The Computational Complexity of Inference Using Rough Set Flow Graphs

C.J. Butz, W.Yan, B. Yang

Department of Computer Science, University of Regina, Regina, Canada, S4S 0A2,
{butz,yanwe111,boting}@cs.uregina.ca

Abstract. Pawlak recently introduced *rough set flow graphs* (RSFGs) as a graphical framework for reasoning from data. Each rule is associated with three coefficients, which have been shown to satisfy Bayes' theorem. Thereby, RSFGs provide a new perspective on Bayesian inference methodology.
In this paper, we show that inference in RSFGs takes polynomial time with respect to the largest domain of the variables in the decision tables. Thereby, RSFGs provide an efficient tool for uncertainty management. On the other hand, our analysis also indicates that a RSFG is a special case of conventional Bayesian network and that RSFGs make implicit assumptions regarding the problem domain.

1 Introduction

Bayesian networks [10] are a semantic modelling tool for managing uncertainty in complex domains. For instance, Bayesian networks have been successfully applied in practice by NASA [4] and Microsoft [5]. A Bayesian network consists of a *directed acyclic graph* (DAG) and a corresponding set of *conditional probability tables* (CPTs). The *probabilistic conditional independencies* [13] encoded in the DAG indicate that the product of the CPTs is a unique joint probability distribution. Although Cooper [1] has shown that the complexity of inference is NP-hard, several approaches have been developed that seem to work quite well in practice. Some researchers, however, reject any framework making probabilistic conditional independence assumptions regarding the problem domain.

Rough sets, founded by Pawlak's pioneering work in [8,9], are another tool for managing uncertainty in complex domains. Unlike Bayesian networks, no assumptions are made regarding the problem domain under consideration. Instead, the inference process is governed solely by sample data. Very recently, Pawlak introduced *rough set flow graphs* (RSFGs) as a graphical framework for reasoning from data [6,7]. Each rule is associated with three coefficients, namely, *strength*, *certainty* and *coverage*, which have been shown to satisfy Bayes' theorem. Therefore, RSFGs provide a new perspective on Bayesian inference methodology.

In this paper, we study the fundamental issue of the complexity of inference in RSFGs. Our main result is that inference in RSFGs takes polynomial time with respect to the largest domain of the variables in the decision tables. Thereby, RSFGs provide an efficient framework for uncertainty management. On the other hand, our analysis also indicates that a RSFG is a special case of Bayesian network. Moreover, unlike traditional rough set research, implicit independency assumptions regarding the problem domain are made in RSFGs.

This paper is organized as follows. Section 2 reviews the pertinent notions of Bayesian networks and RSFGs. The complexity of inference in RSFGs is studied in Section 3. In Section 4, we make a note on RSFG independency assumptions. The conclusion is presented in Section 5.

2 Background Knowledge

In this section, we briefly review Bayesian networks and RSFGs.

2.1 Bayesian Networks

Let $U = \{v_1, v_2, \ldots, v_m\}$ be a finite set of variables. Each variable v_i has a finite domain, denoted $dom(v_i)$, representing the values that v_i can take on. For a subset $X = \{v_i, \ldots, v_j\}$ of U, we write $dom(X)$ for the Cartesian product of the domains of the individual variables in X, namely, $dom(X) = dom(v_i) \times \ldots \times dom(v_j)$. Each element $x \in dom(X)$ is called a *configuration* of X.

A *joint probability distribution* [12] on $dom(U)$ is a function p on $dom(U)$ such that the following two conditions both hold: (i) $0 \leq p(u) \leq 1$, for each configuration $u \in dom(U)$, and (ii) $\sum_{u \in dom(U)} p(u) = 1.0$. A *potential* on $dom(U)$ is a function ϕ on $dom(U)$ such that the following two conditions both hold: (i) $0 \leq \phi(u)$, for each configuration $u \in dom(U)$, and (ii) $\phi(u) > 0$, for at least one configuration $u \in dom(U)$. For brevity, we refer to ϕ as a potential on U rather than $dom(U)$, and we call U, not $dom(U)$, its domain [12].

Let ϕ be a potential on U and $x \subseteq U$. Then the *marginal* [12] of ϕ onto X, denoted $\phi(X)$ is defined as: for each configuration $x \in dom(X)$,

$$\phi(x) = \sum_{y \in dom(Y)} \phi(x, y), \tag{1}$$

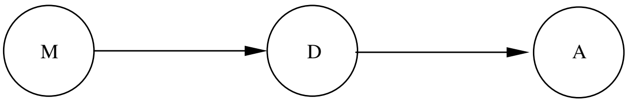
where $Y = U - X$, and x, y is the configuration of U that we get by combining the configuration, x of X and y of Y. The marginalization of ϕ onto $X = x$ can be obtained from $\phi(X)$.

A *Bayesian network* [10] on U is a DAG on U together with a set of *conditional probability tables* (CPTs) $\{\ p(v_i|P_i) \mid v_i \in U\ \}$, where P_i denotes the parent set of variable v_i in the DAG.

Example 1. One Bayesian network on $U = \{Manufacturer\ (M), Dealership\ (D), Age\ (A)\}$ is given in Figure 1.

We say X and Z are *conditionally independent* [13] given Y in a joint distribution $p(X, Y, Z, W)$, if

$$p(X, Y, Z) = \frac{p(X, Y) \cdot p(Y, Z)}{p(Y)}. \tag{2}$$



M	p (M)
Ford	0.20
Honda	0.30
Toyota	0.50

M	D	p (D | M)
Ford	Alice	0.60
Ford	Bob	0.30
Ford	Carol	0.00
Ford	Dave	0.10
Honda	Alice	0.00
Honda	Bob	0.50
Honda	Carol	0.50
Honda	Dave	0.00
Toyota	Alice	0.10
Toyota	Bob	0.30
Toyota	Carol	0.10
Toyota	Dave	0.50

D	A	p (A | D)
Alice	Old	0.30
Alice	Middle	0.60
Alice	Young	0.10
Bob	Old	0.40
Bob	Middle	0.60
Bob	Young	0.00
Carol	Old	0.00
Carol	Middle	0.60
Carol	Young	0.40
Dave	Old	0.10
Dave	Middle	0.30
Dave	Young	0.60

Fig. 1. A *Bayesian network* on $\{Manufacturer\ (M), \textit{Dealership}\ (D), \textit{Age}\ (A)\}$.

The *independencies* [13] encoded in the DAG of a Bayesian network indicate that the product of the CPTs is a unique joint probability distribution.

Example 2. The independency $I(M, D, A)$ encoded in the DAG of Figure 1 indicates that

$$p(M, D, A) = p(M) \cdot p(D|M) \cdot p(A|D), \tag{3}$$

where the joint probability distribution $p(M, D, A)$ is shown in Figure 2.

2.2 Rough Set Flow Graphs

Rough set flow graphs are built from decision tables. A *decision table* is a potential $\phi(C, D)$, where C is a set of conditioning attributes and D is a decision attribute. In [6], it is assumed that the decision tables are normalized, which we denote as $p(C, D)$.

Example 3. Consider the set $C = \{Manufacturer\ (M)\}$ of conditioning attributes and the decision attribute *Dealership* (D). One decision table $\phi_1(M, D)$, normalized as $p_1(M, D)$, is shown in Figure 3 (left). Similarly, a decision table on $C = \{\textit{Dealership}\ (D)\}$ and decision attribute *Age* (A), normalized as $p_2(D, A)$, is depicted in Figure 3 (right).

Each decision table defines a binary flow graph. The set of nodes in the flow graph are $\{c_1, c_2, \ldots, c_k\} \cup \{d_1, d_2, \ldots, d_l\}$, where $c_1, c_2, \ldots, c_k$ and $d_1, d_2, \ldots, d_l$ are the values of C and D appearing in the decision table, respectively. For each row in the decision table, there is a directed edge (c_i, d_j) in the flow graph, where

M	D	A	p(M,D,A)
Ford	Alice	Old	0.036
Ford	Alice	Middle	0.072
Ford	Alice	Young	0.012
Ford	Bob	Old	0.024
Ford	Bob	Middle	0.036
Ford	Dave	Old	0.002
Ford	Dave	Middle	0.006
Ford	Dave	Young	0.012
Honda	Bob	Old	0.060
Honda	Bob	Middle	0.090
Honda	Carol	Middle	0.090
Honda	Carol	Young	0.060
Toyota	Alice	Old	0.015
Toyota	Alice	Middle	0.030
Toyota	Alice	Young	0.005
Toyota	Bob	Old	0.060
Toyota	Bob	Middle	0.090
Toyota	Carol	Middle	0.030
Toyota	Carol	Young	0.020
Toyota	Dave	Old	0.025
Toyota	Dave	Middle	0.075
Toyota	Dave	Young	0.150

Fig. 2. The *joint probability distribution* $p(M, D, A)$ defined by the Bayesian network in Figure 1.

c_i is the value of C and d_j is the value of D. For example, given the decision tables in Figure 3, the respective binary flow graphs are illustrated in Figure 4.

Each edge (c_i, d_j) is labelled with three coefficients: *strength* $p(c_i, d_j)$, *certainty* $p(d_j|c_i)$ and *coverage* $p(c_i|d_j)$. For instance, the strength, certainty and coverage of the edges of the flow graphs in Figure 4 are shown in Figure 5.

It should perhaps be emphasized here that *all* decision tables $\phi(C, D)$ define a *binary* flow graph regardless of the cardinality of C. Consider a row in $\phi(C, D)$, where c and d are the values of C and D, respectively. Then there is a directed edge from node c to node d. That is, the constructed flow graph treats the attributes of C as a whole, even when C is a non-singleton set of attributes. For instance, in Example 1 of [6], the decision table $\phi(C, D)$ is defined over conditioning attributes $C = \{M, D\}$ and decision attribute A. One row in this table has M = "*Ford*", D = "*Alice*" and A = "*Middle*". Nevertheless, the constructed flow graph has an edge from node c_1 to node "*Middle*", where $c_1 = (M = \text{“}Ford\text{”}, D = \text{“}Alice\text{”})$. For simplified discussion, we will henceforth present all decision tables in which C is a singleton set.

M	D	ϕ_1 (M,D)	p_1 (M,D)
Ford	Alice	120	0.120
Ford	Bob	60	0.060
Ford	Dave	20	0.020
Honda	Bob	150	0.150
Honda	Carol	150	0.150
Toyota	Alice	50	0.050
Toyota	Bob	150	0.150
Toyota	Carol	50	0.050
Toyota	Dave	250	0.250

D	A	ϕ_2 (D,A)	p_2 (D,A)
Alice	Old	51	0.051
Alice	Middle	102	0.102
Alice	Young	17	0.017
Bob	Old	144	0.144
Bob	Middle	216	0.216
Carol	Middle	120	0.120
Carol	Young	80	0.080
Dave	Old	27	0.027
Dave	Middle	81	0.081
Dave	Young	162	0.162

Fig. 3. *Decision tables* $p_1(M, D)$ and $p_2(D, A)$, respectively.

In order to combine the collection of binary flow graphs into a general flow graph, Pawlak makes the *flow conservation* assumption [6]. This assumption means that the normalized decision tables are *pairwise consistent* [2,13].

Example 4. The two decision tables $p_1(M, D)$ and $p_2(D, A)$ in Figure 3 are pairwise consistent, since $p_1(D) = p_2(D)$. For instance, $p_1(D = \text{``}Alice\text{''}) = 0.170 = p_2(D = \text{``}Alice\text{''})$.

We now introduce the key notion of rough set flow graphs. A *rough set flow graph* (RSFG) [6,7] is a DAG, where each edge is associated with the strength, certainty and coverage coefficients. The task of inference is to compute $p(X = x|Y = y)$, where x and y are values of two distinct variables X and Y.

Example 5. The rough set flow graph for the two decision tables $p_1(M, D)$ and $p_2(D, A)$ in Figure 3 is the DAG in Figure 6 together with the appropriate strength, certainty and coverage coefficients in Figure 5. From these three coefficients, the query $p(M = \text{``}Ford\text{''}|A = \text{``}Middle\text{''})$, for instance, can be answered.

3 The Complexity of Inference

In this section, we establish the complexity of inference in RSFGs by polynomially transforming a RSFG into a Bayesian network and then stating the known complexity of inference. That is, if the RSFG involves nodes $\{a_1, a_2, \ldots, a_k, b_1, b_2, \ldots, b_l, \ldots, k_1, k_2, \ldots, k_m\}$, then the corresponding Bayesian network involves variables $U = \{A, B, \ldots, K\}$, where $dom(A) = \{a_1, a_2, \ldots, a_k\}$, $dom(B) = \{b_1, b_2, \ldots, b_l\}, \ldots, dom(K) = \{k_1, k_2, \ldots, k_m\}$.

Let G be a RSFG for a collection of decision tables. It is straightforward to transform G into a Bayesian network by applying the definition of RSFGs.

We first show that the Bayesian network has exactly one root variable. Let a_i be a root node in G. The strength of a_i is denoted as $\phi(a_i)$. Let $a_1, a_2, \ldots, a_k$

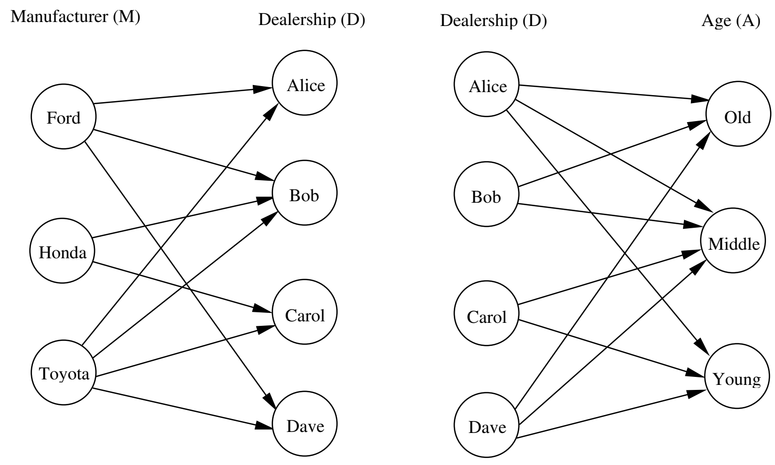


Fig. 4. The respective *binary* flow graphs for the decision tables in Figure 3, where the coefficients are given in Figure 5.

be all of the root nodes in G, that is, $a_1, a_2, \ldots, a_k$ have no incoming edges in G. By the definition of throughflow in [6],

$$\sum_{i=1}^{k} \phi(a_i) = 1.0. \tag{4}$$

In other words, there is one variable A in U, such that $dom(A) = \{a_1, a_2, \ldots, a_k\}$. In the Bayesian network, A is the only root variable.

By definition, the outflow [6] from one node in G is 1.0. Let $\{b_1, b_2, \ldots, b_l\}$ be the set of all nodes in G such that each b_i, $1 \leq i \leq l$, has at least one incoming

M	D	p_1(M,D)	p_1(D|M)	p_1(M|D)
Ford	Alice	0.12	0.60	0.71
Ford	Bob	0.06	0.30	0.16
Ford	Dave	0.02	0.10	0.07
Honda	Bob	0.15	0.50	0.42
Honda	Carol	0.15	0.50	0.75
Toyota	Alice	0.05	0.10	0.29
Toyota	Bob	0.15	0.30	0.42
Toyota	Carol	0.05	0.10	0.25
Toyota	Dave	0.25	0.50	0.93

D	A	p_2(D,A)	p_2(A|D)	p_2(D|A)
Alice	Old	0.05	0.30	0.23
Alice	Middle	0.10	0.60	0.19
Alice	Young	0.02	0.10	0.08
Bob	Old	0.14	0.40	0.63
Bob	Middle	0.22	0.60	0.42
Carol	Middle	0.12	0.60	0.23
Carol	Young	0.08	0.40	0.31
Dave	Old	0.03	0.10	0.14
Dave	Middle	0.08	0.30	0.15
Dave	Young	0.16	0.60	0.62

Fig. 5. The *strength* $p(a_i, a_j)$, *certainty* $p(a_j|a_i)$ and *coverage* $p(a_i|a_j)$ coefficients for the edges (a_i, a_j) in the two flow graphs in Figure 4, respectively.

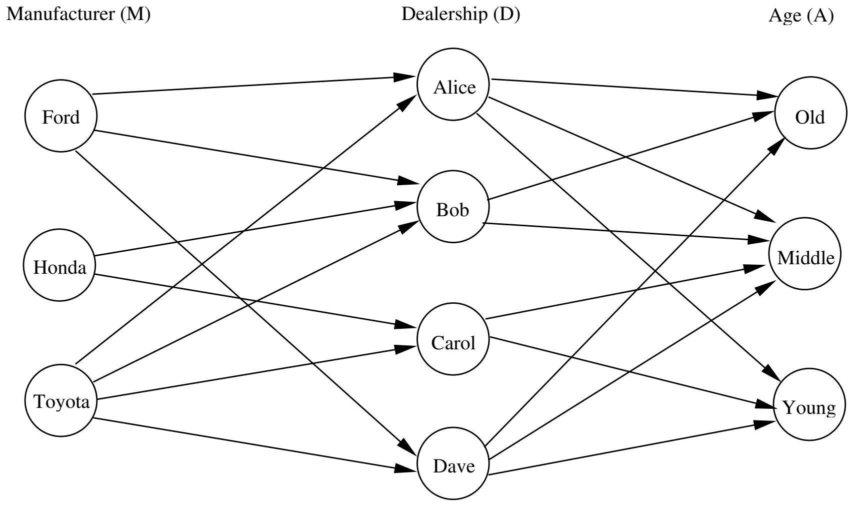


Fig. 6. The *rough set flow graph* (RSFG) for the two decision tables in Figure 3, where the strength, certainty and coverage coefficients can be found in Figure 5.

edge from a root node $a_1, a_2, \ldots, a_k$. By the definition of throughflow in [6],

$$\sum_{j=1}^{l} \phi(b_j) = 1.0. \tag{5}$$

This means there is a variable $B \in U$ such that $dom(B) = \{b_1, b_2, \ldots, b_l\}$. In the constructed Bayesian network of G, the root variable A has exactly one child B. This argument can be repeated to show that variable B has precisely one child, say C, and so on. The above discussion clearly indicates the structure of the Bayesian network constructed from G is a *chain*. In other words, there is only one root variable, and each variable except the last has exactly one child variable.

We now turn to the quantitative component of the constructed Bayesian network. For each variable v_i, a CPT $p(v_i|P_i)$ is required. Consider the root variable A. The CPT $p(A)$ is obtained from the strengths $\phi(a_1), \phi(a_2), \ldots, \phi(a_k)$. By Equation (4), $p(A)$ is a marginal distribution. We also require the CPT $p(B|A)$. Recall that every outgoing edge from nodes $a_1, a_2, \ldots, a_k$ must be an incoming edge for nodes $b_1, b_2, \ldots, b_l$. Moreover, let a_i be any node with at least one edge going to $b_1, b_2, \ldots, b_l$. Without loss of generality, assume a_i has edges to $b_1, b_2, \ldots, b_j$. This means we have edges $(a_i, b_1), (a_i, b_2), \ldots,$ $(a_i, b_j) \in G$. By definition, the certainty is

$$\phi(B = b_j | A = a_i) = \frac{\phi(A = a_i, B = b_j)}{\phi(A = a_i)}. \tag{6}$$

Since every decision table is normalized, $\phi(A = a_i, B = b_j) = p(A = a_i, B = b_j)$. Therefore, the certainty in Equation (6) is, in fact,

$$p(B = b_j | A = a_i). \tag{7}$$

Hence,

$$\sum_{m=1}^{j} p(B = b_m | A = a_i) = 1.0. \tag{8}$$

Equation (8) holds for each value $a_1, a_2, \ldots, a_k$ of A. Therefore, the conditional probabilities for all edges from $a_1, a_2, \ldots, a_k$ into $b_1, b_2, \ldots, b_l$ define a single CPT $p(B|A)$. This argument can be repeated for the remaining variables in the Bayesian network. Therefore, given a RSFG, we can construct a corresponding Bayesian network in polynomial time.

Example 6. Given the RSFG in Figure 6, the corresponding Bayesian network is shown in Figure 1.

There are various classes of Bayesian networks [10]. A *chain* Bayesian network has exactly one root variable and each variable except the last has precisely one child variable. A *tree* Bayesian network has exactly one root variable and each non-root variable has exactly one parent variable. A *singly-connected* Bayesian network, also known as a *polytree*, has the property that there is exactly one (undirected) path between any two variables. A *multiply-connected* Bayesian network means that there exist two nodes with more than one (undirected) path between them. Probabilistic inference in Bayesian networks means computing $p(X = x|Y = y)$, where $X, Y \subseteq U$, $x \in dom(X)$ and $y \in dom(Y)$. While Cooper [1] has shown that the complexity of inference in multiply-connected Bayesian networks is NP-hard, the complexity of inference in tree Bayesian networks is polynomial. Inference, which involves additions and multiplications, is bounded by multiplications. For a m-ary tree Bayesian network with n values in the domain for each node, one needs to store $n^2 + mn + 2n$ real numbers and perform $2n^2 + mn + 2n$ multiplications for inference [11].

We can now establish the complexity of inference in RSFGs by utilizing the known complexity of inference in the constructed Bayesian network. In this section, we have shown that a RSFG can be polynomially transformed into a chain Bayesian network. A chain Bayesian network is a special case of tree Bayesian network, that is, where $m = 1$. By substitution, the complexity of inference in a chain Bayesian network is $O(n^2)$. Therefore, the complexity of inference in RSFGs is $O(m^2)$, where $m = max(|dom(v_i)|)$, $v_i \in U$. In other words, the complexity of inference is polynomial with respect to the largest domain of the variables in the decision tables. This means that RSFGs are an efficient tool for uncertainty management.

4 Other Remarks on Rough Set Flow Graphs

One salient feature of rough sets is that they serve as a tool for uncertainty management without making assumptions regarding the problem domain. On

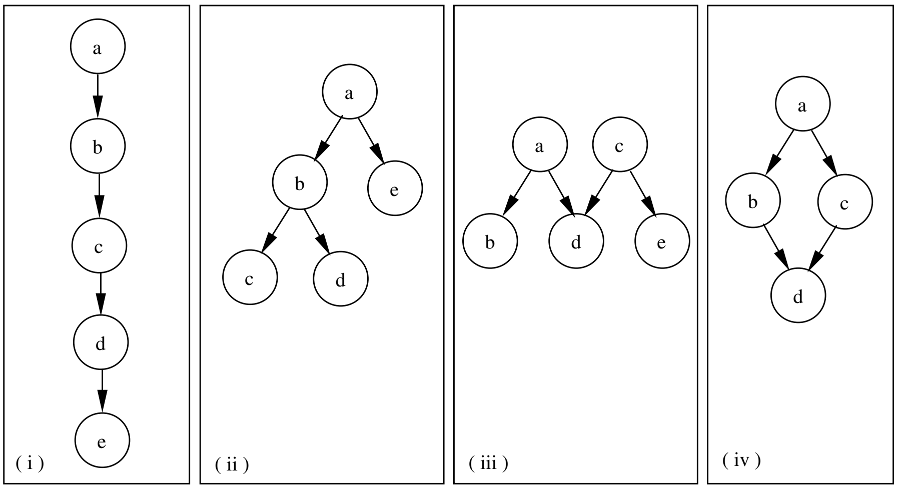


Fig. 7. Types of Bayesian network: (i) chain, (ii) tree, (iii) singly connected, and (iv) multiply-connected.

the contrary, we establish in this section that RSFGs, in fact, make implicit independency assumptions regarding the problem domain.

The assumption that decision tables $p_1(A_1, A_2)$, $p_2(A_2, A_3)$,..., $p_{m-1}(A_{m-1}, A_m)$ are pairwise consistent implies that the decision tables are marginals of a unique joint probability distribution $p(A_1, A_2, \ldots, A_m)$ defined as follows

$$p(A_1, A_2, \ldots, A_m) \;=\; \frac{p_1(A_1, A_2) \cdot p_2(A_2, A_3) \cdot \ldots \cdot p_{m-1}(A_{m-1}, A_m)}{p_1(A_2) \cdot \ldots \cdot p_{m-1}(A_{m-1})}. \qquad (9)$$

Example 7. Assuming the two decision tables $p_1(M, D)$ and $p_2(D, A)$ in Figure 3 are pairwise consistent implies that they are marginals of the joint distribution,

$$p(M, D, A) \;=\; \frac{p_1(M, D) \cdot p_2(D, A)}{p_1(D)}, \qquad (10)$$

where $p(M, D, A)$ is given in Figure 2.

Equation (9), however, indicates that the joint distribution $p(A_1, A_2, \ldots, A_m)$ satisfies $m-2$ probabilistic independencies $I(A_1, A_2, A_3 \ldots A_m)$, $I(A_1 A_2, A_3, A_4 \ldots A_m)$, ..., $I(A_1 \ldots A_{m-2}, A_{m-1}, A_m)$. In Example 7, assuming $p_1(M, D)$ and $p_2(D, A)$ are pairwise consistent implies that the independence $I(M, D, A)$ holds in the problem domain $p(M, D, A)$.

The important point is that the flow conservation assumption [6] used in the construction of RSFGs implicitly implies probabilistic conditional independencies holding in the problem domain.

5 Conclusion

Pawlak [6,7] recently introduced the notion of rough set flow graph (RSFGs) as a graphical framework for reasoning from data. In this paper, we established that the computational complexity of inference using RSFGs is polynomial with respect to the largest domain of the variables in the decision tables. This result indicates that RSFGs provide an efficient framework for uncertainty management. At the same time, our study has revealed that RSFGs, unlike previous rough set research, makes implicit independency assumptions regarding the problem domain. Moreover, RSFGs are a special case of Bayesian networks. Future work will study the complexity of inference in generalized RSFGs [3].

References

1. Cooper, G.F.: The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks. Artificial Intelligence, Vol. 42, Issue 2-3, (1990) 393-405
2. Dawid, A.P. and Lauritzen, S.L.: Hyper Markov Laws in The Statistical Analysis of Decomposable Graphical Models. The Annals of Satistics, Vol. 21 (1993) 1272-1317
3. Greco, S., Pawlak, Z. and Slowinski, R.: Generalized Decision Algorithms, Rough Inference Rules and Flow Graphs. The Third International Conference on Rough Sets, and Current Trends in Computing (2002) 93-104
4. Horvitz, E. and Barry, E.M.: Display of Information for Time Critical Decision Making. Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Francisco (1995) 296-305
5. Horvitz, E., Breese, J., Heckerman, D., Hovel, D. and Rommelse, K.: The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. Madison, WI (1998) 256-265
6. Pawlak, Z.: Flow Graphs and Decision Algorithms. The Ninth International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (2003) 1-10
7. Pawlak, Z.: In Pursuit of Patterns in Data Reasoning from Data - The Rough Set Way. The Third International Conference on Rough Sets, and Current Trends in Computing (2002) 1-9
8. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences, Vol. 11, Issue 5 (1982) 341-356
9. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic (1991)
10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco, California (1988)
11. Pearl, J.: Reverend Bayes on Inference Engines: A Distributed Heirarchical Approach. AAAI (1982) 133-136
12. Shafer, G.: Probabilistic Expert Systems. Society for the Institute and Applied Mathematics, Philadelphia (1996)
13. Wong, S.K.M., Butz, C.J. and Wu, D.: On the Implication Problem for Probabilistic Conditional Independency, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, Vol. 30, Issue 6. (2000) 785-805