An Efficient Algorithm for Inference in Rough Set Flow Graphs

C.J. Butz, W. Yan, and 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. No study, however, has yet investigated the complexity of the accompanying inference algorithm, nor the complexity of inference in RSFGs. In this paper, we show that the traditional RSFG inference algorithm has exponential time complexity. We then propose a new RSFG inference algorithm that exploits the factorization in a RSFG. We prove its correctness and establish its polynomial time complexity. In addition, we show that our inference algorithm never does more work than the traditional algorithm. Our discussion also reveals that, unlike traditional rough set research, RSFGs make implicit independency assumptions regarding the problem domain.

Keywords: Reasoning under uncertainty, rough set flow graphs.

1 Introduction

Very recently, Pawlak [7,8] introduced *rough set flow graphs* (RSFGs) as a graphical framework for uncertainty management. RSFGs extend traditional rough set research [9,10] by organizing the rules obtained from decision tables as a *directed acyclic graph* (DAG). Each rule is associated with three coefficients, namely, *strength*, *certainty* and *coverage*, which have been shown to satisfy Bayes' theorem [7,8]. Pawlak also provided an algorithm to answer queries in a RSFG and stated that RSFGs are a new perspective on Bayesian inference [7]. No study, however, has yet investigated the complexity of Pawlak's inference algorithm, nor the complexity of inference in RSFGs.

In this paper, our analysis of the traditional RSFG inference algorithm [7,8] establishes that its time complexity is exponential with respect to the number of nodes in a RSFG. We then propose a new inference algorithm that exploits the factorization in a RSFG. We prove the correctness of our algorithm and establish its polynomial time complexity. In addition, we show that our algorithm never does more work than the traditional algorithm, where work is the number of additions and multiplications needed to answer a query. The analysis in this manuscript also reveals that RSFGs make implicit assumptions regarding the problem domain. More specifically, we show that the *flow conservation assumption* [7] is in fact a *probabilistic conditional independency* [13] assumption.

J.F. Peters and A. Skowron (Eds.): Transactions on Rough Sets V, LNCS 4100, pp. 102–122, 2006.

It should be noted that the work here is different from our earlier work [2] in several important ways. In this manuscript, we propose a new algorithm for RSFG inference and establish its polynomial time complexity. On the contrary, we established the polynomial complexity of RSFG inference in [2] by utilizing the relationship between RSFGs and *Bayesian networks* [11]. Another difference is that here we show that RSFG inference algorithm in [7,8] has exponential time complexity, an important result not discussed in [2].

This paper is organized as follows. Section 2 reviews probability theory, RSFGs and a traditional RSFG inference algorithm [7,8]. That the traditional inference algorithm has exponential time complexity is shown in Section 3. In Section 4, we propose a new RSFG inference algorithm. We prove the correctness of this new algorithm and establish its polynomial time complexity in Section 5. Section 6 shows that it never does more work than the traditional algorithm. In Section 7, we observe that RSFGs make independence assumptions. The conclusion is presented in Section 8.

2 Definitions

In this section, we review probability theory and RSFGs.

2.1 Probability Theory

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 $c \in dom(X)$ is called a *configuration* of X. If c is a configuration on X and $Y \subseteq X$, then by c_Y we denote the configuration on Y by dropping from c the values of those variables not in Y.

A *potential* [12] on $dom(U)$ is a function ϕ on $dom(U)$ such that the following two conditions both hold: (i) $\phi(u) \geq 0$, 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]. By XY, we denote $X \cup Y$.

A *joint probability distribution* (jpd) [12] on U is a function p on U such that the following two conditions both hold: (i) $0 \leq \phi(u) \leq 1$, for each configuration $u \in U$, and (ii) $\sum_{u \in U} \phi(u) = 1.0$.

Example 1. Consider five attributes Manufacturer (M), Dealership (D), Age (A), Salary (S), Position (P). One jpd $p(U)$ on $U = \{M, D, A, S, P\}$ is depicted in Appendix I.

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

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