

# MINING RULES FROM MONOTONE CLASSIFICATION MEASURING IMPACT OF INFORMATION SYSTEMS ON BUSINESS COMPETITIVENESS

---

Tomáš Horváth, František Sudzina, Peter Vojtáš  
Faculty of Science, University Of Pavol Jozef Šafárik in Košice  
thorvath@science.upjs.sk  
sudzina@euke.sk  
vojtas@kosice.upjs.sk  
SLOVAKIA

*Motivation for this paper are classification problems in which data can not be clearly divided into positive and negative examples, especially data in which there is a monotone hierarchy (degree, preference) of more or less positive (negative) examples.*

*We use data expressing the impact of information systems on business competitiveness in a graded way. The research was conducted on a sample of more than 200 Slovak companies. Competitiveness is estimated by Porter's model.*

*The induction is achieved via multiple use of two valued induction on alpha-cuts of graded examples with monotonicity axioms in background knowledge. We present results of ILP system ALEPH on above data interpreted as annotated rules. We comment on relations of our results to some statistical models.*

## 1. INTRODUCTION

There are many theoretical articles, which argue that usage of information systems increases business competitiveness. But only a few give proofs. Our data were gathered from a sample of 206 Slovak companies, which had to submit their preference (grade) of business competitiveness and information systems usage. These data are monotonous that means that if the company is highly competitive (the grade "best") then its competitiveness is also "medium" or "low".

In the crisp framework we are restricted only to the information that is true absolutely. Since we have uncertain or noisy data, this framework is not suitable to represent this kind of information. This is a significant gap in the expressive power of the framework, and a major barrier to its use in many real-world applications. Besides probabilistic models there is an extensive study of these phenomena in manyvalued logic, especially in fuzzy logic and generalised annotated programs.

Inductive logic programming is an effective tool for the data mining in the case of non numeric data. Information is implemented in the language of predicate logic, since it is easy to understand. Unlike many data mining tools, inductive logic programming is effective for the multi relational problems, too.

After explaining data, we present a new formulation of a many-valued inductive logic programming task in the framework of fuzzy logic in narrow sense. Our construction is based on a syntactical equivalence of fuzzy logic programs FLP and a restricted class of generalised annotated programs. The induction is achieved via multiple use of classical two valued inductive logic programming. Correctness of our method (translation) is based on the correctness of FLP. The cover relation is based on fuzzy Datalog and fixpoint semantics for FLP. We present and discuss results of ILP system ALEPH on our data. Then we compare our results with a statistical method of linear regression.

The information systems usage in 206 Slovak companies was analyzed from the point of view of the most basic model, which divides information systems into enterprise resource planning (ERP) systems, management information systems (MIS) and executive information systems (EIS) (Kokles, Romanová, 2002) and (Dudinská, Mizla, 1994). Information systems cover five main areas - sales and marketing, manufacturing, finance, accounting, human resources; therefore ERP systems were divided into five categories. Data on accounting systems were not used in later analysis because all companies must do accounting therefore it cannot be the factor that might influence business competitiveness. The reason why there are companies, which do not have any accounting system, is that they outsource accounting.

We asked if the company used an information system supporting specific areas and if so, we asked how was the company satisfied with the information system. IS satisfaction was measured on a Likert-type scale 1-7 (self-assessment).

The extent of outsourcing of information systems was quite significant; therefore data on outsourcing were also used as explanatory variables. Data on outsourcing of information systems do not include outsourcing of accounting systems because managers do not consider it to be outsourcing of an accounting system but outsourcing of accounting as of a functional area.

The company size was graded. We used the same 7 intervals, which are used by the Statistical office of the Slovak Republic.

To sum up, explanatory variables include nine columns - on ERP - sales and marketing, ERP - manufacturing, ERP - finance, ERP - human resources management, MIS, EIS, partial outsourcing, total outsourcing and company size.

(Porter, 1979) suggested to analyze the value chain, all the business processes that contribute to the value added. As the main processes he identified input logistics, manufacturing, output logistics, marketing and sales, services; subsidiary processes include administration, human resource management, technological development, buying. We merged manufacturing and services in order to meet the main processes of all sectors in one value. We disagree that marketing (in contrast with selling) is a main process; therefore we included it in subsidiary processes. Each company was asked to provide self-assessment of its competitiveness in all of the dimensions. A Likert-type scale 1-7 was used to measure the competitiveness (self-assessment).

So, the dependent variables are input logistics, manufacturing and services, output logistics, sales, administration, human resource management, technological development, buying and marketing.

## 2. A MONOTONE GRADED ILP PROBLEM

In this section we introduce a problem of the monotone graded inductive logic programming ILP (Horváth, Lencses, Krajči, Vojtáš, 2004). We describe the problem of generalized annotated programs GAP (Kiefer, Subrahmanian, 1992), which herbrand interpretations coincides with interpretations of FLP (Krajči, Lencses, Vojtáš). After we describe our method for a multiple used classical (crisp) ILP systems to solve a monotone graded ILP problem. Our method is based on the correctness of FLP (Vojtáš, 2001).

### 2.1 Generalized annotated programs

Kifer and Subrahmanian (Kiefer, Subrahmanian, 1992) introduced generalized annotated logic programs (GAP) that unify and generalize various results and treatments of multi-valued logic programming. The whole theory of GAP is developed in a general setting for lattices. We restrict ourselves to the unit interval of real numbers  $[0,1]$ .

In fuzzy logic programming rules had weights (or truth values) associated with them as a whole. Annotated logic, on the other hand, appeared to associate truth values with each component of an implication rather than the implication as a whole. This implication is interpreted in a “classical logic“ fashion. We show how truth values in FLP can be propagated across implications to derive clauses in GAP.

**Definition** A function  $A: [0,1]^i \rightarrow [0,1]$  is an annotation function if it is left continuous and order preserving in all variables.

The language of annotated programs consists of a usual language of predicate logic (with object variables, constants, predicates and function symbols) as in FLP and of the quantitative part of the language. The quantitative part of the language has annotation variables and a set of basic annotation terms of different arity. Every annotation term  $\rho$  is a composition of annotation functions. Notice, that  $\rho \bullet c$  can be considered as the truth function of an aggregation operator.

**Definition** If  $A$  is an atomic formula and  $\alpha$  is an annotation term, then  $A : \alpha$  is an annotated atom. If  $A : \rho$  is a possibly complex annotated atom and  $B_1 : \mu_1, \dots, B_k : \mu_k$  are variable- annotated atoms, then  $A : \rho \leftarrow B_1 : \mu_1 \ \& \ \dots \ \& \ B_k : \mu_k$  is an annotated clause. We assume that variables occurring in the annotation of the head also appear as annotations of the body literals and different literals in the body are annotated with different variables.

**Definition** Let  $B_L$  be the Herbrand base. A mapping  $f: B_L \rightarrow [0,1]$  is said to be a Herbrand interpretation for annotated logic.

Note that interpretation for fuzzy logic and interpretations for annotated logic coincide.

Suppose  $I$  is an Herbrand interpretation. Then,

$I$  satisfies a ground atom  $A : \mu$  iff  $\mu \leq I(A)$

$I$  satisfies  $(F \wedge G)$  iff  $I$  satisfies  $F$  and  $I$  satisfies  $G$

(please note that iff and are metamathematical two valued connectives)

$I$  satisfies  $(F \vee G)$  iff  $I$  satisfies  $F$  or  $I$  satisfies  $G$

$I$  satisfies  $F \leftarrow G$  iff  $I$  satisfies  $F$  or  $I$  does not satisfy  $G$ .

**Definition** (FLP and GAP transformations). Assume  $C = A : \rho \leftarrow B_1 : \mu_1 \& \dots \& B_k : \mu_k$  is an annotated clause. Then  $\text{flp}(C)$  is the fuzzy rule  $A \leftarrow \rho(B_1, \dots, B_k)$ . Here  $\rho$  is understood as an  $n$ -ary aggregator operator.

Assume  $D = A \leftarrow @ (B_1, \dots, B_n), r$  is a fuzzy logic program rule. Then  $\text{gap}(D)$  is the annotated clause  $A : C_i (@^*(x_1, \dots, x_n), r) \leftarrow B_1 : x_1 \& \dots \& B_k : x_n$ .

The satisfaction is defined differently (all variables (object and annotation) are implicitly universally quantified).

**Theorem** (Vojtáš, 2001) Assume  $C$  is an annotated clause,  $D$  is a fuzzy logic program rule and  $f$  is a fuzzy Herbrand interpretation. Then

$f$  is a model of  $C$  iff  $f$  is a model of  $\text{flp}(C)$

$f$  is a model of  $D$  iff  $f$  is a model of  $\text{gap}(D)$

This theorem is the main tool in our formal model of fuzzy ILP.

## 2.2 ILP system ALEPH

Since our aim is not to develop a new resp. better ILP algorithm we will not describe the used ILP systems in details – we notice just some basic properties of these systems (we are interested just in the correct transfer of the graded ILP problem to a multiple use of classical – crisp – ILP problem). In a two valued logic the Inductive logic programming (ILP) task is formulated as follows:

In ILP, given is a set of examples  $E = E^+ \cup E^-$ , where  $E^+$  contains positive and  $E^-$  negative examples, and background knowledge  $B$ . The task is to find a hypothesis  $H$  such that  $\forall e \in E^+ : B \wedge H \models e$  ( $H$  is complete) and  $\forall e \in E^- : B \wedge H \not\models e$  ( $H$  is consistent). This setting, introduced in (Muggleton, 1991), is also called learning from entailment. In an alternative setting proposed in (Džeroski, Lavrač, 2001), the requirement that  $B \wedge H \models e$  is replaced by the requirement that  $H$  be true in the minimal Herbrand model of  $B \wedge e$ : this setting is called learning from interpretations. We will follow this in our formal model.

In order to search the space of relational rules (program clauses) systematically, it is useful to impose some structure upon it, e.g. an ordering. One such ordering is based on subsumption (clause  $C$  subsumes  $C'$  if there exist a substitution  $\theta$ , such that  $C\theta \subseteq C'$ ). Notice, that if  $C$  subsumes  $D$  then  $C \models D$ . The converse always not hold. Notice, that the space of clauses ordered by the subsumption is a lattice.

The ILP system ALEPH (Srinivasan, 2000, Aleph) is based on inverse entailment (Muggleton, 1995). For a given background knowledge  $B$  and examples  $E$  and the hypothesis  $H$  it must hold, that  $(B \wedge H) \models E$ . If we rearrange the above using the law of contraposition we get the more suitable form  $(B \wedge \neg E) \models \neg H$ . In general  $B$ ,  $H$  and  $E$  can be arbitrary logic programs but if we restrict  $H$  and  $E$  to being single Horn clauses,  $\neg H$  and  $\neg E$  above will be ground skolemised unit clauses. If  $\perp$  is the conjunction of ground literals which are true in all models of  $B \wedge \neg E$  we have  $(B \wedge \neg E) \models \perp$ . Since  $\neg H$  must be true in every model of  $B \wedge \neg E$  it must contain a subset of the ground literals in  $\perp$ . Hence  $(B \wedge \neg E) \models \perp \models \neg H$  and so  $H \models \perp$ .

The complete set of candidates for  $H$  could in theory be found from those clauses which imply  $\perp$ . A subset of the solutions for  $H$  can then be found by considering

those clauses which subsume  $\perp$ . ALEPH searches the latter subset of solutions for H that subsume  $\perp$ .  $\perp$  is called saturation of example.

### 2.3 A monotone graded ILP problem

In a monotone graded ILP problem (Horváth, Lencses, Krajči, Vojtáš, 2004) data are not clearly divided into positive and negative examples, i.e. there is a monotone hierarchy (degree, preference) of more or less positive (negative) examples. This corresponds to fuzzy set of examples. We assume also on the side of background knowledge a monotone graded (comparative) notion of fulfilment. This corresponds to fuzzy background knowledge in the form of a definite logic program (without negation). We expect to be able to extract rules of the form

IF the satisfaction with ERP-human resources is at least 4 (or better 5, 6, 7)  
 AND the company size is at least 6 (more than 500 employees)  
 THEN the competitiveness in administration is at least 4 (or better 5, 6, 7)

Notice that we assume a positive (monotonic, increasing) influence of background factors on the degree of classification (understood in a monotonic way).

We transfer the problem of graded ILP with fuzzy (graded) background knowledge and fuzzy set of examples (graded examples) to several crisp ILP problems, so that  $c(B)$  is the knowledge acquired from B by adding an additional attribute for the truth value and for every  $\alpha \in [0, 1]$   $E_\alpha^+$  and  $E_\alpha^-$  are cuts of the fuzzy set E. The fuzzy hypothesis  $H \upharpoonright H_\alpha \equiv \alpha$ .

Problem is that the system means the numbers like a syntactic objects and it do not distinguish the ordering between them. Therefore we must define this ordering in the background knowledge – background knowledge of ALEPH can contain rules. Since the truth value (TV) of the atoms in the background knowledge/hypotheses determines the maximum/minimum degree of compatibility (“at most”/“at least”) it is convenient to define for every graded (fuzzy) predicate in the background knowledge a rule

predicate( $X_1, X_2, \dots, X_n, TV_a$ ) :-  $TV_a < TV_b$ , predicate( $X_1, X_2, \dots, X_n, TV_b$ ).

This rule we rewrite to a

predicate( $X_1, X_2, \dots, X_n, TV_a$ ) :- leq( $TV_a, TV_b$ ) predicate( $X_1, X_2, \dots, X_n, TV_b$ ),  
 where leq( $TV_a, TV_b$ ) evaluates the relation „ $TV_a$  is less or equal than  $TV_b$ “.  
 Hereby, we add ground atoms leq( $TV_1, TV_2$ ), ..., leq( $TV_{n-2}, TV_{n-1}$ ) such that for  $i < j$  holds  $TV_i < TV_j$ , and  $TV_1$  is more than the lowest grade ( $TV_{min}$ ), while  $TV_{n-1}$  is the greatest grade ( $TV_{max}$ ). We do this, because we need to generate the truth values (for the saturation) and not to compare them. In our case we add to the background knowledge following facts leq(1,2). ... leq(5,6). leq(6,7). and for every graded background knowledge predicate rule like  
 sales\_marketing(A,C) :- leq(C,D), sales\_marketing(A,D).  
 human\_resources(A,C) :- leq(C,D), human\_resources(A,D). , etc.  
 Except these, the background knowledge contains facts (for every attribute and

object) like `human_resources(object3,7)`, which means, that the company no. 3 is satisfied at the grade 7 with software for `human_resources`.

For example, the saturation of one example looks like

`administration(A) :- company_size(A,4), company_size(A,3), company_size(A,2), company_size(A,1), manufacturing(A,6), manufacturing(A,5), manufacturing(A,4), manufacturing(A,3), manufacturing(A,2), manufacturing(A,1), finance(A,5), finance(A,4), finance(A,3), finance(A,2), finance(A,1).`

One of the assets of this method is that we can define the ordering. We must tell, that in this case except the rules the background knowledge consist similar predicates (similar arity and the domain of attributes), but ILP works effectively in the case of complicated background knowledge, too.

The rules in the result hypothesis must subsume the saturations of some (all) positive and must not subsume the saturations of any negative examples. Some rules from the hypotheses evaluated by expert:

At the grade 4

`marketing(A) :- sales_marketing(A,7), human_resources(A,1).`

At the grade 5

`marketing(A) :- sales_marketing(A,4), finance(A,4), human_resources(A,7).`

`buying(A) :- manufacturing(A,4), finance(A,7).`

`buying(A) :- sales_marketing(A,7), finance(A,7).`

`sales(A) :- sales_marketing(A,6), manufacturing(A,6), finance(A,5).`

`sales(A) :- company_size(A,2), manufacturing(A,1), human_resources(A,7).`

At the grade 6

`sales(A) :- sales_marketing(A,4), finance(A,6), human_resources(A,7).`

**Glueing hypotheses together** Moreover rule obtained on the level  $\alpha$  guarantees the result in degree  $\alpha$ , so it corresponds to a fuzzy logic program rule with truth value  $\alpha$  (because in body there are crisp predicates and the boundary condition of our conjunctors fulfil  $C(x; 1) = x$ ).

The first rule corresponds to fuzzy rule

`(marketing(A) :- sales_marketing(A,7), human_resources(A,1)).4`

The second rule says

`(marketing(A) :- sales_marketing(A,4), finance(A,4), human_resources(A,7)).5`

and so on.

Here we see limitations of fuzzy logic programming in the induction, we are not able to glue them to one hypothesis. On the other side, these rules define a single annotation term for every predicate "p" in the heads of rules - a function of 9 real variables (body can contain 9 atoms) -  $a_p(x_1, x_2, \dots, x_9)$ .

If there is no such rule then the function is the smallest monotone function extending those points, i.e.

$$a_p(x_1, x_2, \dots, x_9) = \max \{a_p(y_1, \dots, y_9) : y_i \leq x_i \text{ for every } i=1, \dots, 9\}$$

For example, if the system for the predicate „sales“ at grade  $\alpha$  has induced the rule `sales(A) :- company_size(A,  $x_1$ ), sales_marketing(A,  $x_2$ ), manufacturing(A,  $x_3$ ), finance(A,  $x_4$ ), human_resources(A,  $x_5$ ), mis(A,  $x_6$ ), eis(A,  $x_7$ ), partial_outsourcing(A,  $x_8$ ), total_outsourcing(A,  $x_9$ ).`, then  $a_{\text{sales}}(x_1, x_2, \dots, x_9) = \alpha$ .

Another challenging problem is to learn the function  $a_p$ , methods of (Železný, 2001) could be appropriate.

### 3. RESULTS AND CONCLUSION

Figure 1 represents how well can be the impact of information systems on main processes identified by linear regression and by ILP. Regression is evaluated by the coefficient of determination ( $R^2$ ) because it represents the ratio of explained dispersion. Other seven bars represent the ratio of correctly classified positive instances (examples) to all positive instances for a certain grade ( $\alpha = 1, 2, \dots, 7$ ) of competitiveness. Both scales are ratio scales and can be well interpreted.

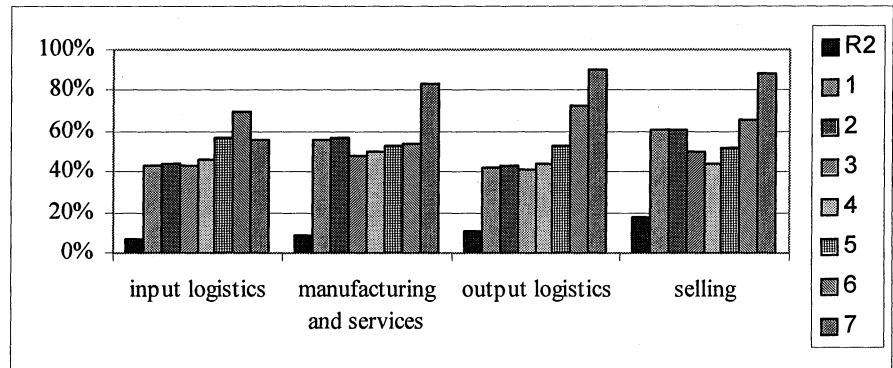


Figure 1 – Impact of information systems on main processes

Figure 2 represents how well can be the impact of information systems on subsidiary processes identified by linear regression and by inductive and logic programming.

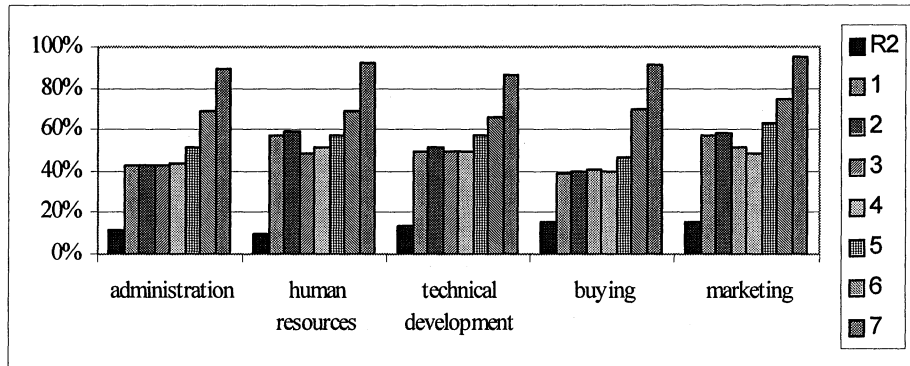


Figure 2 – Impact of information systems on subsidiary processes

To sum up, inductive logic programming on the given data yields better results in every dimension of competitiveness than regression. It could be expected that

large homogenous sets of data will be in most cases better explained by inductive logic programming than by regression. As for the impact of information systems on business competitiveness, the results give evidence that business competitiveness is to some extent influenced by information systems. Differences between  $R^2$  and other bars in Figures 1 and 2 suggest that the impact of information systems is not too linear but it is worth to invest into information systems because their impact is monotonous (non-decreasing). We plan to enhance this method in the future and compare it with other statistical methods.

#### 4. REFERENCES

1. Doucek P. "Integrated Information Systems Improvement". In Strategic Management and its Support by Information Systems, J. Kaluža, ed. Ostrava: VŠB - University of Technology, 1999.
2. Dudinská E, Mizla M. Management information systems. Journal of Economics 1994; 42: 230-38.
3. Džeroski S, Lavrač N. "An introduction to inductive logic programming". In Relational data mining, S. Džeroski, N. Lavrač, eds. Berlin: Springer, 2001.
4. Horváth T, Krajčí S, Lencses R, Vojtáš P. An ILP model for a monotone graded classification problem. Kybernetika 2004: Volume Znalosti 2003.
5. Kifer M, Subrahmanian VS. Theory of generalized annotated logic programming and its applications. J. Logic Programming 1992; 12: 335-67.
6. Kokles M, Romanová A. Information Age. Bratislava: Sprint, 2002.
7. Krajčí S, Lencses R, Vojtáš P. A comparison of fuzzy and annotated logic programming. Accepted in Fuzzy Sets and Systems.
8. Lloyd JW. Foundation of logic programming. Heidelberg: Springer 1987.
9. Muggleton S. Inductive logic programming. New Gen Comp 1991; 8: 295-318.
10. Muggleton S. Inverse entailment and Progol. New Gen Comp 1995; 13: 245-86.
11. Muggleton S, Fhis C. "Efficient induction of logic programs". In Proc. Algorithmic Learning theory, Tokyo: Ohmsha, 1990.
12. Porter M. How competitive forces shape strategy. Harvard Business Review 1979; 57: 137-45.
13. Quinlan JR. Learning logical definitions from relations. Machine Learning 1990; 5: 239-66.
14. Shibata D et al. "An induction algorithm based on fuzzy logic programming". In Proc. PAKDD'99, Ning Zhong, Lizhu Zhou, eds. Beijing: Springer, 1999.
15. Srinivasan A. The Aleph Manual. Tech Rep Comp Lab Oxford Univ. 2000 at <http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.html>
16. Vojtáš P. Fuzzy logic programming. Fuzzy sets and Systems 2001; 124: 361-70.
17. Železný F. "Learning functions from imperfect positive data". In Proc ILP 2001, C. Rouveirol, M. Sebag, eds. Springer, 2001.
18. ILP system ALEPH for Win <http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.pl>.