICANN, Lecture Notes in Computer Science*. 2002;
390 2415:877– 883.
- 391 15. Setiono R. Extracting M-of-N rules from trained neural networks. *IEEE-NN*. 2000;
392 11(2):512.
- 393 16. Gaber K, Bahi J, El-Ghazawi T. Parallel mining of association rules with a hopfield type
394 neural network. *12th IEEE International Conference on Tools with Artificial Intelligence*
395 *(ICTAI 2000)*, Vancouver, Canada. 2000;90–93.
- 396 17. Eom JH, Zhang BT. Prediction of east protein-protein interactions by neural feature
397 association rule. 2005.
- 398 18. Yihong Dong, Xiaoying Tai, Jieyu Zhao. A Distributed Algorithm Based on Competitive
399 Neural Network for Mining Frequent Patterns. *International Conference on Neural*
400 *Networks and Brain*. 2005;1:499- 503.
- 401 19. Vicente Oswaldo Baez Monroy and Simon O'Keefe. The Identification and Extraction of
402 Itemset Support defined by the Weight Matrix of a Self-Organising Map. *IEEE World*
403 *Congress on Computational Intelligence*. 2006.
- 404 20. Tong-yan Li, Xing-ming Li. Study of the neural network applied to weighted Association
405 Rules Mining. *Proc. International Conference on Wavelet Analysis and Pattern*
406 *Recognition*, Beijing, China. 2-4 Nov. 2007.
- 407 21. Au W, Chan K. Mining Fuzzy Association Rules. *Proc. of the 6th international conference*
408 *on Information and knowledge management*, Las Vegas, Nevada, USA. 209-215.
- 409 22. Baesens B, Setiono R, Mues C, Vanthienen J. Using neural network rule extraction and
410 decision tables for credit risk evaluation. *Management Science*. 2003;49(3):312-329.
- 411 23. Markowska U, Wnuk P. Rule Extraction from Neural Network by Genetic Algorithm with
412 Pareto Optimization. *Springer Lecture Notes in Computer Science*. 2004;3070(1):450-
413 455.
- 414 24. Robardet C, Cremilleux B, Boulicaut J. Characterization of Unsupervised Clusters with
415 the Simplest Association Rules: Application for Child's Meningitis. *Seventh International*
416 *Workshop on Intelligent Data Analysis in Biomedicine and Pharmacology (IDAMAP'02)*,
417 Lyon, France. 2002;61-66.
- 418 25. Kamruzzaman SM, Jehad Sarkar AM. A New Data Mining Scheme Using Artificial
419 Neural Networks. *Sensors*. 2011;11:4622-4647, doi:10.3390/s110504622.
- 420 26. Amit Bhagat, Sanjay Sharma, Pardasani KR. Ontological Frequent Patterns Mining by
421 potential use of Neural Network. *International Journal of Computer Applications* (0975–
422 8887). 2011;36(10):44-53.
- 423 27. Pao-Hua Chou, Pi-Hsiang Li, Kuang-Ku Chen, Menq-Jiun Wua. Integrating web mining
424 and neural network for personalized e-commerce automatic service. *Expert Systems*
425 *with Applications*. 2010;37:2898–2910.

- 426 28. Peter Wakabi-Waiswa, Venansius Baryamureeba, K Sarukesi. Generalized Association
427 Rule Mining Using Genetic Algorithms. Computer Science, Proceedings of Seventh
428 International conference in natural computation, IEEE. 2011;59-69.
- 429 29. Frequent Itemset Mining Implementations Repository (FIMI) Repository. 2004. Accessed
430 on 1 June 2013. Available: <http://fimi.cs.helsinki.fi>.