Areas of Intelligent Systems: Reasoning, Discretization, and Learning

K. Truemper
Professor Emeritus of Computer Science
University of Texas at Dallas

Topics:

Intelligent Systems

Definition: Intelligent System

A system is *intelligent* if it accomplishes feats that, when carried out by humans, require a substantial amount of intelligence.

Example Tasks

Medical diagnosis, processing of natural language, supervision of complex processes.

Definition: Expert System

An *expert system* is an intelligent system which in an interactive setting asks a person for information and, based upon the response, draws conclusions or gives advice.

Definition: Intelligent Agent

An *intelligent agent* is an intelligent system which perceives its environment by sensors and which uses that information to act upon the environment.

Observation

Expert systems and intelligent agents are special cases of intelligent systems.

An example of an intelligent system that is not an intelligent agent: an intelligent system that constructs expert systems from data. That intelligent system is not an intelligent agent, unless one accepts the odd notion that creating an expert system constitutes acting upon an environment.

Terminology

"Model": In propositional logic, a satisfying solution. Here, a mathematical formulation of a real-world situation.

"Valid": In propositional logic, a formula that always evaluates to *True*. Here, a formulation that correctly represents a part of the real world of interest.

Levels of Thinking, Informal Definition

First Level (= thinking about problems)

Direct reasoning about problems.

Typical questions:
"Do symptoms s and t imply that disease d or e is present?"
"Is this log-in behavior typical for a hacker?"

Second Level (= thinking about thinking)

We think about which logic module of the first level is to be used. May involve selection, modification, creation of module.

Third Level (= thinking about thinking about thinking)

We think about which process of the second level is to be used.

Reduction to thinking at lower level generally cannot be achieved by a polynomial algorithm. Note: "cannot" is meant pragmatically in the sense that presently nobody knows of a polynomial reduction.

Link to polynomial hierarchy of theory of computation: kth level of thinking corresponds to kth-level of hierarchy.

Quantified Boolean Formula (QBF)

a, b, c, $d, \ldots$ = vectors of Boolean variables
$\forall a \; \exists b \; \forall c \; \exists d \ldots \;\; D(a, b, c, d, \ldots)$

Each quantifier introduces another level.

Examples: First-Level Thinking

Theorem Proving

Deduction of theorems T from axioms of a propositional logic formula S.

Logic Minimization

Finding best satisfying solution for a CNF system propositional logic formula S. Accelerated theorem proving. Handling of special satisfiability cases.

Examples: Second-Level Thinking

Nonmonotonic Reasoning

Conclusion may become invalid when additional information is obtained. Axioms must be modified.

Reasoning with Incomplete Axioms

Desired conclusion cannot be derived. Axioms must be modified.

Discovery of Futility

Decide if desired conclusion cannot be derived even if all as-yet-unknown data are obtained.

Examples: Third-Level Thinking

Selection of Second-Level Reasoning

Decide which type of nonmonotonic reasoning or which discovery method of futility is to be used.

Question Selection

Questions are selected so that, as soon as possible, desired conclusion can be proved or shown to be futile.

Construction

Computations constructing intelligent systems that reason at the first and second levels.

Construction Methods

Direct Construction

Analyze problem. Formulate logic conditions and constraints. Define and implement all system operations.

Construction via Learning

Obtain data about problem. Derive logic conditions and constraints from the data. Use a metamethod to construct modules for all system operations.

An Extension of Propositional Logic

- Cost of *True* or *False* for a variable; call this Truecost and Falsecost.
- Representation of unknown values by two values: *Absent* and *Unavailable*.
- Likelihood of clauses being applicable.
- Predicates with finite quantifications $\exists$ (there exists) and $\forall$ (for all).
- Quantification of propositional variables. For example, for all *True/False* values of a subset of the variables, there exist *True/False* values for the remaining variables so that the formula has a specified value. Can be viewed as special case of predicate with finite quantification.

Examples

For variable x: Truecost $= 10$, Falsecost $= -5$

U, V finite sets

$\forall u \in U \ \forall v \in V \ [\neg g(u, v) \vee h(u, v)]$

Equivalent:

$\bigwedge_{\substack{u \in U \\ v \in V}} [\neg g(u, v) \vee h(u, v)]$

$x = \mathit{Absent}$: value of x unknown, but could be obtained

$x = \mathit{Unavailable}$: value of x cannot be obtained

Example of Quantification of Propositional Variables:

CNF formula R with vectors q and x of variables.

CNF formula S with vectors q and y of variables.

For all *True/False* values of the q_i of vector q:
if R is satisfiable, then S is satisfiable as well.

Write as: $\forall q \ (\exists x \ R \rightarrow \exists y \ S)$

First-Level Thinking

Problems SAT and MINSAT

We often use CNF *system* instead of CNF *formula*.

CNF system: list of variables; list of CNF clauses, each of which uses a subset of the variables.

Problem SAT

Instance: CNF system S.

Solution: Either: A satisfying solution of S. Or: The conclusion that S is not satisfiable.

Problem MINSAT

Instance: CNF system S. For each variable of S, two rational cost values associated with the values *True* and *False* for that variable.

Solution: Either: A satisfying solution of S for which the total cost is minimum. Or: The conclusion that S is unsatisfiable.

Second-Level Thinking

Nonmonotonicity

Deduction: Core approach of mathematics.

Most reasoning in everyday life is not deduction.

Examples

1. Operation of light bulb by switch.
2. Learning to drive by experimentation.

Mathematics: Use probability theory and induction to convert such reasoning to deduction.

Here: Want to use just one tool, an extension of propositional logic.

Monotonicity

When additional facts become available, previously proved theorems remain theorems.

Monotonicity of first-order logic is essential for mathematics.

Nonmonotonicity

Conclusion may become invalid by additional information.

Examples

1. A conclusion by abduction turns out to be in error.
2. Use of the value *Unavailable* in a Question-and-Answer process eliminates CNF clauses and thus may eliminate theorems.

Incompleteness

This is not a case of Gödel's Incompleteness Theorem (1931). We simply do not have the needed clauses.

Example

CNF system S

Statement $Y = \neg X \rightarrow R = X \vee R$ should be a theorem, but is not.

Thus, $S \wedge \neg Y = S \wedge \neg X \wedge \neg R$ is satisfiable, but should be unsatisfiable.

Solution: $S' = S \wedge Y$

New Problem

$S'' = S' \wedge \neg X = S \wedge Y \wedge \neg X$ may be unsatisfiable, but should be satisfiable. Must modify S' to eliminate this effect.

Problem Q-ALL SAT

Subproblem of Treating Futility of Questions in Expert System

Expert system asks for information to prove some result. If the result cannot be proved at all, the system should stop as soon as this fact becomes evident.

A small example . . .

Questions q_1, q_2, q_3. Defects r, s, t.
CNF system R has relationships connecting q_1, q_2, q_3.

$$q_2 \vee q_3$$

CNF system S links questions and defects

$$\begin{aligned} q_1 &\rightarrow (r \vee t) \\ q_2 &\rightarrow (\neg r \vee \neg t) \\ (q_1 \wedge q_2) &\rightarrow \neg r \\ q_3 &\rightarrow (s \vee \neg t) \end{aligned}$$

Suppose we want to prove defect t. We are given value $q_1 =$ *False*. It is easy to see that we cannot prove t with this information.

Problem: Can t be possibly proved by some values for q_2 and q_3?

If answer is "no", then it is useless to ask for the values of q_2 and q_3, and we should stop. Thus, proving t is *futile*.

Solution

CNF system R'

$$q_2 \vee q_3$$
$$\neg q_1$$

CNF system S'; enforces $\neg t$ for theorem proving.

$$\neg q_1 \vee r \vee t$$
$$\neg q_2 \vee \neg r \vee \neg t$$
$$\neg q_1 \vee \neg q_2 \vee \neg r$$
$$\neg q_3 \vee s \vee \neg t$$
$$\neg t$$

Proving t is futile if, for all *True/False* values of q_1, q_2, q_3 satisfying R, the CNF system S' is satisfiable.

Enumeration shows that proving t is indeed futile.

On the other hand, if originally $q_1 = \mathit{True}$, then there are values for q_2, q_3 so that t can be proved; take $q_2 = \mathit{True}$.

Definition: An assignment of *True/False* values to $q_1, \ldots, q_l$ is *R-acceptable* if the SAT instance of R with $q_1, \ldots, q_l$ fixed according to the given assignment is satisfiable.

Problem Q-ALL SAT

Instance: Satisfiable CNF systems R and S that have $l \geq 1$ special variables $q_1, \ldots, q_l$ among their common variables.

Solution: Either: "All R-acceptable assignments of *True/False* values for $q_1, \ldots, q_l$ are S-acceptable." Or: An R-acceptable assignment of *True/False* values for $q_1, \ldots, q_l$ that is S-unacceptable.

Polynomial Hierarchy

a, b, c, d, $\ldots$ = vectors of Boolean variables
$\forall a\ \exists b\ \forall c\ \exists d \ldots\ D(a, b, c, d, \ldots)$

Here: CNF systems R and S

Vectors q, x, y

$\forall q\ [\exists x\ R \rightarrow \exists y\ S]$

Transformation to standard form is trivial. However, even the fastest present-day algorithms for the standard case do not perform well on this particular problem, since the structure is not exploited.

Exact Algorithm for Q-ALL SAT

Ongoing work with A. Remshagen. Recursively fixes q_k variables and tries out effect. Learns clauses and predicts SAT behavior to reduce size of search tree, using an extension of the heuristic algorithm given later. As an aside, the algorithm proves some special cases to be in $\mathcal{NP}$.

Recursive Step

QRSsat(R,S,Q)

1. Do unit-resolution in R on all variables. All Q variables fixed in R by unit-resolution are fixed in S as well.
2. Do unit-resolution in S on the Y variables only.
3. If ($Q = \emptyset$)
   if (R is satisfiable and S is unsatisfiable)
   Return *True*;
   Else
   Return *False*;
4. Select a Q variable q;

5. If (QRSsat($Q \setminus \{q\}, R(q = \mathit{True}), S(q = \mathit{True})) = \mathit{True}$)
   Return *True*;
   Else if (QRSsat($Q \setminus \{q\}, R(q = \mathit{False}), S(q = \mathit{False})) = \mathit{True}$)
   Return *True*;
   Else
   Return *False*;

Learn Clauses and Make SAT Predictions

1. Algorithm learns clauses before backtracking from a given fixing. Main idea: Unfix Q variables while showing that the same conclusion is still valid. The remaining fixed variables produce a clause that excludes that fixing. The learned clauses are added to R. An enhanced version of the next heuristic is used for the learning of the clauses.

2. Predict SAT behavior of S using values of fixed Q and Y, plus a greedy heuristic. Use predictions for backtracking and for selection process of next Q variable to be fixed.

Heuristic Algorithm for Q-ALL SAT

Key ideas

1. If deletion of all free q_k reduces S to a satisfiable S': Fixing of any q_k cannot produce unsatisfiability. Hence, the Q-ALL SAT instance has been solved.
2. (S' is unsatisfiable) Compute a minimal unsatisfiable subset of clauses of S'. Let $\overline{S}$ be corresponding subsystem of S.

   Check if q_k values exist that satisfy R while causing all literals of the q_k in $\overline{S}$ to evaluate to *False*.

   If such q_k exist, then the Q-ALL SAT instance has been solved.

Problem Q-ALL SAT: QRSsat3 Test Results

Test Sets

1. Robot navigation through grid with obstacles.
   Question: Can obstacles be so placed that they block the robot?
   18 instances.

2. Game tree where first player tries to prevent opponent from reaching goal.
   Question: Can first player achieve goal?
   12 sets of 12 instances each.

	$\|Q\|$	$\|X\|$	$\|Y\|$
Robot	64-100	64-100	128-200
Game	15-25	0	275-405

	$\#R$ clauses	$\#S$ clauses
Robot	925-1683	596-1125
Game	1	781-841

Computational Results

	Solution Times (sec)[1]			
	QRSsat3	yQuaffle	QuBERel1.3	Semprop
Robot	15	2,342[2]	3,231	54,090[3]
Game	12	5,197[4]	6,292[5]	6,135[6]

(1) Robot case: Each figure is total time (sec) for 18 instances. Game case: Each figure is average time per set of 12 instances.
(2) Error termination in 4 instances.
(3) 1 hr time limit exceeded in 15 out of 18 instances. Used 3,600 sec for these cases in the statistic.
(4) 1 hr limit exceeded for 14 instances.
(5) 1 hr limit exceeded for 4 instances. Error termination for 10 instances.
(6) 1 hr limit exceeded for 10 instances.

Conclusion

Robot instances: QRSsat3 at least 200 times faster.
Game instances: QRSsat3 at least 400 times faster.

Optimization Versions of Q-ALL SAT

Problem Q-MINFIX UNSAT

Subproblem of Learning to Ask Relevant Questions

Problem Q-MINFIX UNSAT

Instance: Satisfiable CNF system S containing $l \geq 1$ special variables $q_1, \ldots, q_l$. An S-unacceptable assignment. For each q_j, a cost of obtaining the given *True/False* value of q_j.

Solution: An S-unacceptable partial assignment with minimum total cost.

Variation

Must carry out tests with given costs to get sets of values.

Application

In medical diagnostic system, have obtained symptom values q_1, $q_2, \ldots,$ q_l by some tests, and have proved some disease t to be present.

Want to find out in hindsight which tests should have been done to get values for some of the variables $q_1, \ldots,$ q_l so that the disease could have been proved at least total cost.

Problem Q-MINFIX SAT

Subproblem of Learning to Ask Relevant Questions

Problem Q-MINFIX SAT

Instance: Satisfiable CNF systems R and S having $l \geq 1$ variables $q_1, \ldots, q_l$ among their common variables. An assignment for $q_1, \ldots, q_l$ that is both R-acceptable and S-acceptable. For each q_j, a cost of obtaining the given *True/False* value for q_j.

Solution: A partial assignment such that each R-acceptable extension assignment is S-acceptable. Subject to that condition, the total cost of the partial assignment must be minimum.

Variation

Must carry out tests with given costs to get sets of values.

Application

In medical diagnostic system, have obtained symptom values q_1, $q_2, \ldots,$ q_l by some tests, and have determined that some disease t cannot be proved.

Want to find out in hindsight which tests should have been done to get values for some of the variables $q_1, \ldots, q_l$ so that, at least total costs, the same conclusion could have been obtained.

Separations

Separations

Training Sets

Training sets A and B consist of records of length n. The kth entry is the value for attribute k. Entries may be *True*, *False*, or *Unavailable*.

The value *Unavailable* means that one cannot get a *True/False* value.

Note: Generally, another value is possible: *Absent*. In that case, one does not know value, but can obtain it. In this talk, we will not make use of that option.

Separation Problem

Find a logic formula that is *True* on A and *False* on B, or show that this cannot be done. The formula *separates* A from B.

Populations

The sets A and B usually are taken from two populations $\mathcal{A}$ and $\mathcal{B}$. A separating formula for A and B may then be used to predict whether a record is in $\mathcal{A}$ or $\mathcal{B}$.

Finding Separating Formulas

There are effective methods to find separating formulas. In our approach, the formulas are in disjunctive normal form (DNF), and we construct them by a recursive process.

Example: $(x \wedge \neg y \wedge z) \vee (\neg x \wedge v) \vee (y \wedge w)$

Discretization

Discretization

Given: Numerical data sets A and B.

Goal: Logic data sets A' and B' representing A and B.

Most popular method: Entropy plus Minimum Description Length Principle (MDLP). The principle generally selects the hypothesis that minimizes the description length of the hypothesis plus the description length of the data given the hypothesis.

Here, we apply a new method of pattern analysis.

Processing of one Attribute

1. Sort the numerical entries of the attribute. Label each entry of the sorted list by "A" (resp. "B") if coming from a record of A (resp. B). The result is a *label sequence*.

   Example:

10.8	3.7	2.9	1.7	0.5	−1.0	−3.5	−11.9
A	A	A	B	A	B	B	B

3. Find an interval where the sequence switches from mostly As to mostly Bs. The more rapid the switch, the more important the interval. Put a cutpoint c into the middle of the interval. Details are given in a moment.

   Logic attribute y_c: if $x \leq c$, then $y_c = \mathit{False}$
   if $x > c$, then $y_c = \mathit{True}$

Details

1. In the label sequence, replace each A by 1 and each B by 0.
2. Smooth the sequence of 0s and 1s by Gaussian convolution. The variance of Gaussian convolution is determined by evaluation of a *competing random process*. Goal is smoothing so that randomly introduce differences are eliminated.
3. Select cutpoint where the smoothed data change by maximum difference.

Example

$$\begin{array}{cccccccc} 10.8 & 3.7 & 2.9 & 1.7 & 0.5 & -1.0 & -3.5 & -11.9 \\ A & A & A & B & A & B & B & B \end{array}$$

$$\uparrow$$

$$\text{cutpoint } c = (1.7 + 0.5)/2 = 1.1$$

Important Factors

Important Factors

Discussion via example of cancer data.

Cervical Cancer: FIGO I-III versus Recurrence

Goal

Derive factors explaining difference between FIGO I-III and Recurrence, using lab data.

Sketch of Method

1. Partition the data into FIGO I-III cases and Recurrence cases.
2. Discretize the two data sets, getting sets A for FIGO I-III and B for Recurrence.

3. Compute a separating logic formula achieving *True* for A and *False* for B.

   For each literal (= occurrence of a possibly negated variable) of the formula, count the number of records for which the literal is needed to achieve *True* for the formula. The count is the *importance value* of the literal for the formula.

   Repeat the above step, but exchange the roles of A and B.

4. For each literal l, normalize the two importance values using the number of records of A or B, whichever applies. Thus, get average importance values $\overline{f}_{X,l}$ for $X = A$ and B and for all l.

5. The variables for which at least one of the two possible literals l has $\overline{f}_{X,l} \geq 0.4$, are the *important factors*.

Explanations

Explanations

Discussion uses example of FIGO I-III versus Recurrence.

1. Delete from the data all attributes except for the important factors determined in the previous step.
2. Discretize the sets of FIGO I-III and Recurrence records using the important factors.
3. Compute a logic formula that is *True* on A and *False* on B, and a second formula that is *False* on A and *True* on B.

   The logic formulas, combined with the discretization information, constitute the desired explanations.

Statistical Significance

Statistical Significance

Approach

We define a statistic with 0/1 value via the explanations obtained in the previous step.

Hypothesis H_0: The explanations produce accurate predictions.

Hypothesis H_1: The explanations do not produce accurate predictions. Indeed, with some luck, the same accuracy can be achieved by flipping an unbiased coin, which statistically is a Bernoulli trial with $\alpha = 0.5$.

Procedure: Find Explanations and Establish Significance

1. Split the given data into a training set and a testing set.
2. Obtain explanations from the training data using the earlier described process.
3. Apply the explanations to the testing data and determine how often the explanations are correct/incorrect.
4. Compare the outcome of Step 3 with results of Bernoulli trials with $\alpha = 0.5$.
5. Use direct computation or approximation by normal distribution to obtain probability p that Bernoulli trials obtain the same results or better. If p is very small, then accept H_0. Otherwise, accept H_1.

Explanation Examples

Explanation Examples: Breast Cancer

The data sets used in this subsection were supplied by the Frauenklinik of the Ruhr-Universität Bochum.

1. Herceptin/Xeloda Study

14 patients

Records have 38 attributes:

SURVZ (living at present)
HER2 (value of second test)
Thymidinphosphorylase:
  TP_TUMOR
  TP_TISSUE
  TP_OVERALL
VEGF (Vascular Endothelial Growth Factor)
COX2 (Cyclooxygenase 2)
K18 (Keratin 18)

HAT_AD_MED (adjuvant hormone therapy: medication
(1 = tamoxifen,2 = GnRH-Analogon,
3 = Aromatase Inhibitor,4 = other,?))
HER2_STAT (HER2 status (2, 3, ?))
FISH_STAT (FISH Status (0,1))
HISTO_TYP (histological tumor type (1,2,3))
PT (tumor size (1,2,3,4,?))
PN (lymphnode status (1,2,3,?))
M (metastasis (0,1,?))
G (grading (1,2,3,?))
REZ_ER (estrogen receptor expression (0,1,?))
REZ_PR (progesterone receptor expression (0,1,?))
Local recurrence and distant metastasis status:
AT_LOK (local)
AT_ABD (abdominal)
AT_HEP (liver)
AT_PUL (lungs)
AT_ZNS (central nervous system)

AT_PERI (heart)
AT_PLEU (pleura)
AT_ASCI (ascites)
AT_LYM (lymphangiosis)
AT_KNO (bone)
HT_AD (hormone therapy)
HT_PA (palliative hormone therapy)
CT_AD (chemotherapy)
CT_PA (palliative chemotherapy)
ST (radiation)
BT (bisphosphonates)
Age (years)
BEST_RES (best response
(1 = complete response,2 = partial response,
3 = no change,4 = progressive disease))
TTP (time to progression (weeks))
SURV (survival time (weeks))

Question

Which factors influence time to progression (TTP)?

Answer

The most important factors influencing TTP are:

Attribute	Importance Value*
TP_TISSUE	0.576
K18	0.215
HER2	0.229
COX2	0.206

*Note: Values computed by initial method for importance factors. Threshold for that method was 0.2.

High TTP Cases

The data contain three patients with amazingly high TTP values. Analysis of reasons produces the follow explanation:

In two situations, high TTP values are predicted:

Case 1: TP_TISSUE ≥ 6 and K18 ≥ 9

Case 2: TP_TISSUE ≥ 6 and COX2 ≤ 2

2. Local Recurrence Versus Metastasis

Explain the difference between local recurrence and metastasis using lab data.

Data

15 patients (3 local recurrence, 12 metastasis)

Records have 17 attributes.

GRADING
T
N
M
KI67
HER2NEU
ER
K18

PLAKO
PL_M
PL_Z
PL_K
DESMO
DESMO1
DE_M
DE_Z
DE_K

Important Factors

Attribute	Importance Value
PL_M	0.900
K18	0.733

Explanation

The difference between local recurrence and metastasis can be explained as follows.

If K18 $\geq$ 3.0 and PL_M $\geq$ 2.0,
then local recurrence case.

If K18 $\leq$ 2.0 or PL_M $\leq$ 1.0,
then metastasis case.

Explanation Examples: Cervical Cancer

The data sets used in this subsection were supplied by the Frauenklinik of the Charité, Berlin.

1. Factors Influencing Time to Progression (TTP)

11 patients

Records have 20 attributes:

RESP (responder (0 = no, 1 = yes))
AGE (years)
PARTUS (number of born children)
MENOP (menopause (0 = no, 1 = yes))
T (Tumor size (FIGO))
N (Lymphnode metastasis (0 = no, 1 = yes))

HISTO (Histology
(0= poorly diff. squamous cell carcinoma,
1= moderately diff. squamous cell carcinoma,
2= poorly diff. Adeno-Ca,
3= SCC))
TTP (time to progression (months))
SCC_0 (squamous cell carcinoma antigen, baseline)
SCC_4 (after 4 weeks chemotherapy)

SVEGF_0 (SVEGF-A = vascular endothelial growth factor A in pg/ml in serum, baseline)
SVEGF_1 (after 1st cycle 4th dose)

PVEGF_0 (PVEGF-A = vascular endothelial growth factor A in pg/ml in Plasma, baseline)
PVEGF_1 (after 1st cycle 4th dose)

SVEGF_D_0 (SVEGF-D = vascular endothelial growth factor D in pg/ml in Serum, baseline)
SVEGF_D_1 (after 1st cycle 4th dose)

EGF_0 (epidermal growth factor in pg/ml, baseline)
EGF_1 (after 1st cycle 4th dose)

IGF_0 (IGF = insulin like growth 1 factor in ng/ml, baseline)
IGF_1 (after 1st cycle 4th dose)

Question

Which factors influence time to progression (TTP)?

Answer

Two cases: (1) Using initial data only. (2) Using initial and subsequent data.

Both cases result in the same most-important attributes:

Attribute	Importance Value*
SVEGF_D__0	0.603
MENOP	0.400
PARTUS	0.247

*Note: Values computed by initial method for importance factors. Threshold for that method was 0.2.

High TTP

High TTP values (TTP ≥ 39) are predicted if

MENOP $= 1$ and SVEGF_D_0 ≥ 357

Note: We also have PARTUS ≥ 1.

Low TTP

Low TTP values (TTP ≤ 19) are predicted if

MENOP $= 0$ or SVEGF_D_0 < 357

Note: In the case of SVEGF_D_0 < 357, we also have PARTUS $= 0$.

2. Difference Between FIGO I-III and Recurrence

Note: At present, treatments cannot utilize this information. We include it here to demonstrate validation.

57 patients (31 for training, 26 for testing)

31 training cases: 19 FIGO I-III, 12 Recurrence

26 testing cases: 14 FIGO I-III, 12 Recurrence

Note: FIGO IV excluded since too few cases.

Records have 14 attributes:

Attribute	Uncertainty Interval
VEGF_PLASMA	[74.30 , 97.30]
VEGFD_SERUM	[381.00 , 441.00]
VEGFC_SERUM	[8455.00 , 9416.00]
ENDOGLIN	[4.06 , 4.63]

ENDOSTATIN	[123.00, 136.00]
ANGIOGENIN	[335.00 , 364.00]
FGFB_SERUM	[5.10 , 8.50]
VEGFR1_SERUM	[74.50 , 80.00]
VEGFR2_SERUM	[10995.00 , 11114.00]
M2PK_PLASMA	[20.80 , 21.80]
SICAM1_SERUM	[325.00 , 344.00]
SVCAM1_SERUM	[624.00 , 635.00]
IGFI_SERUM	[113.00 , 122.00]
IGFBP3_SERUM	[2552.00 , 2592.00]

Important Factors

Attribute	Importance Value
M2PK_PLASMA	0.467
ENDOSTATIN	0.400

Result

If ENDOSTATIN < 123.0 or M2PK_PLASMA < 18.8,
then FIGO I-III case.

If ENDOSTATIN > 123.0 and M2PK_PLASMA > 21.8,
then Recurrence case.

Accuracy of Prediction

22 of 26 cases are predicted correctly. Accuracy = 85%.

Significance

Significance of conclusion is $p < 0.0002$.

References

References

Books

1. K. Truemper, “Effective Logic Computation” (Wiley, 1998).
2. K. Truemper, “Design of Logic-based Intelligent Systems” (Wiley, 2004).

Book Chapters

Book “Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques”, E. Triantaphyllou and G. Felici, eds., Springer Verlag, Berlin, 2006:

1. “Learning Logic Formulas and Related Error Distributions” (coauthored, G. Felici, F. Sun, and K. Truemper).
2. ”Learning to Find Context-based Spelling Errors”, (coauthored, H. Almubaid and K. Truemper).

3. "Transformation of Rational and Set Data to Logic Data" (coauthored, S. Bartnikowski, M. Granberry, J. Mugan, and K. Truemper).

Technical Paper

"A MINSAT approach for learning in logic domains," (coauthored, G. Felici and K. Truemper), INFORMS *Journal on Computing* 14 (2002) 20–36.

Software

Leibniz System, Version 9.0, Leibniz Company, Plano, Texas.

Web

www.utdallas.edu/~klaus

www.bioxsys.com