An Improved Rectangular Decomposition Algorithm for Imprecise and Uncertain Knowledge Discovery

Jiyoung Song, Younghee Im, and Daihee Park

Korea Univ. Dept. of Computer & Information Science
{songjy, yheeim, dhpark}@korea.ac.kr

Abstract. In this paper, we propose a novel improved algorithm for the rectangular decomposition technique for the purpose of performing fuzzy knowledge discovery from large scaled database in a dynamic environment. To demonstrate its effectiveness, we compare the proposed one which is based on the newly derived mathematical properties with those of other methods with respect to the classification rate, the number of rules, and complexity analysis.

1 Introduction

In the real world applications, the database is usually updated by an adding of new objects or by modification of old measurements. Maddouri[1] combined rectangular decomposition algorithm which is the incremental updates possible with fuzzy set theory to process imprecise and uncertain knowledge. The rectangular decomposition algorithms of Khcherif[2] and Maddouri[1] transform a binary matrix to a general graph first and find the maximum clique from the general graph. It is NP-hard problem[2]. However, our algorithm transforms a binary matrix to a bipartite graph first and finds bicliques from the bipartite graph by using node-deletion[3]. This can be solved in polynomial times[4].

In Section 2, we review rectangular decomposition techniques. In Section 3, we introduces a newly created rectangular decomposition algorithm and analyse the complexity. In Section 4, we present and analyze the result of experiments. Finally, concluding remarks are given in section 5.

2 Rectangular Decomposition Technique

Rectangular decomposition is a technique that finds an optimal coverage of binary relation R, where relational database is considered as binary relation, denoted $R(O, P)$, between a set of objects, O, and a set attribute value or properties P. The rectangle and the optimal coverage obtained in this step mean one rule by one-to-one mapping and minimum rule-base respectively [1] [2].

L. Wang and Y. Jin (Eds.): FSKD 2005, LNAI 3613, pp. 355–359, 2005.

3 Improved Algorithm

In this paper, we propose a novel improved algorithm that notably reduces the costs for optimal coverage keeping the advantages of incremental update feature of the rectangular decomposition technique under dynamic environments where there are more frequent insertions, deletions, and updates. For this purpose following theorems are derived based on bipartite graph obtained from transformation of the binary matrix.

Theorems 1. Let the given bipartite graph be $B = (V_1 \bigcup V_2, E)$, and the complete bipartite graph, a subset of B, be $B_c = (W_1 \bigcup W_2, E_c)$. Where $B_c = W_1 \times W_2$, $W_1 \subset V_1$, $W_2 \subset V_2$, $E_c \subset E$. If the rectangle obtained by the node-deletion is B_c, then B_c is the maximal rectangle.

Proof. To prove it by contradiction, let's assume that the conclusion is false. Then the B_c is not a maximal rectangle and there exists B'_c such that $B_c = W_1 \times W_2 \subset W'_1 \times W'_2 = B'_c$. Hence, the only following three cases are valid when the B_c becomes the proper subset of B'_c.

i) $W'_1 = W_1 + x \quad and \quad W'_2 = W_2$
ii) $W'_1 = W_1 \quad and \quad W'_2 = W_2 + y$
iii) $W'_1 = W_1 + x \quad and \quad W'_2 = W_2 + y$

For the case of i), let's define the elements of V_1 as $u_1, u_2, \cdots, u_m$, elements of V_2 as $v_1, v_2, \cdots, v_n$, elements of W_1 as $w_1, w_2, \cdots, w_i$, and elements of W_2 as $z_1, z_2, \cdots, z_j$. Let x be a temporary element of $V_1 - W_1$ that satisfies $x \in V_1 \quad and \quad x \notin W_1$. If we rearrange the elements in the sets as $u_k = w_k, v_l = z_l (1 \le k \le i, 1 \le l \le j)$ then x is one of elements in the set,$u_i+1, u_i+2, \cdots, u_m$. And if we let $W_1+x = W'_1$ then there exists $W'_1 \times W_2 = B'_c$. Since B'_c is a complete bipartite graph, x must have edges that connect all the elements of W_2 . However, in the node deletion technique, only the nodes without the edges to $\forall z (\in W_2)$ are deleted. Accordingly, no element x satisfies $x \in V_1 \quad and \quad x \notin W_1$ exists. This is a contradiction to the initial condition i). Therefore, the rectangle B_c which is obtained by node-deletion is maximal rectangle. ii) and iii) can be proved likewise.

Corollary. Let the given bipartite graph be $B = (V_1 \bigcup V_2, E)$, and the complete bipartite graph, a subset of B, be $B_c = (W_1 \bigcup W_2, E_c)$. Here, we know that $B_c = W_1 \times W_2, W_1 \subset V_1, W_2 \subset V_2, E_c \subset E$. Now, let be B_{c_1} from B using node deletion and B_{c_2} be B that was obtained from $e_2 \notin E_{c_1}$, the edge that is not included in B_{c_1}. Using the same method let B_{c_1} be B_c that was obtained from $e_3 \notin (E_{c_1} \bigcup E_{c_2})$ and get B_{c_n} for all edges,$e_i \in E$. If the set of all B_c is $CV = \{B_{c_1}, B_{c_2}, B_{c_3}, \cdots, B_{c_n}\}$ then is a coverage.

Proof. Since all $B_{c_i} \in CV$ are rectangle by **theorem 1** and all the edges, $e_i \in E$ is contained in CV , $CV = \{B_{c_1}, B_{c_2}, B_{c_3}, \cdots, B_{c_n}\}$ is a coverage by the definition of coverage.

Theorems 2. Let define the maximal rectangle B_{c_i} obtained by a node-deletion as $Rec_i (i = 1, 2, \cdots, n)$ and the coverage obtained from the corollary as $CV_{opt} =$

$\{Rec_1, Rec_2, \cdots, Rec_n\}$. If we set the each maximal rectangle as optimal rectangle then CV_{opt} is the optimal coverage.

Proof. Since we define each rectangle as optimal rectangle, it is sufficient to show that all elements of Rec_i in CV_{opt} are not redundant rectangles. This is same as showing that we can't form a coverage with $n-1$ number of Rec_i. Let's assume e_i is an edge which is the first pair of nodes to get $Rec_i(i = 1, 2, \cdots, n)$ in node-deletion. Let's first find out if the Rec_i is remaining rectangle. By the definition of the coverage all the edges must belong to at least one of rectangles. However, e_n was not included in any member of $Rec_i(i = 1, 2, \cdots, n-1)$ when $Rec_i, Rec_2, \cdots, Rec_{n-1}$ belong to CV_{opt} and each member of $Rec_i(i = 1, 2, \cdots, n-1)$ is optimal rectangle hence the optimal rectangle, Rec_n, which contains e_n must be included in CV_{opt}. Therefore, Rec_n is not a redundant rectangle. Likewise,$Rec_i, Rec_2, \cdots, Rec_{n-1}$) must be included in the coverage. A rectangle can't be included in one or multiple rectangles(since other rectangles are optimal rectangles) because each rectangle is an optimal rectangle. Consequently, the coverage can't be formed with $n-1$ number of rectangles. Therefore CV_{opt} is an optimal coverage by definition.

From the theorems derived above we could prove that it's possible to get desired optimal coverage by finding biclique directly from bipartite graph without transforming the bipartite graph that was obtained from binary matrix during the process of rectangular decomposition to general graph. Because the searching space that includes the solution for the problem to find biclique from the bipartite graph is $m \times n$ matrix, it becomes significantly smaller than $(m+n) \times (m+n)$ matrix from the methodology of Maddouri[1]. Moreover, finding the maximum clique is a problem regarding NP-hard while the problem of finding biclique from bipartite graph can only be solved by polynomial time. Hence this proves the method proposed in this paper is more efficient than Maddouri's methodology that solves NP-hard problem using the heuristic.

```
Input : Information Table (Binary relation)
Output : Optimal Coverage (Rule-base)
Opt_Cov()
{
  Coverage = ∅;
  Remain_Edge = All_Edge;
  All_List = Dom ∪ Cod;
  Candidate = ∅ ;
  while ( Remain_Edge != ∅ )
  {
    Coverage = Coverage ∪ Max_Rec(All_List, All_Edge);
    Remain_Edge = Remain_Edge − edge(Max_Rec);
    Candidate = Candidate_Node(Remain_Edge);
  }
  return(Coverage);
}
```

(a)

```
Input : List and Edge
Output : Maximal Rectangle
Max_Rec(List, Edge)
{
  Maximal = Candidate ;
  List = List − Candidate ;
  ∀n(∈List)   s. t. (Candidate, n)∈Edge
      List = List − n;
  while ( List != ∅ );
  {
    n = Best_node(List);
    Maximal = Maximal ∪ {n};
    List = List − n;
    ∀n'(∈List)   s. t. (n, n')∈Edge
      List = List − n';
  }
  return(Maximal);
}
```

(b)

Fig. 1. (a) Algorithm that finds the optimal coverage from the bipartite graph. (b) Algorithm that finds the maximal rectangle

	Method	Complexity	Knowledge representation
Rule Induction Methods	PVM	$\sum_{k=1}^{c}(c^2 d)^{2^{k-1}}$	Symbolic
	IPR	$(cd)^2(c+d)^2$	Symbolic
	FIPR	$(cd)^2(c+d)^2$	Linguistic
	Difunctional relation	$O(cd^2)$	Linguistic
	Proposed method	$O((c+d)d)$	Linguistic
Decision Tree Methods	CART	$c^{11}\times d$	Tree
	ID3	$c^2\times d^2$	Tree
	SPINA	$c^2\times d^2$	Latticial graph

Fig. 2. Comparison of machine learning methods

Fig. 1 show the proposed rectangular decomposition algorithm. The node of domain, node of co-domain, and edges represent property, object, and relation respectively.

The improved rectangular decomposition algorithm proposed in this paper modifies only the part of finding the optimal coverage. Therefore we can still keep the advantage of incremental updates from method proposed by Khcherif[2].

3.1 Complexity Analysis

To show its effectiveness, we compare the proposed method with other methods with respect to complexity and knowledge representation in Fig. 2

4 Experimental Results

To assess the proposed algorithm, experiments were performed using the IRIS data.

Table 1. Comparison of number of fuzzy rules and classification rate between conventional methods and proposed method

Number of training data	NM criterion[6]	RM criterion[6]	Jang [7]	Proposed method
21	89.8(71)	89.6(72)	88.3(32)	90.8(14)
30	93.0(83)	93.3(87)	91.6(36)	92.8(17)
60	93.9(105)	94.1(107)	93.3(46)	94.5(24)
90	94.8(150)	94.6(150)	95.0(46)	95.6(28)

In Table 1, the proposed method has less rules and higher classification rate than conventional methods. In newly proposed method, we created and classified the rules with 21 test data first and added 9, 30, and 30 data gradually to the initial 21 data for incremental updates on rule-base since the our method supports the incremental updates.

5 Conclusions

In this paper, we proposed a further improved algorithm for the rectangular decomposition technique with incremental updating for large scaled database in a dynamic environment. The conventional methods transform a binary matrix to a general graph first and find the maximum clique from the general graph. This is considered as an NP-hard problem. However the proposed method transforms a binary matrix to a bipartite graph first and finds bicliques from the bipartite graph by using node-deletion. This can be solved in polynomial times. The proposed algorithm is valid because it's based on the newly derived mathematical proofs. Also, it is not only effective but also has better results than conventional methods in comparisons of number of rules and the classification rates.

References

1. M. Maddouri, S. Elloumi, A. Jaoua: An Incremental Learning System for Imprecise and Uncertain Knowledge Discovery. Information Sciences. **109** (1998) 149-164
2. R. Khcherif, A. Jaoua: Rectangular Decomposition Heuristics for Documentary Databases. Information Sciences. **102** (1997) 187-202
3. M. Yannakakis: Node deletion problems on bipartite graphs. SIAM J. Comput. **10** (1981) 310-327
4. D. S. Hochbaum: Approximating Clique and Biclique Problems. Journal of Algorithm. **29** (1998) 174-200
5. R. Khcherif, M. M. Gammoudi, A. Jaoua: Using difunctional relations in information organization. Information Sciences. **125**(2000) 153-166
6. H. Ishibuchi, K. Nozaki, H. Tanaka: Efficient fuzzy partition of pattern space for classification problems. Fuzzy Sets and Systems. **59** (1993) 295-304
7. Jang, D-S., Choi, H-I.: Automatic Generation of Fuzzy Rules with Fuzzy Associative Memory. Proceeding of the ISCA 5th International Conference. (1996) 182-186