The Variable Precision Rough Set Inductive Logic Programming Model — a Statistical Relational Learning Perspective

R. S. Milton[1], **V. Uma Maheswari**[2] **and Arul Siromoney**[2]

[1] Department of Computer Science, Madras Christian College, Chennai – 600 059, India
[2] School of Computer Science and Engineering, Anna University, Chennai – 600 025, India
Contact email: asiro@vsnl.com

Abstract

The Variable Precision Rough Set Inductive Logic Programming model (VPRSILP model) extends the Variable Precision Rough Set (VPRS) model to Inductive Logic Programming (ILP). The VPRSILP model is considered from the Statistical Relational Learning perspective, by comparing and contrasting it with Stochastic Logic Programs.

Keywords: Rough Set Theory; Variable Precision Rough Sets; Inductive Logic Programming; Machine Learning; Statistical Relational Learning

1 Introduction

Inductive Logic Programming (ILP) [Muggleton, 1991] is the research area formed at the intersection of logic programming and machine learning. ILP uses background knowledge, and positive and negative examples to induce a logic program that describes the examples. The induced logic program consists of the original background knowledge along with an induced hypothesis.

Rough set theory [Pawlak, 1982; 1991] defines an indiscernibility relation, where certain subsets of examples cannot be distinguished. A concept is rough when it contains at least one such indistinguishable subset that contains both positive and negative examples. It is inherently not possible to describe the examples accurately, since certain positive and negative examples cannot be distinguished.

The gRS–ILP model [Siromoney, 1997; Siromoney and Inoue, 2002] introduces a rough setting in Inductive Logic Programming. It describes the situation where the background knowledge, declarative bias and evidence are such that any induced logic program cannot distinguish between certain positive and negative examples. Any induced logic program will either cover both the positive and the negative examples in the group, or not cover the group at all, with both the positive and the negative examples in this group being left out.

The Variable Precision Rough Set (VPRS) model [Ziarko, 1993] is a generalized model of rough sets that inherits all basic mathematical properties of the original rough set model. Rough Set Theory assumes that the universe under consideration is known and all the conclusions derived from the model are applicable only to this universe. In practice, however, there is an evident need to generalize conclusions obtained from a smaller set of examples to a larger population. The VPRS model allows for a controlled degree of misclassification. Any partially incorrect classification rule provides valuable trend information about future test cases if the majority of available data to which such a rule applies can be correctly classified.

This paper presents the Variable Precision Rough Set Inductive Logic Programming model [Maheswari *et al.*, 2001b], an extension of the gRS–ILP model using features of the VPRS model, and compares and contrasts this model with Stochastic Logic Programs [Muggleton, 2000].

2 Inductive Logic Programming

The semantics of ILP systems are discussed in [Muggleton and Raedt, 1994]. In ILP systems, background (prior) knowledge B and evidence E (consisting of positive evidence E^+ and negative evidence E^-) are given, and the aim is then to find a hypothesis H such that certain conditions are fulfilled.

In the *normal semantics*, the background knowledge, evidence and hypothesis can be any well-formed logical formula. The conditions that are to be fulfilled by an ILP system in the normal semantics are

Prior Satisfiability: $B \wedge E^- \not\models \Box$
Posterior Satisfiability: $B \wedge H \wedge E^- \not\models \Box$
Prior Necessity: $B \not\models E^+$
Posterior Sufficiency: $B \wedge H \models E^+$

However, the *definite semantics*, which can be considered as a special case of the normal semantics, restricts the background knowledge and hypothesis to being definite clauses. This is simpler than the general setting of normal semantics, since a definite clause theory T has a unique minimal Herbrand model $\mathcal{M}^+(T)$, and any logical formula is either true or false in the minimal model. The conditions that are to be fulfilled by an ILP system in the definite semantics are

Prior Satisfiability: all $e \in E^-$ are false in $\mathcal{M}^+(B)$
Posterior Satisfiability: all $e \in E^-$ are false in $\mathcal{M}^+(B \wedge H)$
Prior Necessity: some $e \in E^+$ are false in $\mathcal{M}^+(B)$
Posterior Sufficiency: all $e \in E^+$ are true in $\mathcal{M}^+(B \wedge H)$

The Sufficiency criterion is also known as *completeness* with respect to positive evidence and the Posterior Satisfiability criterion is also known as *consistency* with the negative evidence.

The special case of definite semantics, where evidence is restricted to true and false ground facts (examples), is called the *example* setting. The example setting is thus the normal semantics with B and H as definite clauses and E as a set of ground unit clauses. The example setting is the main setting of ILP employed by the large majority of ILP systems.

3 Formal definitions of the gRS–ILP model

The generic Rough Set Inductive Logic Programming (gRS–ILP) model introduces the basic definition of elementary sets and a rough setting in ILP [Siromoney, 1997; Siromoney and Inoue, 2002]. The essential feature of an elementary set is that it consists of examples that cannot be distinguished from each other by any induced logic program in that ILP system. The essential feature of a rough setting is that it is inherently not possible for certain positive and negative examples to be distinguished, since both these positive and negative examples are in the same elementary set. The basic definitions formalised in [Siromoney and Inoue, 2000] follow.

The ILP system in the example setting of [Muggleton and Raedt, 1994] is formally defined as follows.

Definition 3.1. An *ILP system in the example setting* is a tuple $S_{es} = (E_{es}, B)$, where
(1) $E_{es} = E^+_{es} \cup E^-_{es}$ is the *universe*, where E^+_{es} is the set of positive examples (true ground facts), and E^-_{es} is the set of negative examples (false ground facts), and
(2) B is a background knowledge given as definite clauses such that (i) for all $e^- \in E^-_{es},\ B \not\vdash e^-$, and (ii) for some $e^+ \in E^+_{es},\ B \not\vdash e^+$.

Let $S_{es} = (E_{es}, B)$ be an ILP system in the example setting. Then let $\mathcal{H}(S_{es})$ (also written as $\mathcal{H}(E_{es}, B)$) denote the set of all possible definite clause hypotheses that can be induced from E_{es} and B, and be called the *hypothesis space* induced from S_{es} (or from E_{es} and B). Further, let $\mathcal{P}(S_{es})$ (also written as $\mathcal{P}(E_{es}, B) = \{P = B \wedge H \mid H \in \mathcal{H}(E_{es}, B)\}$) denote the set of all the programs induced from E_{es} and B, and be called the *program space* induced from S_{es} (or from E_{es} and B).

The aim is to find a program $P \in \mathcal{P}(S_{es})$ such that the next two conditions hold: (iii) for all $e^- \in E^-_{es},\ P \not\vdash e^-$, (iv) for all $e^+ \in E^+_{es},\ P \vdash e^+$.

The following simple illustration is used to explain this definition. Let $S = (E, B)$ where $E = E^+ \cup E^-$,
$E^+ = \{p(d1), p(d2), p(d3)\}$,
$E^- = \{p(d4), p(d5), p(d6)\}$ and
$B = \{atom(d1, c), atom(d2, c), atom(d3, o), atom(d4, o), atom(d5, n), atom(d6, n)\}$. Without loss of generality, only six examples are considered $p(d1)$, $p(d2)$, $p(d3)$, $p(d4)$, $p(d5)$, $p(d6)$ in our universe of examples. The background knowledge B indicates that the positive example molecule d_1 has a carbon atom, negative example molecule d_4 has an oxygen atom, negative example molecule d_5 has a nitrogen atom, and so on. The background knowledge B has only ground facts, using the predicate $atom$, and so does not cover any example. It is seen that for all $e^- \in E^-,\ B \not\vdash e^-$, and for some $e^+ \in E^+,\ B \not\vdash e^+$. (Two conditions (i) and (ii) of an ILP system in the example setting hold.) Let $H = \{p(d1), p(d2), p(d3)\}$. Then for all $e^- \in E^-,\ B \wedge H \not\vdash e^-$, and for all $e^+ \in E^+,\ B \wedge H \vdash e^+$. (Two conditions (iii) and (iv) also hold.)

The following definitions of Rough Set ILP systems in the gRS–ILP model (abbreviated as *RSILP systems*) use the terminology of [Muggleton and Raedt, 1994].

Definition 3.2. An *RSILP system in the example setting* (abbreviated as RSILP–E system) is an ILP system in the example setting, $S_{es} = (E_{es}, B)$, such that there does not exist a program $P \in \mathcal{P}(S_{es})$ satisfying both the conditions (iii) and (iv) above.

Definition 3.3. An *RSILP–E system in the single–predicate learning context* (abbreviated as RSILP–ES system) is an RSILP–E system, whose *universe* E is such that all examples (ground facts) in E use only one predicate, also known as the *target predicate*.

A *declarative bias* [Muggleton and Raedt, 1994] restricts the set of acceptable hypotheses, and is of two kinds: *syntactic bias* (also called *language bias*) that imposes restrictions on the form (syntax) of clauses allowed in the hypothesis, and *semantic bias* that imposes restrictions on the meaning, or the behaviour of hypotheses.

Definition 3.4. An *RSILP–ES system with declarative bias* (abbreviated as RSILP–ESD system) is a tuple $S = (S', L)$, where
(i) $S' = (E, B)$ is an RSILP–ES system, and
(ii) L is a declarative bias, which is any restriction imposed on the hypothesis space $\mathcal{H}(E, B)$.
We also write $S = (E, B, L)$ instead of $S = (S', L)$.

For any RSILP–ESD system $S = (E, B, L)$, let
$\mathcal{H}(S) = \{H \in \mathcal{H}(E, B) \mid H \text{ is allowed by } L\}$, and
$\mathcal{P}(S) = \{P = B \wedge H \mid H \in \mathcal{H}(S)\}$.
$\mathcal{H}(S)$ (also written as $\mathcal{H}(E, B, L)$) is called the *hypothesis space* induced from S (or from E, B, and L). $\mathcal{P}(S)$ (also written as $\mathcal{P}(E, B, L)$) denotes the set of all the programs induced by S, and is called the *program space* induced from S (or from E, B, and L).

It is seen in the illustration used earlier that the ILP system can exactly describe the set of positive examples, but in a manner that is not very useful, since the hypothesis is the same as the positive example ground facts. If the terms $d1, \ldots, d6$ are not allowed in H, then with $H = \{p(A) \leftarrow$

$atom(A,c)\}$, for all $e^- \in E^-$, $B \wedge H \not\vdash e^-$. However it is not true that for all $e^+ \in E^+$, $B \wedge H \vdash e^+$, since $B \wedge H \not\vdash p(d3) \in E^+$. (Condition (iii) holds, but not condition (iv).)
With $H = \{p(A) \leftarrow atom(A,c), p(A) \leftarrow atom(A,o)\}$, for all $e^+ \in E^+$, $B \wedge H \vdash e^+$. However it is not true that for all $e^- \in E^-$, $B \wedge H \not\vdash e^-$, since $B \wedge H \vdash p(d4) \in E^-$. (Condition (iv) holds, but not condition (iii).)

This is formalised in the definition of the RSILP–ESD system. Let $S = (E, B, L)$ where E and B are as given above, and L is the declarative bias such that $d1, \ldots, d6$ is not a term in $q(\ldots)$ for any $H \in \mathcal{H}(S)$, any $C \in H$, and any predicate $q(\ldots) \in C$.

An equivalence relation on the universe of an RSILP–ESD system is now defined.

Definition 3.5. Let $S = (E, B, L)$ be an RSILP–ESD system. An indiscernibility relation of S, denoted by $R(S)$, is a relation on E defined as follows: $\forall x, y \in E,\ (x, y) \in R(S)$ iff
$(P \vdash x \Leftrightarrow P \vdash y)$ for any $P \in \mathcal{P}(S)$ (i.e. iff x and y are inherently indistinguishable by any induced logic program P in $\mathcal{P}(S)$).

The following fact follows directly from the definition of $R(S)$.

Fact 1 *For any RSILP–ESD system S, $R(S)$ is an equivalence relation.*

Definition 3.6. Let $S = (E, B, L)$ be an RSILP–ESD system. An *elementary set* of $R(S)$ is an equivalence class of the relation $R(S)$. For each $x \in E$, let $[x]_{R(S)}$ denote the elementary set of $R(S)$ containing x. Formally,
$[x]_{R(S)} = \{y \in E \mid (x, y) \in R(S)\}$.
A *composed set* of $R(S)$ is any finite union of elementary sets of $R(S)$.

Definition 3.7. An RSILP–ESD system $S = (E, B, L)$ is said to be in a *rough setting* iff
$\exists e^+ \in E^+\ \exists e^- \in E^-\ (\ (e^+, e^-) \in R(S)\ )$.

It is seen from E, B, and L in the illustration used earlier that $R(S) = \{\ (p(d1), p(d2)),\ (p(d2), p(d1)),\ (p(d3), p(d4)),\ (p(d4), p(d3)),\ (p(d5), p(d6)),\ (p(d6), p(d5))\ \}$.

The elementary sets of $R(S)$ are
$\{p(d1), p(d2)\}$, $\{p(d3), p(d4)\}$, $\{p(d5), p(d6)\}$.

The composed sets of $R(S)$ are
$\{\}$, $\{p(d1), p(d2)\}, \ldots, \{p(d1), p(d2), p(d3), p(d4)\}, \ldots,$
$\{p(d1), p(d2), p(d3), p(d4), p(d5), p(d6)\}$.

S is in a rough setting since $p(d3) \in E^+$, $p(d4) \in E^-$ and $(p(d3), p(d4)) \in R(S)$.

Other work in Rough Set Inductive Logic Programming include [Midelfart and Komorowski, 2000; Liu and Zhong, 1999].

4 Formal definitions of the VPRSILP model

The formal definitions of the VPRSILP model are defined in [Maheswari *et al.*, 2001b].

A parameter β, a real number in the range $(0.5, 1]$, is used in the VPRS model as a threshold in elementary sets that have both positive and negative examples. This threshold is used to decide if that elementary set can be classified as positive or negative, depending on the statistical occurrence of positive and negative examples in it.

Definition 4.1. A *Variable Precision RSILP–ESD system* (abbreviated as VPRSILP–ESD system) is a tuple $S = (S', \beta)$, where
(i) $S' = (E, B, L)$ is an RSILP–ESD system, and
(ii) β is a real number in the range $(0.5, 1]$.
It is also written $S = (E, B, L, \beta)$ instead of $S = (S', \beta)$.

The definitions of hypothesis space, program space, equivalence relation, elementary sets, composed sets and rough setting defined above for RSILP–ESD systems hold for the VPRSILP–ESD system.

The following definitions use the VPRS terminology from [An *et al.*, 1997].

Definition 4.2. The *conditional probability* $P(E^+ \mid [x]_{R(S)})$ is defined as

$$P(E^+ \mid [x]_{R(S)}) = \frac{P(E^+ \cap [x]_{R(S)})}{P([x]_{R(S)})} = \frac{\mid E^+ \cap [x]_{R(S)}) \mid}{\mid [x]_{R(S)} \mid}$$

where $P(E^+ \mid [x]_{R(S)})$ is the probability of occurrence of event E^+ conditioned on event $[x]_{R(S)}$.

It is noted that $P(E^+ \mid [x]_{R(S)}) = 1$ if and only if $[x]_{R(S)} \subseteq E^+$;
$P(E^+ \mid [x]_{R(S)}) > 0$ if and only if $[x]_{R(S)} \cap E^+ \neq \emptyset$;
and $P(E^+ \mid [x]_{R(S)}) = 0$ if and only if $[x]_{R(S)} \cap E^+ = \emptyset$.

Definition 4.3. The *β–positive region* of S, $Pos_\beta(S)$, is defined as
$Pos_\beta(S) = \bigcup_{P(E^+ \mid [x]_{R(S)}) >= \beta,\ \text{for all } [x]_{R(S)} \text{ in } R(S)} \{[x]_{R(S)}\}$
The *β–negative region* of S, $Neg_\beta(S)$, is defined as
$Neg_\beta(S) = \bigcup_{P(E^+ \mid [x]_{R(S)}) < \beta,\ \text{for all } [x]_{R(S)} \text{ in } R(S)} \{[x]_{R(S)}\}$

Definition 4.4. The *β–restricted program space* of S, $\mathcal{P}_\beta(S)$ (also written as $\mathcal{P}_\beta(E, B, L, \beta)$), is defined as
$\mathcal{P}_\beta(S) = \{P \in \mathcal{P}(S) \mid P \vdash x \Rightarrow x \in Pos_\beta(S)\}$.
Any $P \in \mathcal{P}_\beta(S)$ is called a β–restricted program of S.

Our aim is to find a hypothesis H such that $P = B \wedge H \in \mathcal{P}_\beta(\mathcal{S})$.

The VPRSILP model has been applied in illustrative experiments to determine the transmembrane domains in amino acid sequences [Maheswari *et al.*, 2001b] and to analyse and classify web log data [Maheswari *et al.*, 2001a; 2003].

5 The VPRSILP model and Stochastic Logic Programs (SLP)

A clause C is said to be *range–restricted* if and only if every variable in the head of C is found in the body of C. A *stochastic clause* is a pair $p : C$ where p in the interval $[0, 1]$ is the probability associated with C, a range–restricted clause. A set of stochastic clauses P is called a *Stochastic Logic Program* (SLP) if and only if for each predicate symbol q in P the probabilities associated with all clauses with q in the head sum to 1 [Muggleton, 2000]. Every example derived from an SLP has a probability associated with it. This is the product of the probabilities associated with every clause used in the derivation of the example.

In a VPRSILP–ESD system $S = (E, B, L, \beta)$, every example x in E falls into one of the elementary sets, $[x]_{R(S)}$. An elementary set in VPRSILP, by definition, consists of examples that are indistinguishable by any logic program P that can be induced from the examples E, background knowledge B and declarative bias L. Every elementary set $[x]_{R(S)}$ has a conditional probability $P(E^+ \mid [x]_{R(S)})$ associated with it depending on the statistical occurrence of positive and negative examples in it.

Hence every example in VPRSILP has a probability based value associated with it, namely the conditional probability of the elementary set in which this example occurs. This is in some sense similar to the probability associated with an example derived from a stochastic logic program.

However, further study of the comparison between these two models is required.

The following points are observed.

In VPRSILP, the conditional probability associated with an elementary set is based on the examples. In other words, the conditional probability is obtained from the observed data. In SLP, a probability is associated with each clause, probably using domain knowledge.

In VPRSILP, a probability value is associated with the negative examples also. In SLP a probability value of 0 is assigned to an example that is rejected by the SLP.

In VPRSILP, the conditional probability associated with an elementary set uses both negative and positive examples in that elementary set. In other words, the negative examples also play a role in determining the probability value. In SLP, the probability value is associated with a clause, and so only the examples that are derived (the positive examples) play a role.

In VPRSILP, the conditional probabilities associated with the elementary sets do not form a probability distribution. In SLP, the probabilities associated with each clause with the same head predicate form a probability distribution.

6 The VPRSILP model and application to Predictive Toxicology

In this section, the cVPRSILP approach based on the VPRSILP model is outlined [Milton *et al.*, 2003]. In the cVPRSILP approach, elementary sets are defined using attributes that are based on a finite number of clauses of interest.

Predictive Toxicology Evaluation

The rodent carcinogenicity tests conducted within the US National Toxicology Program by the National Institute of Environmental Health Sciences (NIEHS). has resulted in a large database of compounds classified as carcinogens or otherwise. The Predictive Toxicology Evaluation project of the NIEHS provided the opportunity to compare carcinogenicity predictions on previously untested chemicals. This presented a formidable challenge for programs concerned with knowledge discovery. The ILP system Progol [Muggleton, 1995] has been used in this Predictive Toxicology Evaluation Challenge [Srinivasan *et al.*, 1997a; 1997b].

Elementary Sets

In [Pawlak and Skowron, 1999], two finite, nonempty sets U and A are considered, where U is the universe of objects, and A is a set of attributes. With every attribute $a \in A$ is associated a set V_a of its values, called the domain of a.

The set of attributes A determines a binary relation R on U. R is an indiscernibility relation, defined as follows: xRy if and only if $a(x) = a(y)$ for every $a \in A$; where $a(x) \in V_a$ denotes the value of attribute a for object x. Obviously R is an equivalence relation. Equivalence classes of the relation R are referred to as elementary sets.

In cVPRSILP, let $A = \{A_1, \ldots, A_{i_{max}}\}$ be the set of attributes, with $V_a = \{\texttt{true}, \texttt{false}\}$ for every $a \in A$. Every $A_i \in A$ is associated with the clauses of interest $C'_i, i = 1, ..., i_{max}$, such that $A_i = \texttt{true}$ if the example can be derived from $C'_i \wedge B$, and $A_i = \texttt{false}$ otherwise. In this context, it is seen that these attributes form an equivalence relation.

Beta positive and beta negative regions

Elementary sets formed from the training examples fall into either the β–positive or the β–negative region, depending on the value of β.

A test example is decided as being positive or negative, depending on whether its elementary set is in the β–positive or the β–negative region.

Experimental illustration

An illustrative experiment is performed using the cVPRSILP model. The dataset used is the Predictive Toxicology Evaluation Challenge dataset found at
`http://web.comlab.ox.ac.uk/oucl/research/areas/`

In this experimental illustration, two predicates `has_property` and `atm` with four properties and three atom types are considered. These have been heuristically chosen based on visual inspection of clauses induced by Progol. Further studies are in progress to arrive at a more systematic choice.

The maximum number of predicates in a clause is taken as 2 and a finite set of clauses of interest is generated.

Each of the clauses of interest is treated as an attribute, and every example is placed in the appropriate elementary set, based on the subset of clauses which cover that example. Each elementary set falls in the β–positive or the β–negative region, depending on the chosen value of β. In this illustration, we use the value of 0.5. An example is predicted positive if its elementary set falls in the β–positive region, and is predicted negative if the elementary set falls in the β–negative region.

The following table is obtained when prediction is done on the training set itself. The overall prediction accuracy is 86%. Further analysis needs to be done.

	Actual Positive	Actual Negative	
Predicted Positive	142	22	164
Predicted Negative	16	111	127
Unclassified	0	2	2
	158	135	293

7 Conclusions

The VPRSILP model combines statistical and relational perspectives. The utility of the model has already been shown in classification experiments in computational biology and web mining. A brief discussion on the similarities and differences of the model with Stochastic Logic Programs, a Statistical Relational Learning paradigm, is presented.

References

[An *et al.*, 1997] A. An, C. Chan, N. Shan, N. Cercone, and W. Ziarko. Applying knowledge discovery to predict water-supply consumption. *IEEE Expert*, 12(4):72–78, 1997.

[Liu and Zhong, 1999] C. Liu and N. Zhong. Rough problem settings for Inductive Logic Programming. In N.Zhong and A.Skowron ad S.Ohsuga, editors, *New Directions in Rough Sets, Data Mining, and Granular–Soft Computing — 7th International Workshop, RSFDGrC'99*, Lecture Notes in Artificial Intelligence 1711, pages 168–177, Yamaguchi, Japan, November 1999. Springer.

[Maheswari *et al.*, 2001a] V. Uma Maheswari, Arul Siromoney, and K. M. Mehata. The Variable Precision Rough Set Inductive Logic Programming model and web usage graphs. In *New Frontiers in Artificial Intelligence — Joint JSAI 2001 Workshop Post–Proceedings*, volume 2253, pages 339–343. Lecture Notes in Computer Science, Springer, 2001.

[Maheswari *et al.*, 2001b] V. Uma Maheswari, Arul Siromoney, K. M. Mehata, and K. Inoue. The Variable Precision Rough Set Inductive Logic Programming Model and Strings. *Computational Intelligence*, 17(3):460–471, August 2001.

[Maheswari *et al.*, 2003] V. Uma Maheswari, Arul Siromoney, and K. M. Mehata. The Variable Precision Rough Set Inductive Logic Programming model and future test cases in web usage mining. In Masahiro Inuiguchi, Shusaku Tsumoto, and Shoji Hirano, editors, *Rough Set Theory and Granular Computing*. Physica–Verlag, 2003.

[Midelfart and Komorowski, 2000] H. Midelfart and J. Komorowski. A Rough Set approach to Inductive Logic Programming. In W. Ziarko and Y. Yao, editors, *Rough Sets and Current Trends in Computing — Second International Conference, RSCTC 2000*, Lecture Notes in Artificial Intelligence 2005, pages 190–198, Banff, Canada, October 2000. Springer.

[Milton *et al.*, 2003] R. S. Milton, V. Uma Maheswari, and Arul Siromoney. The variable precision rough set inductive logic programming model and predictive toxicology. 2003. Submitted.

[Muggleton and Raedt, 1994] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19(20):629–679, 1994.

[Muggleton, 1991] S. Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

[Muggleton, 1995] S. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[Muggleton, 2000] S. Muggleton. Learning Stochastic Logic Programs. In *AAAI 2000 Workshop on Learning Statististical Models from Relational Data*, Austin, Texas, USA, July 2000.

[Pawlak and Skowron, 1999] Z. Pawlak and A. Skowron. Rough set rudiments. In *Bulletin of International Rough Set Society*, volume 3. 1999.

[Pawlak, 1982] Z. Pawlak. Rough sets. *International Journal of Computer and Information Sciences*, 11(5):341–356, 1982.

[Pawlak, 1991] Z. Pawlak. *Rough Sets — Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.

[Siromoney and Inoue, 2000] A. Siromoney and K. Inoue. Elementary sets and declarative biases in a restricted gRS–ILP model. *Informatica*, 24:125–135, 2000.

[Siromoney and Inoue, 2002] A. Siromoney and K. Inoue. The generic Rough Set Inductive Logic Programming (gRS–ILP) model. In T. Y. Lin, Y. Y. Yao, and L. A. Zadeh, editors, *Data Mining, Rough Sets and Granular Computing*, volume 95, pages 499–517. Physica–Verlag, 2002.

[Siromoney, 1997] A. Siromoney. A rough set perspective of Inductive Logic Programming. In Luc De Raedt and Stephen Muggleton, editors, *Proceedings of the IJCAI-97 Workshop on Frontiers of Inductive Logic Programming*, pages 111–113, Nagoya, Japan, 1997.

[Srinivasan *et al.*, 1997a] A. Srinivasan, R.D. King, S.H. Muggleton, and M. Sternberg. The predictive toxicology evaluation challenge. In *Proceedings of the Fifteenth International Joint Conference Artificial Intelligence (IJCAI-97)*, pages 1–6. Morgan-Kaufmann, 1997.

[Srinivasan *et al.*, 1997b] A. Srinivasan, R.D. King, S.H. Muggleton, and M. Sternberg. Carcinogenesis predictions using ILP. In N. Lavrač and S. Džeroski, editors, *Proceedings of the Seventh International Workshop on Inductive Logic Programming*, pages 273–287. Springer-Verlag, Berlin, 1997. LNAI 1297.

[Ziarko, 1993] W. Ziarko. Variable precision rough set model. *Journal of Computer and System Sciences*, 46(1):39–59, 1993.