Distr.
GENERAL

Working Paper No.22
28 April 2009

ENGLISH ONLY

UNITED NATIONS STATISTICAL COMMISSION and ECONOMIC COMMISSION FOR EUROPE CONFERENCE OF EUROPEAN STATISTICIANS

EUROPEAN COMMISSION STATISTICAL OFFICE OF THE EUROPEAN COMMUNITIES (EUROSTAT)

ORGANISATION FOR ECONOMIC COOPERATION AND DEVELOPMENT (OECD) STATISTICS DIRECTORATE

Meeting on the Management of Statistical Information Systems (MSIS 2009)
(Oslo, Norway, 18-20 May 2009)

Topic (iii): Architecture

SOFT COMPUTING TECHNIQUES FOR STATISTICAL DATABASES

Supporting Paper

Prepared by Miroslav Hudec, Infostat, Slovakia

I. INTRODUCTION

1. The expression “hard computing” is used for software tools which are based on the crisp or {true, false} logic and uses usual computing techniques. In many cases this approach meets all users' needs. In several cases the crisp definition of an analysed task cannot be created or the user wants to obtain additional valuable information. The new approach called soft computing offers techniques to meet these needs. This paper examines soft computing techniques for two often used processes: data selection (database queries) and data classification.

2. When users work with usual software tools they have to change their many-valued logical thinking (approximate reasoning) into the two-valued computer logic. If two-valued logic in selection process is used then the small error in data values or cases when user can not unambiguously define the criterion by crisps boundaries may involve some inadequately selected or non-selected data. To avoid this, the wanted scenario for the user is to determine what kind of data he wants to select or classify by linguistic expressions and degrees of truth. These expressions have the logical meaning for user and describe an analysed task in the natural language.

3. Statistics is a perspective area of the soft computing approach. Statistical indicators are often collected with some errors and vagueness and classical techniques may involve some inadequately selected or classified data for example. In this paper data selection and classification are analysed through many-valued fuzzy logic, the constituent part of soft computing. For this purpose mathematical equations are created in order to develop easy to use soft computing tool and to reuse it for other databases and purposes.

II. SOFT COMPUTING

4. A term soft computing does not have a precise and generally accepted definition. Furthermore, the field that has been labeled soft computing is fairly new and thus still evolving, making it difficult to provide

a final definition. Nevertheless, the advantages of soft computing are obvious. The essential property of soft computing is that soft computing contains a family of techniques which are a complement of hard computing for coping with the imprecision, ambiguity and uncertainty. One of definitions is as follows: Every computing process that purposely includes imprecision into the calculation on one or more levels and allows this imprecision either to change (decrease) the granularity of the problem, or to "soften" the goal of optimization at some stage, is defined as to belonging to the field of soft computing [5].

5. The principal constituents of soft computing are fuzzy logic, neural networks, genetic algorithms, belief networks, chaotic systems and other techniques. These techniques are advisable when hard computing techniques can not find the solution or it takes too long to reach solution of a problem. It is recommended to use soft computing in following situations:

(a) full data availability does not exist and some data noise is detected;
(b) imprecision and vagueness in indicator values or model parameters;
(c) nonlinear and chaotic relationships;
(d) adaptive feedback processes;
(e) the user wants to uncover more information than hard computing tools allow.

6. Computing with words is inspired by the remarkable human capability to perform a wide variety of physical and mental tasks without measurements and computations. For example, the wanted scenario for the user is to determine what kind of data he wants to select or classify by linguistic expressions. These linguistic expressions have the logical meaning for user and describe the analysed task in the natural language. The conversion from human into computer language and all mathematical operations are implemented behind the user interface and the user obtains the final solution of a task.

A. Some of differences between soft and hard computing

7. The core of both hard computing and soft computing represented by fuzzy logic is the idea of a set. In the classical set theory an element belongs or does not belong to a set. For example consider a set called high unemployment (HU) defined as follows: HU={x| unemployment(x)>=10%} where x is a region. It means that region with 9.95% unemployment does not belong to the HU but region with 10% belongs. These constraints are drawback when the boundaries between values of some attributes are continuous.

8. The fuzzy set theory permits the gradation of the membership of the element in a set. This gradation is described by a membership function μ valued in the interval [0, 1]. The HU example can be presented by fuzzy sets shown in figure 1. User could define that the unemployment equal and bigger than 10% (L_p in figure 1) is HU, the unemployment smaller than 8% (L_d in figure 1) definitely is not HU and unemployment between 8% and 10% partially belongs to the HU concept. The fuzzy approach uses knowledge that does not have clearly defined boundaries. Many of the phenomena from real world fall into this class.

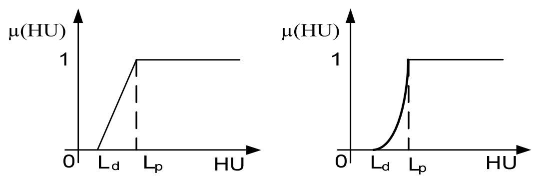


Figure 1: Fuzzy sets for big unemployment concept

9. There is no competition between hard and soft computing. For example soft database queries provide flexibility for the inclusion of records that almost meet the query criterion (potential candidates) and to rank them according to their compatibility with query. Hard database queries are useful when clean and exact boundary between selected and non selected data is required. Figure 2 shows these two concepts and user decides which one is better for each task.

10. Fuzzy sets and fuzzy logic play a necessarily basic role in soft computing. Fuzzy sets and fuzzy logic are used in our research in order to improve selection and classification processes.

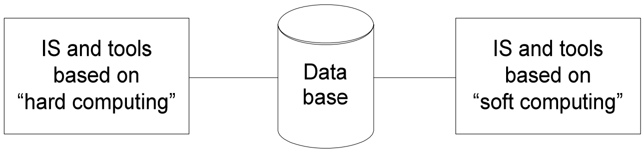


Figure 2: Soft and hard computing tools

III. DATABASE QUERIES

11. Users search databases in order to obtain data needed for analysis, decision making or just to find interesting information. The following question arises: Could be soft computing useful in this process? The answer is yes.

12. The SQL is used to obtain data from relational databases. For this purpose the SQL uses the two-valued logic (crisp logic) in querying process. This querying process is a good example for situations when constraints of two-valued logic may occur. These constraints are explained on the following example:

```
select attribute_1,…,attribute_n
from Table
where attribute_p > P and attribute_r < R.
```

13. The best way how to describe limitations of a SQL query is in the graphic mode shown in figure 3. Values P and R delimit the space of selected data. The user can not obtain any information about records that are close to meet the query criterion (areas marked with gray shadows). The area marked with the darkest gray shadow contains records that almost meet the intent of a query. It means that the record would not be selected even if it is extremely close to the intent of query criteria. Records belonging to shadowed areas could be potential customers and direct marketing could attract them or territorial units which almost satisfy criterion for some financial support for example. In case of no data is selected by SQL, there is not any information concerning possible records that almost meet the query criterion. This is the penalty paid to use the crisp logic in selection process.

14. In cases when the user can not unambiguously separate data he is interested in from those he is not interested in by sharp boundaries or when the user wants to obtain data that are very close to meet the query criterion and to know the index of distance to full query satisfaction, it is necessary to adapt the SQL to these requirements. In this paper the new way of evaluating the WHERE clause of a SQL query is explained. It is interesting to point out that this new way is common for both: data selection and data classification.

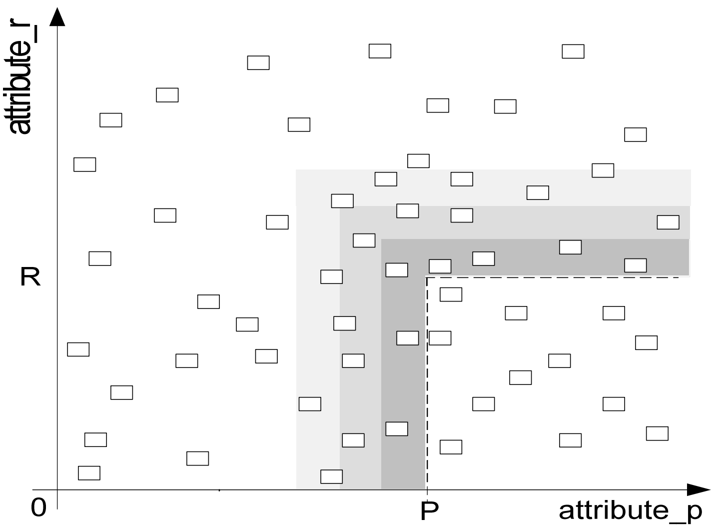


Figure 3. SQL query

A. Fuzzy queries

15. The basic characteristics of fuzzy logic (truth values in the [0,1] interval) provides flexibility for the inclusion of records that are close or almost meet the query criteria. The shape of membership function ($\mu(x)$) can be adjusted according to user's requirements without changing the structure and the meaning of a query.

16. The starting point of our research was the following premise: To make easy to use querying process based on linguistic expressions on client side and to access to relational databases in the same way as SQL. For this purpose the generalized logical condition (GLC) for the WHERE part of the SQL was created in our research. For the further reading it is important to define the query compatibility index (QCI). The query compatibility index (QCI) is used to indicate how the selected record satisfies a query request. The QCI has values from the [0,1] interval: 0 – record does not satisfy a query, 1-record has full query satisfaction, interval (0,1) – record partially satisfies a query and obtained value from this interval is the distance to the full query satisfaction. According to our research the following GLC was achieved:.

$$\text{WHERE} \bigotimes_{i=1}^{n} (a_i \circ L_{ix})$$

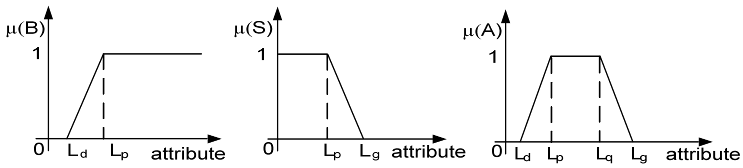
where n denotes number of attributes with fuzzy constraints in a WHERE clause of a query,

$$\otimes = \begin{cases} \mathit{and} \\ \mathit{or} \end{cases}$$

where *and* and *or* are fuzzy logical operators,
and

$$a_i \circ L_{ix} = \begin{cases} a_i > L_{id}, & \mathrm{a_i\ is\ Big} \\ a_i < L_{ig}, & \mathrm{a_i\ is\ Small} \\ a_i > L_{id} \text{ and } a_i < L_{ig}, & a_i \text{ is About} \end{cases}$$

where a_i is a database attribute, L_d is the lower bound and L_g is upper bound of a linguistic expression described by fuzzy set. One type of fuzzy set for "big", "small" and "about" expressions is shown in figure 4.



a) Big value b) Small value c) About value

Figure 4: Fuzzy sets

17. In the first step lower and/or upper bounds of linguistic expressions (fuzzy sets) are used as parameters for database queries (WHERE clause). It means that all records that have QCI greater than zero are selected. In the next step the chosen analytical form of the fuzzy set is used to calculate the membership degree of each selected record to appropriate fuzzy set. Finally, appropriate t-norms or t-conorms are used to calculate QCI values for all retrieved records. When the classical query contains more than one condition in the WHERE clause *and* and *or* logical operator are used. In two-valued case only one logical function for *and* and *or* operators exist because condition is satisfied (value 1) or not (value 0). In many-valued logic there exist many functions describing *and* operator (t-norms) and *or* operator (t-conorms) because each of conditions inside the WHERE clause can be partially satisfied. The following functions can be used as t-norm:

(f) minimum: $\mathrm{QCI} = \min(\mu_i(\mathrm{a_i}))$, i=1,...,n

(g) product: $\mathrm{QCI} = \prod(\mu_i(\mathrm{a_i}))$, i=1,...,n

(h) bounded difference (BD): $\mathrm{QCI} = \max(0, \sum_{i=1}^{n} \mu_i(a_i) - n + 1)$

where $\mu_i(a_i)$ denotes the membership degree of the attribute a_i to the i-th fuzzy set. These functions are chosen because they are easy to aggregate when more than two conditions are used in a WHERE clause and they cover all usual situations. For example, the territorial unit satisfy “high unemployment“ concept with 0.9 and “high migration level“ with 0.7. Both condition are partially satisfied so {0,1} logic is not useful. Min t-norm takes into account the lowest value of membership degrees to fuzzy sets (0.7 in this example). Product t-norm takes into account all membership degrees and balances the query truth membership value across each of the conditions in the WHERE statement clause (0.63 in this example). The similar discussion is valid for t-conorms.

18. If a query needs to contain fuzzy as well as classical constraints, these classical constraints could be easy added to the WHERE clause.

19. A fuzzy query interpreter which transforms fuzzy queries to the classical SQL structure was developed. In this way, queries based on linguistic expressions on the client side are supported and accessing relational databases in the same way as the classical SQL is enabled.

B. Case study

20. This system is tested on data from the Urban and Municipality Statistics database used in the Statistical Office of the Slovak Republic. In this case study, districts with high length of road and small area size are sought. The high road infrastructure density is analysed as an illustrative example. The query has the following form:

```
select district, roads, area
from T
where roads is Big and area is Small
```

The length of road indicator is represented by „Big value“ fuzzy set with these parameters L_d=200km and L_p =300km and the shape as from Figure 4a). The „Small value“ fuzzy set with parameters L_p=450km^2 and L_g =650km^2 and shape as from Figure 4b) describes the area of district attribute.

21. Result of fuzzy query is shown in Table 1. The value of min t-norm is used for the calculation of QCI and district ranking. The Table 1 shows six districts fully satisfying the query; one district is extremely close to satisfying the query and another two districts are close to the query criterion. These three records are marked with darker gray shadow. Other records partially satisfying the query criterion are marked with brighter gray shadow. It means for example that even small changes in districts attributes could involve that another records fully satisfy the query. If SQL was used, this additional valuable information would remain hidden.

Table 1: Result of fuzzy query

District	Roads [km]	Area [km^2]	μ (Road)	μ (Area)	QCI (min)
Bratislava I	335,1	9,6	1	1	1
Piešťany	305,6	381,1	1	1	1
Myjava	563,9	327,4	1	1	1
Púchov	320,9	375,4	1	1	1
Detva	567,2	449,2	1	1	1
Žarnovica	366,6	425,5	1	1	1
Považská Bystrica	324,5	463	1	0,935	0,935
Kysucké N. M.	269,9	173,7	0,7	1	0,7
Senec	269,1	359,9	0,69	1	0,69
Žiar nad Hronom	249,8	517,6	0,5	0,662	0,5
Nové Mesto n V.	528,5	580	1	0,35	0,35
Krupina	334,9	584,9	1	0,326	0,326
Spišská Nová Ves	388,9	587,4	1	0,313	0,313
...					

22. If SQL were used the criterion would be as follows: where roads > 300 and area < 450.

It means that records marked with gray shadows in Table 1 will not be selected. If no data is selected by SQL, fuzzy SQL can inform if there are some records that almost meet the query criterion.

23. In cases when the user uses SQL and wants to obtain similar results like result presented in Table 1 it is needed to make small changes in criterion parameters and to execute very high number of SQL queries. The WHERE clause needs to be modified in the following way:

- **(roads > 300 and area < 450)** extracts records that satisfy initial conditions.
- **(roads > 300 -p_1 and roads < 300) and (area < 450+r_1 and area > 450)** selects records that meet query criterion with value of e.g. 0.9 or almost meet the query criterion (where p_1 and r_1 are small real values greater than zero),
- **(roads > 300 -p_2 and roads < 300 -p_1) and (area < 450+r_2 and area > 450+r_1)** selects records that meet query criterion with value of e.g. 0.8 or records that are very close to query criterion (where p_2>p_1 and r_2>r_1),
- etc.

The following conclusion appears: for the very soft gradation, the infinite number of SQL queries has to be used. In case of fuzzy queries, one query is sufficient.

24. The initial proposal of user interface has been done. The advantages of this approach for users are as follows:

(i) the connection to a database (connection string) and data accessing (SQL command) do not have to be modified;
(j) users do not need to learn a new query language;
(k) the interface supports (quasi) natural language;
(l) presenting of obtained data is in similar way as from SQL but with additional valuable information;
(m) users see data “behind the corner“ (gray areas on Figure 1 or Table 1) and can take into account possible interested data.

IV. DATA CLASSIFICATION

25. Users classify data in order to find where each of classified record belongs. The same question arises again: Could be soft computing useful in this process? The answer is same as for database queries.

26. The usual classification by expert system is illustrated on the following example:

- If I_1 < 70 and I_2 < 10 Then Maintenance is Small (Class C_1);
- If I_1 > 70 and I_2 < 10 Then Maintenance is Medium High (Class C_2);
- If I_1 < 70 and I_2 > 10 Then Maintenance is Medium (Class C_3);
- If I_1 > 70 and I_2 > 10 Then Maintenance is Big (Class C_4).

27. The classification diagram is presented in figure 5. Objects are divided into four classes from class C1 (the smallest) to class C4 (the biggest). This method treats the top rated object T4 in the same way as T3. Units T2 and T3 have similar indicators values. However, T2 and T3 are treated in different classes.

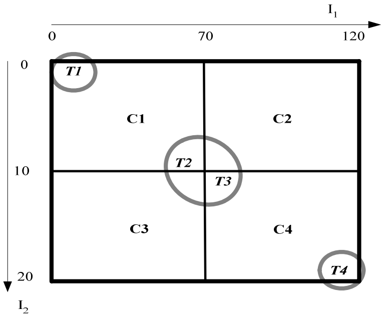


Figure 5: Crisp classes

28. Expert systems offer a good support for classification but again, limitations of two-valued logic may occur. If data values of attributes are similar for two objects (customers, territorial units), they are similar too. In the classical case they may fall into different classes and be treated differently. Soft computing offers a solution to avoid this disadvantage.

A. Classification by the GLC

29. For classification purpose we have examined fuzzy expert systems. The powerful fuzzy system softwares are highly parametric in order to solve wide variety of tasks which implies complicated work for users. If the goal is to create easy to use soft computing tool for classification and to reuse it for other databases and purposes, fuzzy systems are not very suitable from users' point of view. The new idea for classification has been found during our work on fuzzy database queries. Researches have shown that the GLC formula is the good core for data classification too. It was found out that queries are equivalent with the IF part of the rules and result of the query are records that fully or partially belong to the output class. The new approach leads to the integration of selection and classification into one integrated tool by the GLC.

30. The classification query language is designed in the spirit of the above described fuzzy queries. The difference is in the added clause *classify_into*. The *classify_into* clause specifies the name of the output class to which selected records satisfying the GLC are classified. The structure is as follows:

classify_into [class_i]
select [attribute_1],….[attribute_n]
from [tables, relations]
where $\overset{K}{\underset{k=1}{\oplus}} \overset{n}{\underset{i=1}{\otimes}} (a_i \circ L_{ix})$

where $\otimes$ is AND operator, *n* is the number of attributes inside the IF part of the rule, $\oplus$ is OR operator which connects those *k* antecedents in IF part that have common THEN part or the same output class.

31. The result of all queries are records selected into overlapping classes. The final rank for each record can be calculated from the equation:

$$R_O = \sum_{i=1}^{m} \mu_{Oci} P_i$$

where *m* is number of classes, μ_{Oci} is the membership degree of object *O* to class C_i and P_i is the coefficient describing class C_i.

32. Advantages of this approach are as follows:
(n) Queries select only records that will be classified. Records that do not belong to any class are not needlessly selected;
(o) Data preparation to adequate input vector or matrix for fuzzy expert systems is not needed;
(p) Presentation of results in a useful and understandable form could be easy implemented.

B. Case study

33. The system is applied and tested for municipalities classification using data of Banská Bystrica region. This region is one of the eight administrative regions in the Slovak Republic. In this case study municipalities are classified according to the percentage of needs for the winter road maintenance.

34. In this illustrative example two attributes are used and fuzzified into two sets: length of roads in kilometres (Road) and number of days with snow (Snow). These sets are shown in Figure 6.

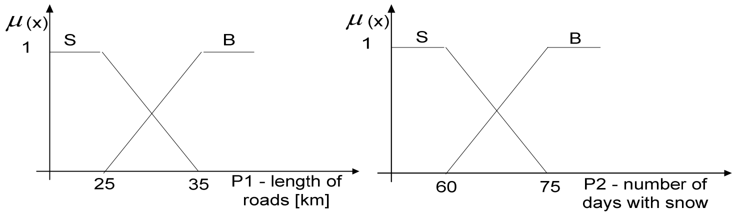


Figure 6: fuzzy sets small (S) and big (B) for Roads (on the left side) and Snow (on the right side) indicators.

35. This example contains four fuzzy rules with the following structure:

- If Road is Small and Snow is Small Then Maintenance is Small;
- If Road is Small and Snow is Big Then Maintenance is Medium;
- If Road is Big and Snow is Small Then Maintenance is Medium;
- If Road is Big and Snow is Big Then Maintenance is Big.

36. Three fuzzy queries are created from these four rules i.e. one query for the corresponding output class:

```
classify_into S
select municipality, roads, snow
from Table
where roads is Small and snow is Small;

classify_into M
select municipality, roads, snow
from Table
where (roads is Small and snow is Big) or (roads is Big and snow is Small);

classify_into B
select municipality, roads, snow
from Table
where roads is Big and snow is Big.
```

37. The percentage of needs for winter road maintenance can be associated with each fuzzy output class: for instance class S (small) gets a percentage of needs of 10% or P_s=0.1, class M (medium) gets 50% and municipalities from class B (big) gets 90% from considered needs.

38. From membership degree to each class the coefficient of needs (R) is calculated. Four municipalities are taken as example:

- **Dudnice** fully belongs to class S. P(Dudnice)=1*0.1=0.1;
- **Čebovce** belongs to class S with degree of 0.267 and M with degree of 0.733. P(Čebovce)=0.267*0.1+0.733*0.5=0,3932;
- **Donovaly** belongs to class M with degree of 0.1 and B with degree of 0.9. P(Donovaly)=0.1*0.5+0.9*0.9=0.86;
- **Banská Bystrica** fully belongs to class B P(Banská Bystrica)=1*0.9=0.9.

39. This tool classified 126 municipalities (only those for which all needed data were available). Table 2 shows ranking results for some municipalities: 2 of them fully belong to class S, 4 fully belong to class M, 5 fully belong to class B and other 15 partially belong to more than one class (records marked with gray shadow). In case of the hard computing classification these municipalities will be classified into one class only and they will not receive adequate amount of resources. In the usual classification two municipalities with very similar indicators values near the boundary value are classified into different classes and it will cause greater difference between obtained and required resources. To avoid this disadvantage the user has to create significantly greater number of output classes and rules if he wants to use tools based on hard computing. Indeed for very soft ranking the user needs nearly infinite number of input ranges, rules and output classes.

Table 2: Some of classified municipalities

Municipality	Coefficient of needs (P)	Municipality	Coefficient of needs (P)
Radzovce	0,1	D. Harmanec	0,5
Dudince	0,1	Kremnica	0,5
Lipovany	0,1268	Trnavá Hora	0,5
Teplý Vrch	0,2068	Sása	0,5
Tornaľa	0,2103	Poltár	0,5146
Veľký Blh	0,26	Kokava n/R.	0,57
Kalinovo	0,2868	H. Tisovník	0,62
Hajnáčka	0,34	Donovaly	0,86
Vinica	0,34	B. Bystrica	0,9
Čebovce	0,3932	Ľubietová	0,9
D.Plachtince	0,3932	B. Štiavnica	0,9
Pôtor	0,4468	Zvolen	0,9
Skerešovo	0,4732	Detva	0,9

40. Comparisons of SQL and fuzzy queries and the usual and fuzzy classification have proven the following constatation: Small number of sentences based on many-valued logic is as powerful as infinite number of sentences based on two-valued logic. The next step is the design and implementation of here suggested approach.

V. IMPLEMENTATION PROPOSAL

41. The proposed fuzzy model for data selection and classification is shown in Figure 7. When the user wants to select data, the procedure is marked with dashed line. In other case when the user wants to classify data, the process is marked with solid line. In case of reuse the core of fuzzy module (namely the GLC and the knowledge base module) remains the same. Input and output parts have to be adapted to achieve new needs: to connect to appropriate databases and to present results in ways expected by the user (tables, maps).

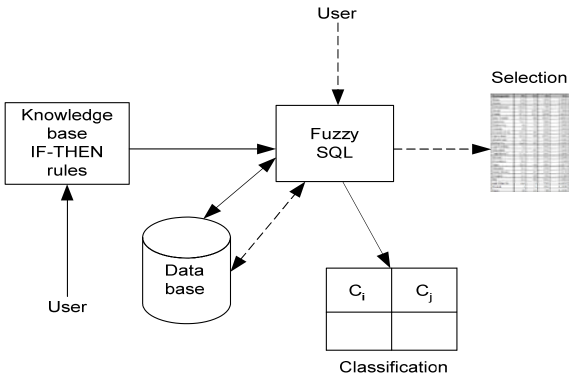


Figure 7: Integration of selection and classification.

VI. CONCLUSION

42. In this paper, the concept of soft computing is outlined and fuzzy logic, as a constituent part of soft computing, is suggested for data selection and classification. When users work with usual software tools they have to change their many-valued logical thinking (approximate reasoning) into the two-valued computer logic. For example, the SQL require crisp specification of a query criterion, while for users a query is better described in terms of a natural (or quasi) natural language with ambiguities and uncertainties. With this approach the user is given a powerful and easy-to-use data mining tool which allows him to query and

classify data from databases using linguistic expressions in order to improve the quality of these two often used processes.

43. Expressions like high rate of unemployment or high number of days with snowing etc. are very often used in statistics. The goal is to capture these expressions and make them suitable for database queries and classification purposes. In many cases the user (analyst, decision maker, etc.) cannot unambiguously separate data by sharp boundaries or user can not expressly argue, why the chosen boundary value is the best one. Users also want to obtain data that are very close to satisfy queries and to know the index of distance to full query satisfaction. This approach gives the user two advantages: the model is described in a (quasi) natural language and the result gives additional valuable information that would remain uncovered if classical methods were used.

44. It is important to point out that there is no competition between hard and soft computing. Soft database queries provide flexibility for the inclusion of records that almost meet the query criterion (potential candidates) and make possible to rank them according to their compatibility with query. Hard techniques are useful when clean and exact boundary in data selection and classification is required.

45. A fuzzy query interpreter was developed to transform fuzzy queries to the classical ones. The interpreter is based on the GLC. The interpreter also creates fuzzy queries from fuzzy rules and converts them into SQL queries using the same GLC. At the end of classification process the interpreter puts records into appropriate classes. In this way, queries and classification rules based on linguistic expressions on the client side are supported and accessing relational databases in the same way as the classical SQL is enabled. No modification of databases has to be undertaken.

46. The fuzzy SQL and fuzzy classification are independent modules. The modularity of here mentioned module allows its uses, modifications and improvements independently. This approach can be reused for another databases or purposes. The core of fuzzy module remain the same only an input and output parts have to be adapted to achieve users' needs. If statistical users are prepared to accept a system that contains the approximate reasoning, the fuzzy tool should be the choice. The soft computing concept has brought many advantages in different areas from industrial process control to direct marketing and decision processes. It could be interesting to use soft computing for statistical purposes and on statistical data. For example, to develop a fuzzy logic tool for data dissemination on statistical websites too to provide querying process across databases using the approximate reasoning instead of the two-valued logic.

REFERENCES

[1] Cox E.: Fuzzy modeling and genetic algorithms for data mininig and exploration, Morgan Kaufmann Publishers, San Francisco, 2005.

[2] Hudec M., Vujošević M.: Fuzzy systems and neuro-fuzzy systems for the municipalities classification. Eurofuse anniversary workshop on “Fuzzy for Better”, Belgrade, 2005.

[3] Hudec M.: Fuzzy improvement of the SQL, Balkan Conference on Operational Research, Belgrade, 2007. http://balcor.fon.bg.ac.yu/unos/Docs/sec06/ 4.pdf.

[4] Hudec M.: Fuzzy Structured Query Language (SQL) for statistical databases, Joint UNECE/Eurostat/OECD Meeting on the Management of Statistical Information Systems, Luxembourg, 2008. http://www.unece.org/ stats/documents/ece/ces/ge.50/2008/wp.12.e.pdf

[5] Li, X., Ruan, D. and van der Wal, A.J.: Discussion on soft computing at FLINS'96, International Journal of Intelligent Systems, 13, 2-3, 287- 300, 1998.

[6] Werro N., Meier A., Mezger C., Schindler G.: Concept and Implementation of a Fuzzy Classification Query Language, DMIN International Conference on Data Mining, Las Vegas, 2005.