ASSO

Help Guide
for SODAS 2 software

Project number:	**IST-2000-25161**
Project acronym:	**ASSO**
Project full title:	**Analysis System of Symbolic Official data**

Deliverable number:	**D3.4a**
Title:	**Help Guide** for SODAS 2 Software
Workpackage contributing to the deliverable:	**WP3: Workbench**
Lead participant (short name):	**FUNDP**
Deliverable type:	**Software-Report**
Security:	**Public**
Author:	**Edited by FUNDP with the help of ASSO scientific partners.**

Abstract:	**This document contains the help guide for the SODAS 2 software of the ASSO project [IST-2000-25161]. It explains the menus and options of all the modules.**
Keywords:	**user, documentation, user manual, help guide**

Reference:	**ASSO/WP3/D3.4a**
Version:	**1.0**
Date:	**28/04/2004**
Total number of pages:	**370**
Status:	**Official**
Type of the document:	**Scientific**
Authorized by:	**M. Noirhomme-Fraiture (FUNDP)**

Distribution List

Participant number	Participant short name	Recipients
1	FUNDP	Monique Noirhomme-Fraiture Edwin Diday Anne de Baenst-Vandenbroucke
2	DECISIA	Alain Morineau
3	TES	Driss Afza
4	FUNDPMa	Jean-Paul Rasson André Hardy
5	INRIA	Yves Lechevallier
6	DAUPHINE	Fabrice Rossi
7	RWTH	Hans Hermann Bock
8	UFPE	Francisco de Assis
9	INE	Carlos Marcelo
10	EUSTAT	Marina Ayestaran Marta Mas
11	STATFI	Seppo Laaksonen
12	FEP	Paula Brito
13	DMS	Carlo Lauro Rosanna Verde
14	DIB	Floriana Esposito Donato Malerba
15	UOA	Haralambos Papageorgiou Maria Vardaki

Amendment History

Version	Date	Actions	Observations
0.1	10 June 2003	Preliminary version	
0.2	16 December 2003	New version	First complete version
1.0	28 April 2004	Final version	Revisted With updating of screen shots

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

Help guide

SODAS 2 Software

Edited by FUNDP
with the help of ASSO scientific partners

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: 28/04/2004

Contents

DATA MANAGEMENT METHODS

Import

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

DB2SO Help guide

From DataBase to Symbolic Objects

G. Hébrail (EDF)
M. Csernel, A. El Golli and Y. Lechevallier
INRIA

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **27/09/2003**

1 DB2S0 : From DataBase to Symbolic Objects

1.1 General information on DB2SO

DB2SO is the part of SODAS software which enables the user to build a set of assertions from data stored in a relational database.

It is assumed that a set of individuals is stored in the database and that these individuals are distributed into some groups. Then DB2SO can build one assertion for each group of individuals. In this process, mother/daughter variables and taxonomies on variable domains can also be associated with generated assertions.

Theoretical aspects of DB2SO are developed in Chapter 5 of the SODAS Scientific Report.

DB2SO is invoked from the IMPORT Menu of the SODAS software.

1.2 Typical use of DB2SO

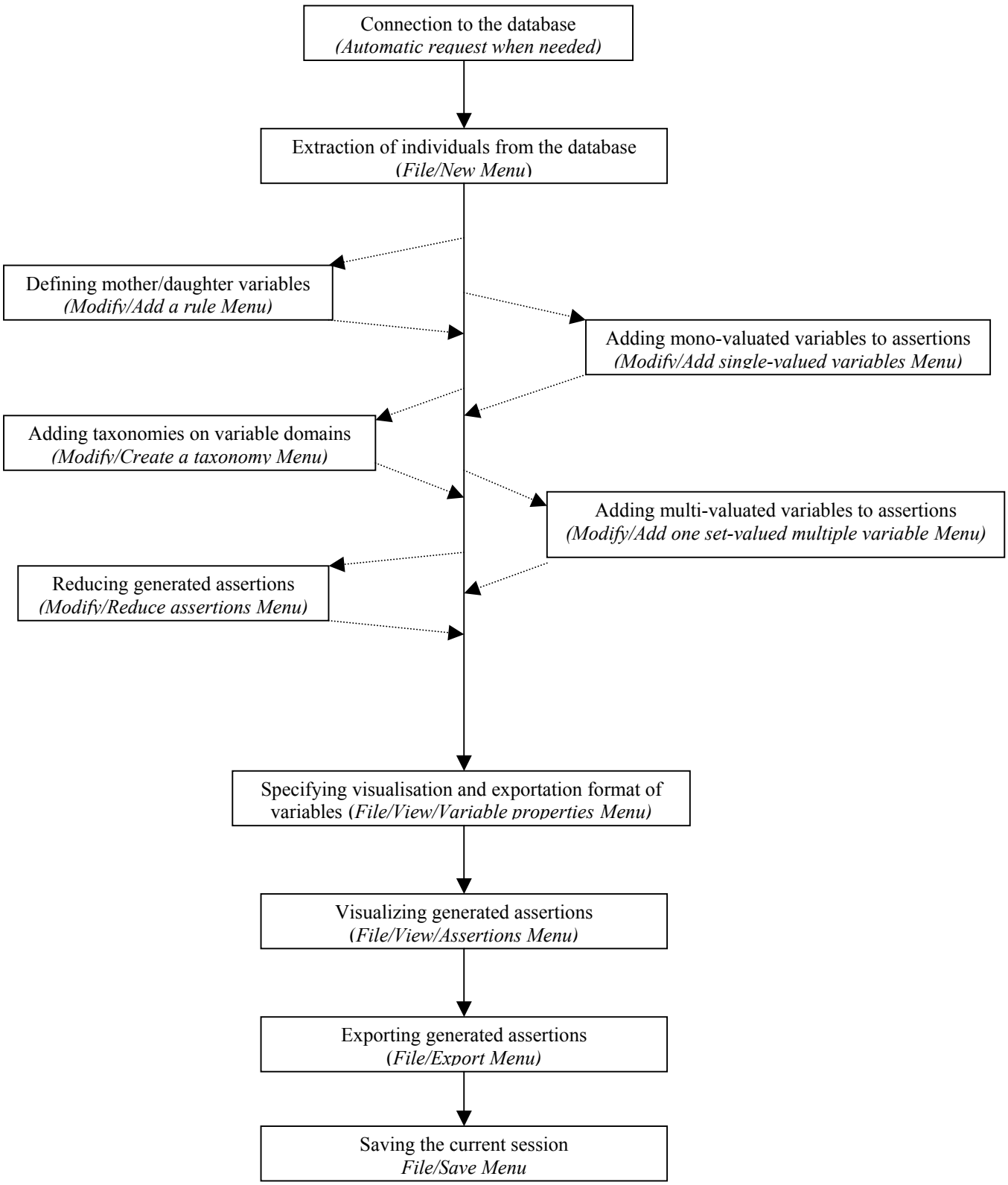
The typical interaction between the end-user and DB2SO is the following:

+ connection to a database : see p.15
+ retrieving individuals distributed into groups by a SQL query [see the *File/New Menu* on p.18]
. optionally defining mother/daughter variables among variables describing individuals [see the *Modify/Add a dependence Menu* on p.24]
. optionally adding single-valued variables to assertions [see the *Modify/Add single-valued variables Menu* on p.22]
. optionally adding multi-valued variables to assertions [see the *Modify/Add one set-valued multiple variable Menu* on p.23]
. optionally adding taxonomies on variable domains [see the *Modify/Create a taxonomy Menu* on p.23]
. optionally simplifying generated assertions using the reduction facility [see the *Modify/Reduce assertions Menu* on p.25]
+ specifying exportation and visualisation format of variables [see the *Modify/Variable properties Menu* on p.26]
+ visualising all work that you have already done [see the *View Menu* on p.27]
+ exporting generated assertions to a SODAS file [see the *File/Export (and view)* Menu on p.20]
+ saving the current session to be able to restart it later [see the *File/Save (as) Menu* on p.20]

Above items marked with a "+" are compulsory steps you typically always go through, while those marked with a "." are optional.

Refer to the figure on next page for a synoptic representation of this typical interaction.

Typical interaction between the end-user and DB2SO



1.3 Connection to the database

DB2SO access to relational database is done by using the Microsoft ODBC facility. Please refer to Microsoft ODBC documentation which is available on your computer under Windows 95 or Windows NT.

We assume here you have already installed on your PC the necessary ODBC drivers to access your database. Standard ODBC drivers are included with Microsoft software (like EXCEL, MSACCESS drivers). If you need to access ORACLE databases or SAS files, you must ask your retailer for the corresponding driver.

DB2SO does not provide any menu to connect to the database because the user is automatically prompted to connect when necessary, i.e. if not yet connected before running a query.

If the user wants to change the database during a DB2SO session, the procedure is the following :

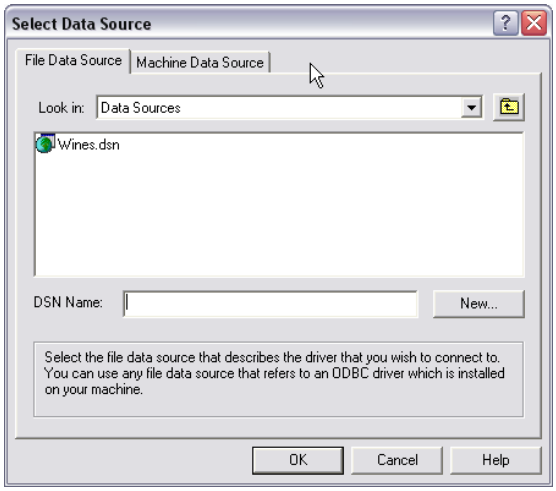
- the user disconnects from the current database by using the *File/Disconnect Menu* (see p.21),

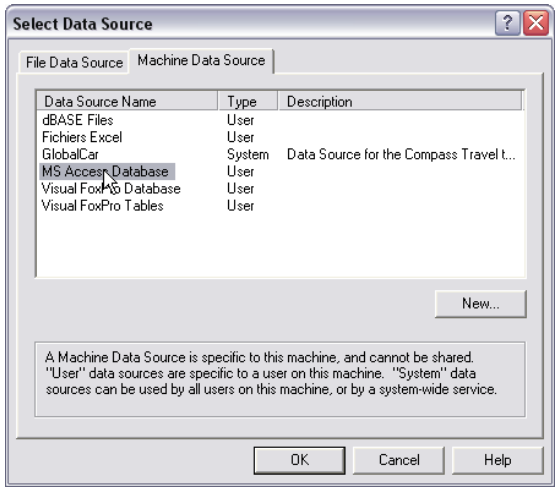
- the user will then be prompted to specify a database before running the next query.

Another way to change the database is to click on the *Modify* button on the window in which SQL queries are written (see the *Writing a SQL query* Section on p.17).

Note that simultaneous connections to several databases are not supported.

When the user is prompted to connect to a database, the left window below appears. The user has to **click on the Machine Data Source button**, which leads to the right window below. In this window, you select the database. If you use an ACCESS database, you have to select the MS ACCESS driver which is available on your computer, and you will then be prompted to choose the file containing the database.



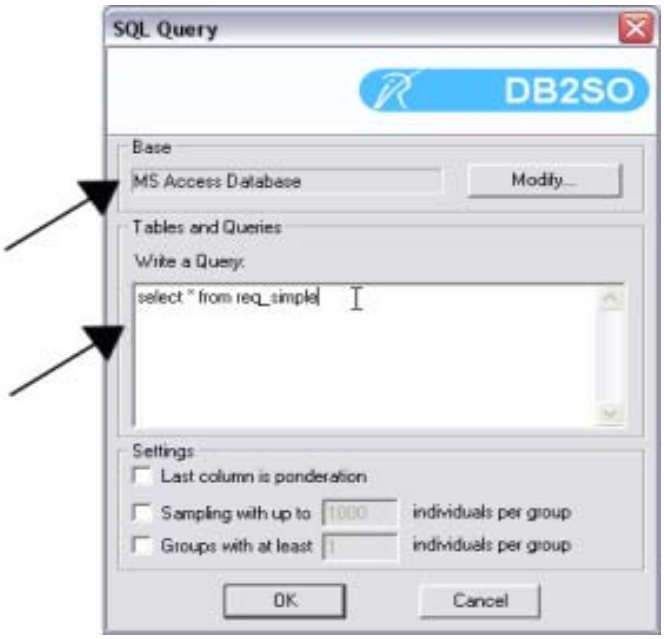


If you use another database system, you have to choose the corresponding ODBC driver.

Note : if you often use the same database file, it is possible to define an entry in the above right window which corresponds to a particular database file : this is done by clicking the New button in this window and follow instructions.

1.4 Writing a SQL query

In different steps of DB2SO, you will be prompted to write SQL queries. Depending on the step you are performing, a different window appears (below is shown the window for retrieving individuals from the database). But these windows always contain the two parts marked below by an arrow :



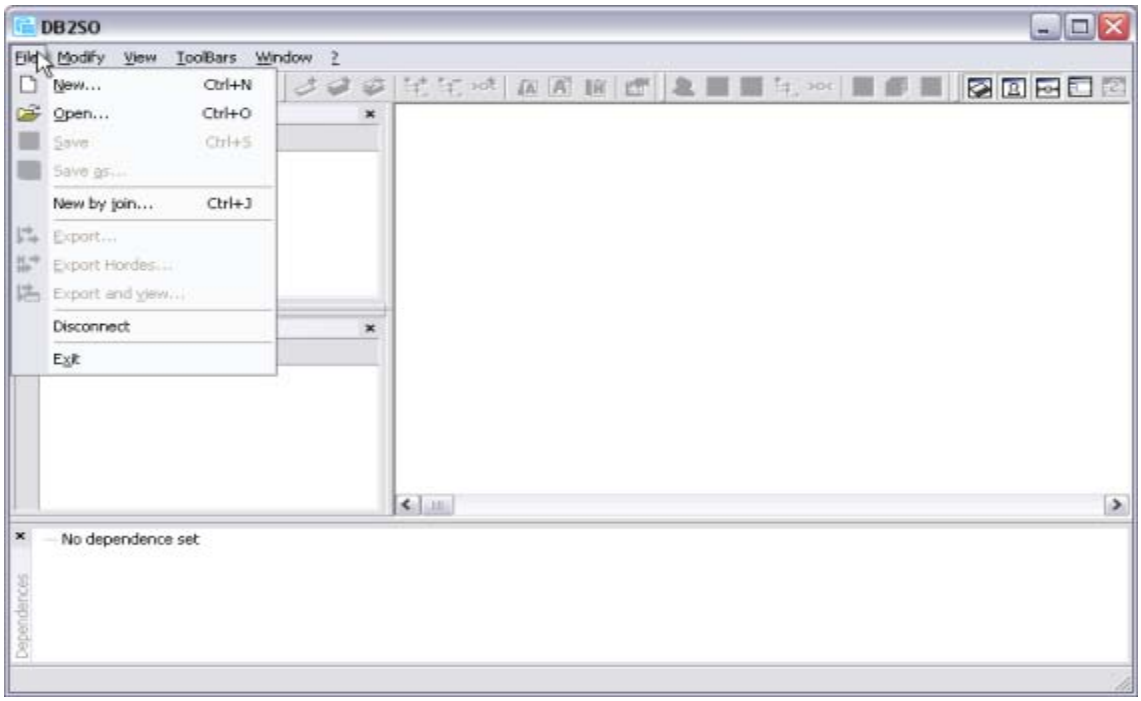
The *Modify* button enables the user to change the database to which the query is submitted.
The *SQL query* space is for typing the query in the SQL language.

Important note on SQL :

- **you have to type standard ODBC SQL** in this window but **not** SQL of the database system. For instance, if you use MS ACCESS database, strings are delimited by ″ while they are delimited by ′ in the standard ODBC SQL.

- **if you ignore everything about ODBC SQL**, the best way to run a query is to prepare it in your database system, store it under a name (for instance *Q1*) as a query or as a view, and type the following query in the above window : *select * from Q1*. This will return to DB2SO exactly what the query returns to your favourite database system.

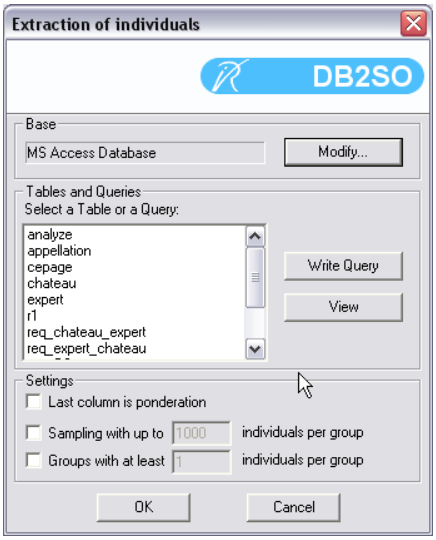
1.5 File Menu



1.5.1 File/New Menu

Initialises a new DB2SO session : this is the command which leads you to specify the query which extracts individuals from the database. So, if you are not already connected to a database, you will be prompted by the ODBC menu to choose the database SQL queries will be submitted (see the Section on p.15).

Once connected to the database, the window above appears :



In this window you can choose a table or a query to extract individuals from.

If the last column of the query is a weight associated with each individual, select the '*Last column is ponderation'* option in the window. If you forget to select this option, the last column will lead to an interval variable in generated assertions.

If the query is supposed to return a very large number of tuples, this may overload the main memory: in this case, the user should select the *sampling* option in the window and specify the maximum number of individuals to be kept for each group (1000 by default, as shown in the above window).

If you have privacy concerns, select "Groups with at least" option, any group with less than the specified number of individuals will be ignored.

You can write your own query using "Write query".

In this window, you have to type a **SQL query of type 1**, as specified in Chapter 5 of the SODAS Scientific Report. Please refer to *Annex 1 : SQL query types* on p.32 for a summary of this Chapter 5.

The query must return one row for each individual and the expected structure of rows is (ID_IND, ID_GROUP, ATT1, ..., ATTp), where ID_IND is the individual identifier, ID_GROUP is the group identifier, and ATTi are attributes describing individuals. The number of attributes is variable but a minimum of one is required.

With either the direct choice of table/query or the user-defined query, the following operations are performed :

- individuals are retrieved from the database and stored in main memory,

- the array of assertions is generated[1].

Both the array of individuals and the array of assertions can then be viewed using the *View Menu* (see p.27).

1.5.2 File/New by Join Menu

This option allows to build an array of assertions by joining two arrays of assertions stored into **.gaj* files. Files named **.gaj* are files storing DB2SO sessions (see the *File/Save (as) Menu* on p.20).

The join operation takes as input two arrays of assertions. It produces a new array of assertions featuring the intersection of the two sets of assertions described by the union of the two corresponding sets of variables. Please refer to Chapter 5 of the SODAS Scientific Report for more information on the join operation on two arrays of assertions.

After specifying the names of two **.gaj* files, the join of the two arrays of assertions is performed and can be saved in another **.gaj* file, or exported to a SODAS file (**.sds*).

Note that no array of individuals is associated with an array of assertions which is the result of a join operation. Consequently, all DB2SO options related to individuals are inactive in this case. MetaData from the original projects will be ignored.

Join of two SODAS files (**.sds*) is not supported by DB2SO.

1.5.3 File/Save (as) Menu

Saves the current session in a **.gaj* file. This enables the user to retrieve later the current state of a DB2SO session, by using the *File/Open Menu* (see p.20).

Just give the name of the file. The extension of these files is *.gaj* by default.

MetaData associated with the current project are saved using the same filename with an appended ***.****xml*.

1.5.4 File/Open Menu

Recovers the session stored in the selected **.gaj* file. Just give the name of the file when prompted.

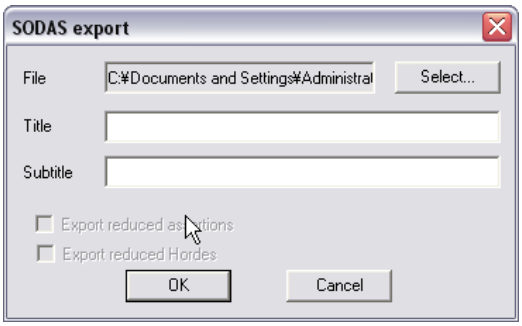
Note that **.sds* files (see the *File/Export (and view) Menu* Section on p.20) cannot be opened by this command.

MetaData associated with this project can be ignored using the “ignore MetaData” option.

1.5.5 File/Export (and view) Menu

Builds a SODAS or XML file containing the array of assertions of the current session. SODAS files are files used as the input of SODAS methods. The extension of SODAS files is *.sds*.

[1] As described in Chapter 5 of the SODAS Scientific Report, numerical variables describing individuals lead to interval variables describing assertions, while non-numerical variables lead to multi-valued – possibly modal – variables.



A title and a subtitle can be given as comments to the contents of the SODAS file.

If assertions/hordes have been reduced (see the *Modify/Reduce assertions Menu* on p.25), the user can either export the reduced assertions/hordes (by selecting the option in the window above) or export the non-reduced assertions/hordes.

Assertions are exported according to the properties defined in the *Modify/Variable properties Menu* (see p.26). These properties enable the user to specify which variables describing the assertions are exported and if they are exported as multi-valued (possibly modal) variables in the case of categorical variables.

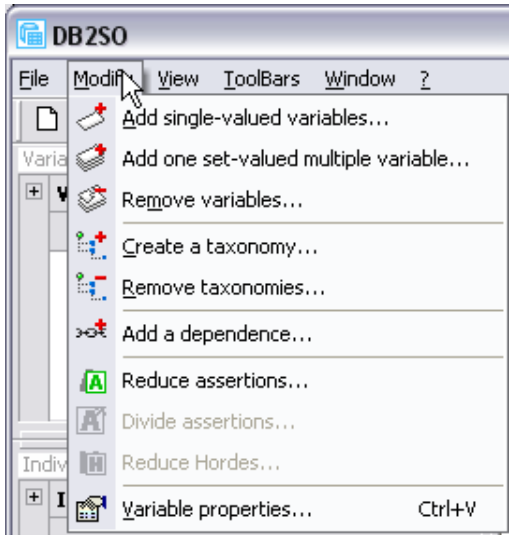
The *File/Export* command just builds the **.sds* of **.xml* file while the *File/Export and view* one also shows the generated file using Wordpad (**.sds*) or Internet Explorer (**.xml*).

MetaData will be saved using the notation filename***_meta.xml***

1.5.6 File/Disconnect Menu

Disconnects DB2SO from the database it is currently connected. This command is useful if the user wants to extract data from another database for further operations with DB2SO.

1.6 Modify Menu

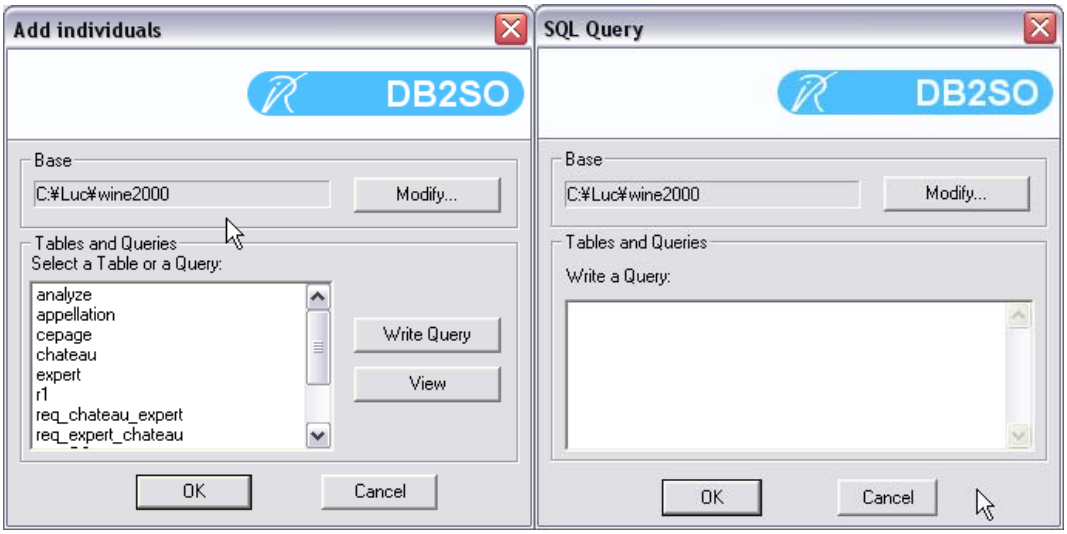


After running a *File/New*, or *File/New by join* command, an array of assertions is generated by DB2SO. In the case of *File/New*, variables describing individuals lead to multi-valued variables describing assertions.

The *Modify Menu* enables the user to modify assertions handled by the current session. Several operations can be performed:

- adding and removing single-valued or multi-valued[2] variables to assertions,
- associating taxonomies with variable domains,
- specifying mother/daughter variables by the mean of rule definition,
- reducing assertions/hordes.
- dividing assertions

1.6.1 Modify/Add single-valued variables Menu



Adds one or several **single-valued** variables to assertions handled by the current session. Values of these variables for the assertions of the current session are extracted from the database by a direct choice or a **SQL query of type 2** (please refer to *Annex 1 : SQL query types* on p.32 for more information about SQL query types in DB2SO).

A SQL query of type 2 has the following structure:

ID_ASSERTION, ASSERTION_ATT1, ..., ASSERTION_ATTp

where ID_ ASSERTION is the assertion identifier and ASSERTION_ATTi are one or several variables describing assertions. Only one row should be returned by this query for each assertion ID.

If the query returns a null value for some variables of some assertions, a missing value will be associated with the corresponding assertion for these variables.

If the query returns ASSERTION ID's which do not exist in the array of the current session, information about these ASSERTION ID's is lost.

[2] Also called a *set-valued multiple* variable.

If the query does not return a row for an ASSERTION ID of the array of the current session, a missing value will be associated with all created variables of this assertion.

1.6.2 Modify/Add one set-valued multiple variable Menu

Adds **one** set-valued multiple variable to the array of assertions. Values for this new variable are extracted from the database by a **SQL query of type 3** (please refer to *Annex 1 : SQL query types* on p.32 for more information about SQL query types in DB2SO).

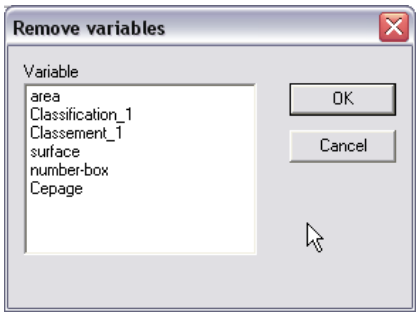
The user just has to do a direct choice or supply a SQL query in a window which is similar to the one for adding a single-valued variable.

In the case where the set of assertion ID's of the current session differs from those returned by the query, same rules apply as described in the preceding Section.

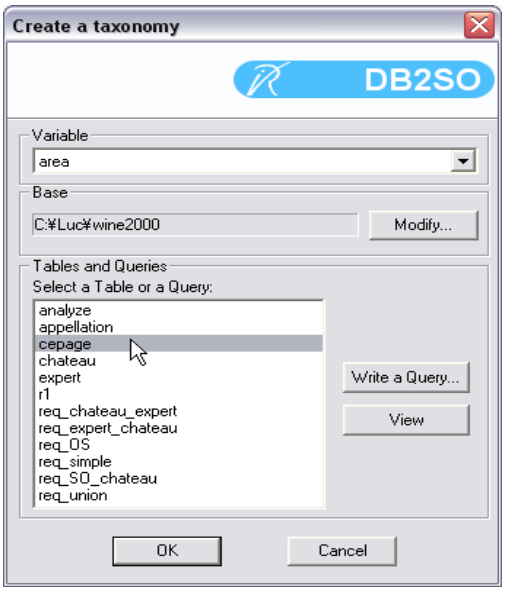
1.6.3 Modify/Remove variables Menu

Enables the user to remove variables which have been added either by the *Modify/Add single-valued variables Menu* or the *Modify/Add one set-valued multiple variable Menu*.

Just select the variable you want to remove in the window shown below :



1.6.4 Modify/Create a taxonomy Menu

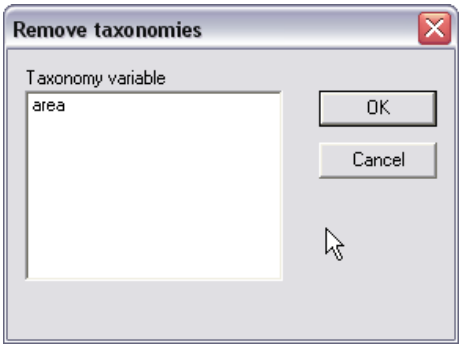


Adds a taxonomy structure to the domain of a variable describing generated assertions.

Data associated with the taxonomy is extracted from the database by a direct choice or a **SQL query of type 5a or 5b** (please refer to *Annex 1 : SQL query types* on p.32 for more information about SQL query types in DB2SO), the system recognises if it is a query of type 5a or 5b.

The user has to select in the pop-down list the variable to which the taxonomy will be added. Messages are displayed to the user if data returned by the query is invalid.

1.6.5 Modify/Remove taxonomies Menu



Removes the taxonomy associated with the selected variable.

1.6.6 Modify/Add a dependence Menu

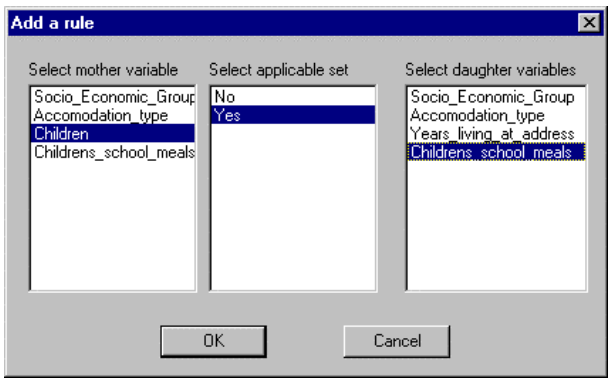
Enables the user to define mother/daughter variables **among variables describing individuals** (it is not possible to define mother/daughter variables on variables added with the *Modify/Add single-valued variables Menu* or *Modify/Add one set-valued multiple variable Menu*).

Mother/daughter variables are defined by rules of the form:

Salary IS APPLICABLE IF *Activity* IN (*'Working', 'Retired with a pension'*)

where *Salary* and *Activity* are variable names, and *'Working'* and *'Retired with a pension'* are values of the *Children* variable.

This rules mean that there is no value of the *Salary* variable for individuals having a value of the *Activity* variable different from *'Working'* and *'Retired with a pension'*, for instance for individuals having a value *'Unemployed'*.



The specification of one rule is done through the window shown above. In this sample window, the following rule has been specified:

Childrens_school_meals IS APPLICABLE IF *Children* IN (*'Yes'*)

First select the mother variable in the left list, then all values of the mother variable appear in the central list. Select one or several values in the central list. Finally select one or several variables which will be the daughter variables. When several daughter variables are selected, there will be as many rules generated.

After clicking OK in this window, DB2SO checks if individuals respect the definition of the rules:

- individuals supposed to have a NA (NA stands for NON-APPLICABLE) value for the daughter variable are expected to have a NULL value in the database. If it not the case, DB2SO asks the user a confirmation before forcing the value to NA for all such individuals. **This operation cannot be undone**.

- Individuals having a NULL value in the database and supposed to have a non-NA value for the daughter variable will be considered as having a missing value in DB2SO.

Limitations:

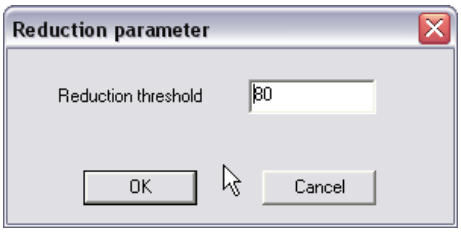
- a variable can be the daughter variable of **only one** mother variable,

- only nominal variables can be mother variables,

- since values of individual daughter variables may be changed by this operation, **adding a rule cannot be undone**. If you are not sure of what you are doing, please save your session before adding rules.

Mother/daughter variables and their associated rules can be displayed by the *View/Dependences Menu* (see p.29).

1.6.7 Modify/Reduce assertions Menu



Performs a reduction of assertions: DB2SO automatically simplifies generated assertions by removing untypical individuals.

Please refer to the SODAS Scientific Report for more information about this reduction process.

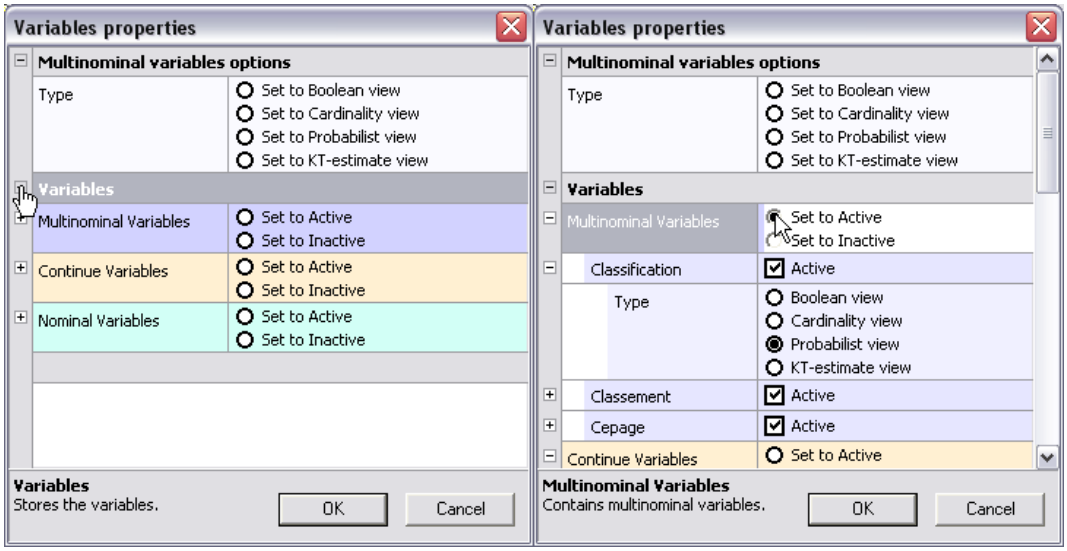
The user has to give a threshold which corresponds to the minimum percentage of individuals still recognized by each assertion after the reduction process.

1.6.8 Modify/Variable properties Menu

In this window, the user specifies the format of assertions for display (in the *View/Assertions Menu* and the *View/Reduced assertions Menu*) and exportation (in the *File/Export (and view) Menu*). Both variables generated from individual variables and variables directly added to assertions are accessible in this menu.

An *Inactive* variable will be invisible for both view and exportation. For **multinominal** variables, the user can choose between a boolean, cardinality, probabilist or KT-estimate probabilist representations.

Note that cardinality representation is not available if a weight has been associated with individuals (see *File/New Menu* on p.18).



Anchors on the left expand/collapse the tabs.

A **double click** on a tab does the same job but may result in **system overload** if more than two clicks are made.

Each specific variable type (i.e. with different properties) has its own tab.

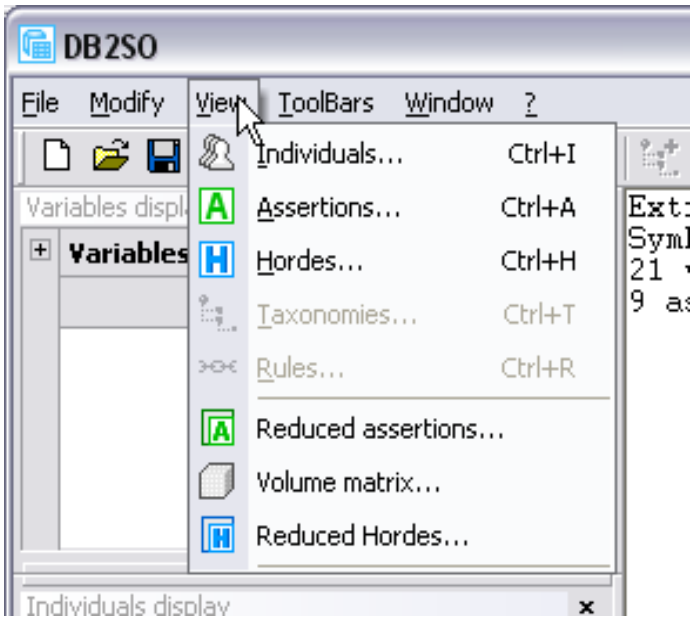
You can activate or de-activate every variable of a type by clicking on the corresponding radio in the type-tab.

You can set the default representation for multinominal variables in the same fashion (*Multinominal variables options* tab).

To activate your changes you **MUST** click on the control (for example a tab) before clicking on OK or the last operation will be discarded. You will know when the changes are applied when the background color will go back from white to its original color.

Side note: the “contagious” options may disappear, this is a windows GUI bug, but they are still “here”, they will reappear with the next click on them.

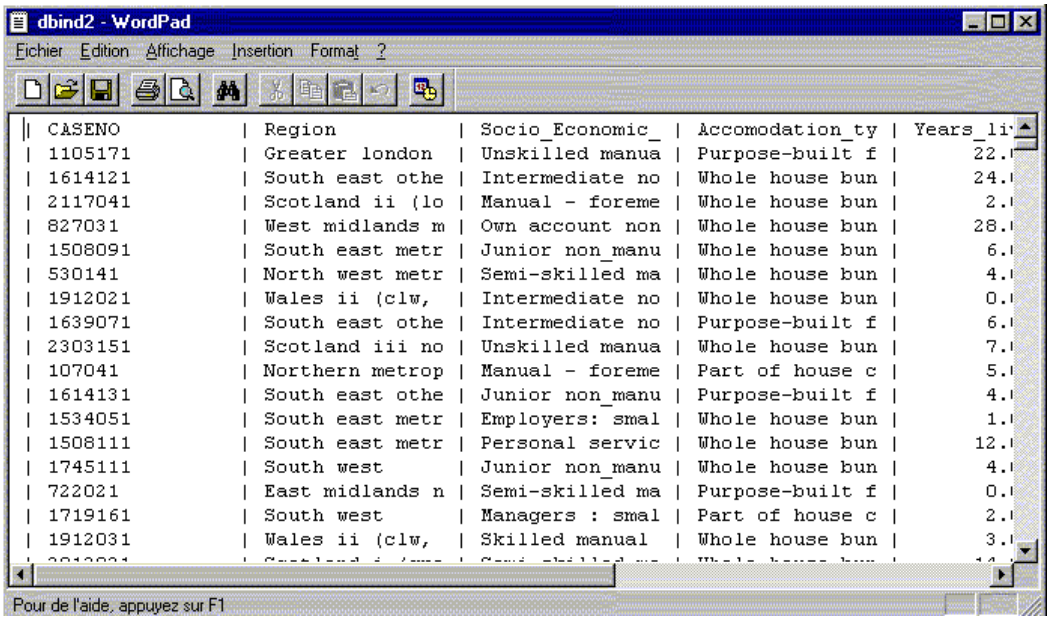
1.7 View Menu



1.7.1 View/Individuals Menu

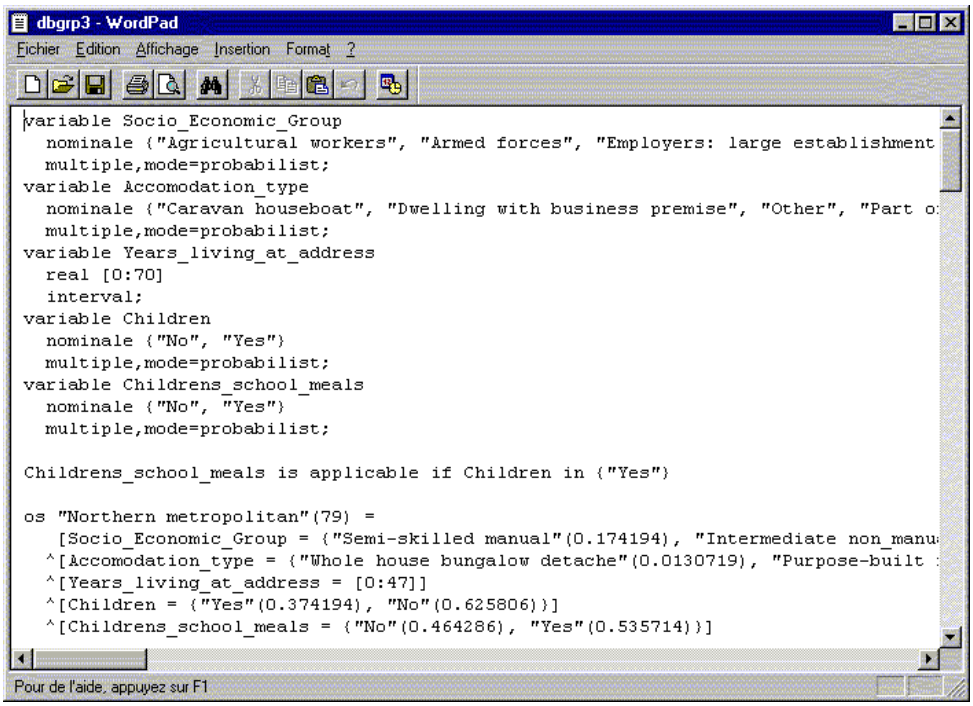
Displays in a Wordpad window the current array of individuals (see the example below). Changes made in Wordpad are ineffective in DB2SO.

Note that you can save what is displayed with Wordpad in a file you may use for other purposes.

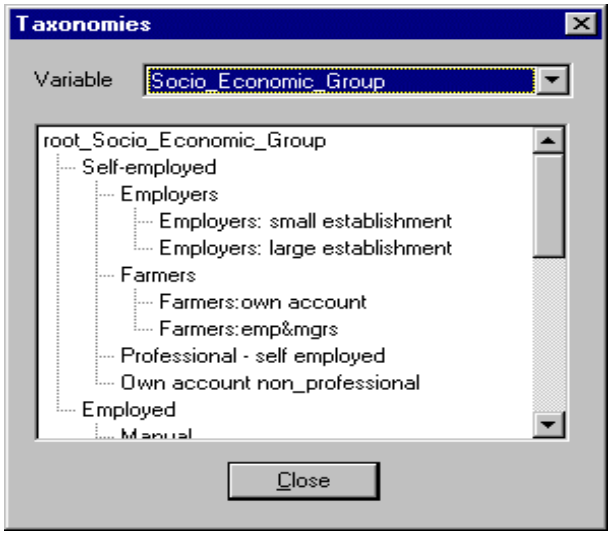


1.7.2 View/Assertions Menu

Displays in a Wordpad window the generated assertions in the SOL language (see the example below).

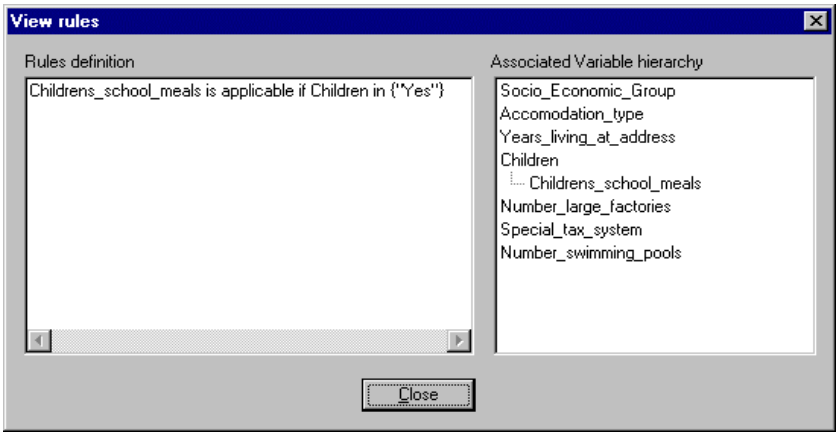


1.7.3 View/Taxonomies Menu



Selecting this option in the *View Menu* opens the window shown above. Select in the pop-down list the variable (here *Socio_Economic_Group*) for which the associated taxonomy has to be displayed. Then the taxonomy is displayed in the scrollbar window as shown above.

1.7.4 View/Dependences Menu

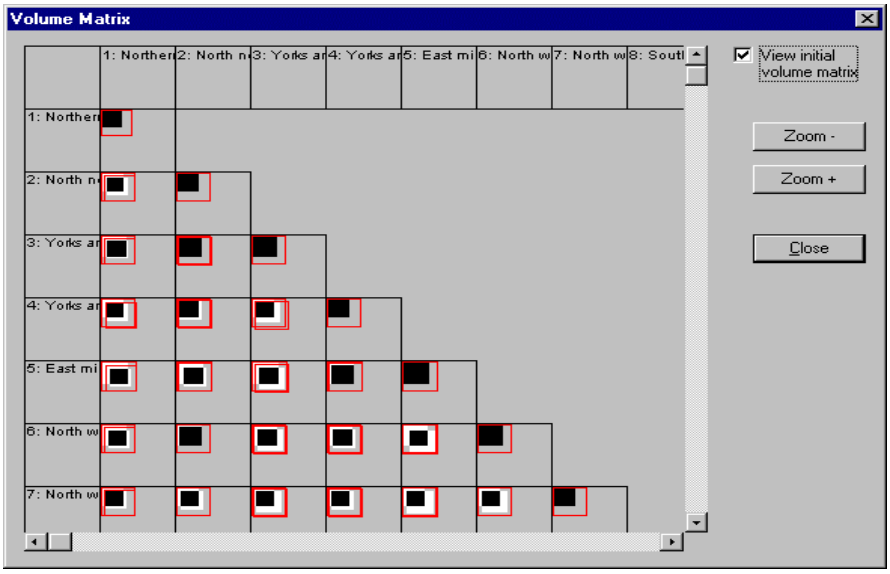


Selecting this option in the *View Menu* opens the window shown above. In the left part, rules representing mother/daughter variables are displayed. In the right part, the hierarchy induced by the definition of mother/daughter variable rules is displayed.

1.7.5 View/Reduced assertions Menu

Same as the *View/Assertions Menu* except that reduced assertions are displayed, if a reduction process (see *Modify/Reduce assertions Menu* on p.25) has been performed before.

1.7.6 View/Volume Matrx Menu



Displays the volume matrix associated with the reduction process of assertions. For a definition of the volume of an assertion, please refer to Section 5.4.3 of Chapter 5 of the SODAS Scientific Report.

The matrix is a square matrix in which each cell corresponds to a couple of assertions. Each cell shows :

- a white square with a surface proportional to the volume of the corresponding row assertion,

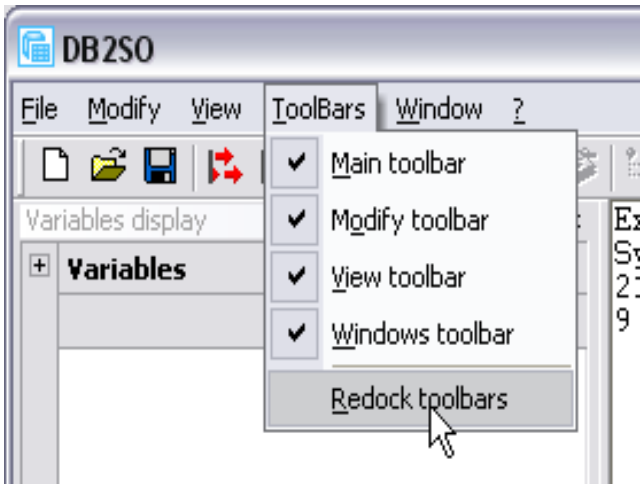
- another white square with a surface proportional to the volume of the corresponding column assertion,

- a black square representing the volume of the intersection of the row and column assertions. (note that the two white squares are laid out so that their intersection is the black square).

When selecting the *View initial volume matrix*, red line squares show the state of this volume matrix without running the reduction process. This visualisation gives an idea of the effect of assertion reduction (the reduction process is supposed to reduce intersection - overlap - between assertions).

A zoom is available in this window.

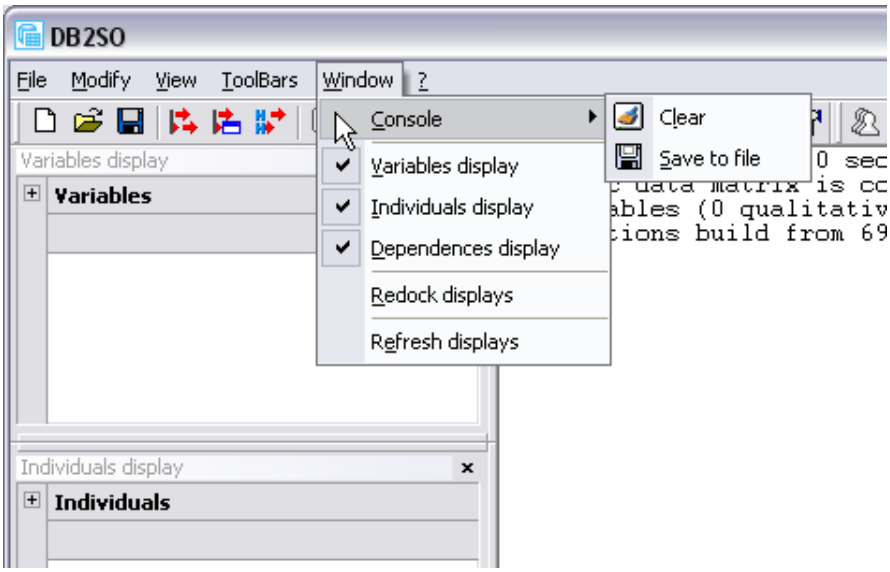
1.8 ToolBars Menu



This menu allows you to display or hide the toolbars associated with the DB2SO menus.

Redock toolbars will replace all the visible toolbars to a default docked position (usefull if the toolbars are messed up by a window resizing or by an improper un-docking).

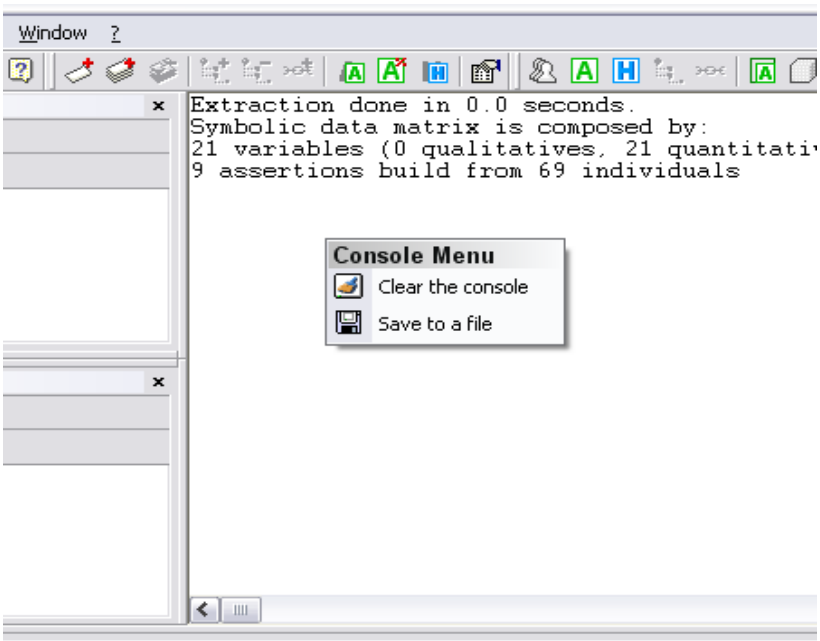
1.9 Window Menu



This menu allows you to show/hide the information windows (called *displays*), redock them, refresh them, clear the console and save the console content to a text file.

1.10 The Console

The console is where all texts are displayed. Some of its information may be important to you, that s why we included the Save to File feature. A right-click on the console will popup the console menu.



Annex 1 : SQL query types

SQL query of type 1:

This query type is used to retrieve the following information from the database:

- *individuals ID* and *attributes*, possibly associated with a weight for each individual. These attributes lead to multi-valued variables describing generated assertions.

- *groups ID* which become the names of generated assertions.

- membership of each individual to one or several groups.

To summarize, the structure of this query type is the following:

IND_ID, GROUP_ID, IND_V1, IND_V2, ..., IND_Vk [, IND_WEIGHT]

This query must contain at least three columns (individual ID, group ID and one variable describing individuals). The weight column is optional and the number of individual attributes is variable.

Example:

CASENO	Region	Bedroom	Dining\living	Socio - Economic Group
114051	Northern metropolitan	2	1	Managers : large establishment
114061	Northern metropolitan	2	1	Own account non_professional
114071	Northern metropolitan	2	1	Intermediate non_manual ancill
114101	Northern metropolitan	3	2	Semi-skilled manual
114111	Northern metropolitan	2	1	Managers : large establishment
114131	Northern metropolitan	1	1	Intermediate non_manual ancill
114141	Northern metropolitan	3	1	Unskilled manual
114161	Northern metropolitan	3	1	Managers : large establishment
114171	Northern metropolitan	3	1	Skilled manual
201081	North non-metropolitan	2	1	Semi-skilled manual
201091	North non-metropolitan	4	3	Skilled manual
201101	North non-metropolitan	3	1	Personal service
201111	North non-metropolitan	2	1	Skilled manual
201131	North non-metropolitan	2	1	Junior non_manual
201141	North non-metropolitan	3	2	Unskilled manual
201171	North non-metropolitan	3	1	Intermediate non_manual ancill
...	...			

This query returns one row per household. Individuals are households identified by the first column: CASENO. The group ID is defined by the second column, here REGION. There will be one assertion generated for each region. Households are described by two numerical and one qualitative attribute: the number of two types of rooms in the house and the socio-economic group of the household. Both numerical and qualitative attributes can describe individuals. The only difference is in the generalisation process where numerical variables lead to intervals of values while qualitative ones lead to lists of values or probability distributions. In this query, there is no weight column. If a weight column is present, this must be the last column and the user has to tell it to the system.

The weight column is used in the generalisation process for the computation of probability values associated with modal multi-valued variables. When no weight column is present, all individuals are assumed to have the same weight, equal to 1.

SQL query of type 2:

This query type is used to retrieve the following information from the database:

- *single-valued attributes* describing *groups*: these attributes lead to single-valued variables describing generated assertions.

The structure of this query type is the following:

GROUP_ID, GROUP_V1, GROUP_V2, ..., GROUPE_Vp

This query must contain at least two columns: the group ID and one group attribute. The number of group attributes is variable.

Example:

Region	Number of cars	%Unemployment
East anglia	273000	8
East midlands non-metropolitan	491000	9
Greater london north east	197000	10
Greater london north west	154000	7
Greater london south east	149000	8
Greater london south west	160000	6
North non-metropolitan	260000	10
North west metropolitan	458000	11
...	...	...

This query returns two single-valued attributes describing regions. There is one row per region, thus one value per region for every attribute. Here the two attributes are numerical but they could have been qualitative. Both qualitative and numerical attributes can be returned in the same query.

SQL query of type 3:

This query type is used to retrieve the following information from the database:

- exactly one *native multi-valued (qualitative) attribute* describing *groups*, which lead to a multi-valued variable describing generated assertions

The structure of this query type is the following:

GROUP_ID, MULT_VAL_ATT, CARDINALITY

Example:

Region	Accomodation type	Number of cases
East anglia	Caravan houseboat	1
East anglia	Part of house converted flat o	16
East anglia	Purpose-built flat or maisonet	22
East anglia	Whole house bungalow detache	85
East anglia	Whole house bungalow semi-de	83
East anglia	Whole house bungalow terrace	66
East midlands non-metropolitan	Caravan houseboat	3
East midlands non-metropolitan	Dwelling with business premise	1
East midlands non-metropolitan	Part of house converted flat o	17
East midlands non-metropolitan	Purpose-built flat or maisonet	42
East midlands non-metropolitan	Whole house bungalow detache	134
East midlands non-metropolitan	Whole house bungalow semi-de	182
East midlands non-metropolitan	Whole house bungalow terrace	112
Greater london north east	Dwelling with business premise	1
Greater london north east	Part of house converted flat o	17
...	...	...

The first column of the query describes the group ID, the second one contains modalities of the multi-valued attribute, and the third one is the cardinality of the modality within the group.

This query type is useful to acquire multi-valued variables describing assertions when they are not computed from an underlying population. It is not adviced to use this feature if the attribute describes an underlying population: in this case, it is preferable to use a SQL query of type 1 retrieving the underlying population, run the reduction process, and finally join generated assertions to the current ones.

SQL query of type 4: *(not supported by version V2 of DB2SO)*

This query type is used to retrieve the following information from the database:

- exactly one *native interval (numerical) attribute* describing *groups*, which lead to an interval variable describing generated assertions

The structure of this query type is the following:

GROUP_ID, MIN_VALUE, MAX_VALUE

where MIN_VALUE and MAX_VALUE are the bounds of the interval of the new variable describing groups identified by GROUP_ID.

Example:

REGION	LOWER	UPPER
East anglia	1531,59	3058,34
East midlands non-metropolitan	-84,31	259,95
Greater london north east	3525,06	6878,74
Greater london north west	27132,72	32852,73
Greater london south east	33847,21	41307,62
Greater london south west	-52,99	326,95
North non-metropolitan	-65,69	405,41
North west metropolitan	1833,84	3336,49
...	...	...

In this example, this query returns, for each region, a confidence interval for the average income of a particular class of people. This will create an interval numerical variable describing assertions. As for queries of type 3, it is not adviced to use this feature when this data can be calculated from an underlying population.

SQL query of type 5:

This query type is used to retrieve the following information from the database:

- a *taxonomy* on values of a particular variable

This information can be retrieved from the database by two means, corresponding to two different ways of representing a taxonomy in a table:

SQL query of type 5.a ATT_VALUE, LEV1_VALUE, LEV2_VALUE, ..., LEVp_VALUE

This is the representation suited when all leaves of the taxonomy are at the same depth in the tree. A typical example is a query returning TOWN, DEPARTMENT, REGION, COUNTRY to define a taxonomy on a TOWN variable.

SQL query of type 5.b CHILD_ATT_VALUE, PARENT_ATT_VALUE

This is a child/parent representation of the taxonomy which links together the different values of an attribute domain. An example of this representation is given below.

Example of the child/parent representation

Socio - Economic Group	Parent Group
Agricultural workers	Employed
Armed forces	Employed
Employers	Self-employed
Employers: large establishment	Employers
Employers: small establishment	Employers
Farmers	Self-employed
Farmers:emp&mgrs	Farmers
Farmers:own account	Farmers
Intermediate non_manual ancill	Non-manual
Intermediate non_manual foreme	Non-manual
Junior non_manual	Non-manual
Managers	Employed
Managers : large establishment	Managers
Managers : small establishment	Managers
Manual	Employed
Manual - foremen\supervisors	Managers
Non-manual	Employed
Own account non_professional	Self-employed
Personal service	Employed
Professional - employee	Employed
Professional - self employed	Self-employed
Semi-skilled manual	Manual
Skilled manual	Manual
Unskilled manual	Manual

which leads to the following taxonomy:

- Employed
  - Agricultural workers
  - Armed forces
  - Personal service
  - Professional - employee
  - Manual
    - Semi-skilled manual
    - Skilled manual
    - Unskilled manual
  - Non-manual
    - Intermediate non_manual ancill
    - Intermediate non_manual foreme
    - Junior non_manual
  - Managers
    - Managers : large establishment
    - Managers : small establishment
    - Manual - foremen\supervisors
- Self-employed
  - Professional - self employed
  - Own account non_professional
  - Employers
    - Employers: large establishment
    - Employers: small establishment
  - Farmers
    - Farmers:own account
    - Farmers:emp&mgrs

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

ND2SO Help Guide

From Native Data to Symbolic Objects

Edited by FUNDP

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

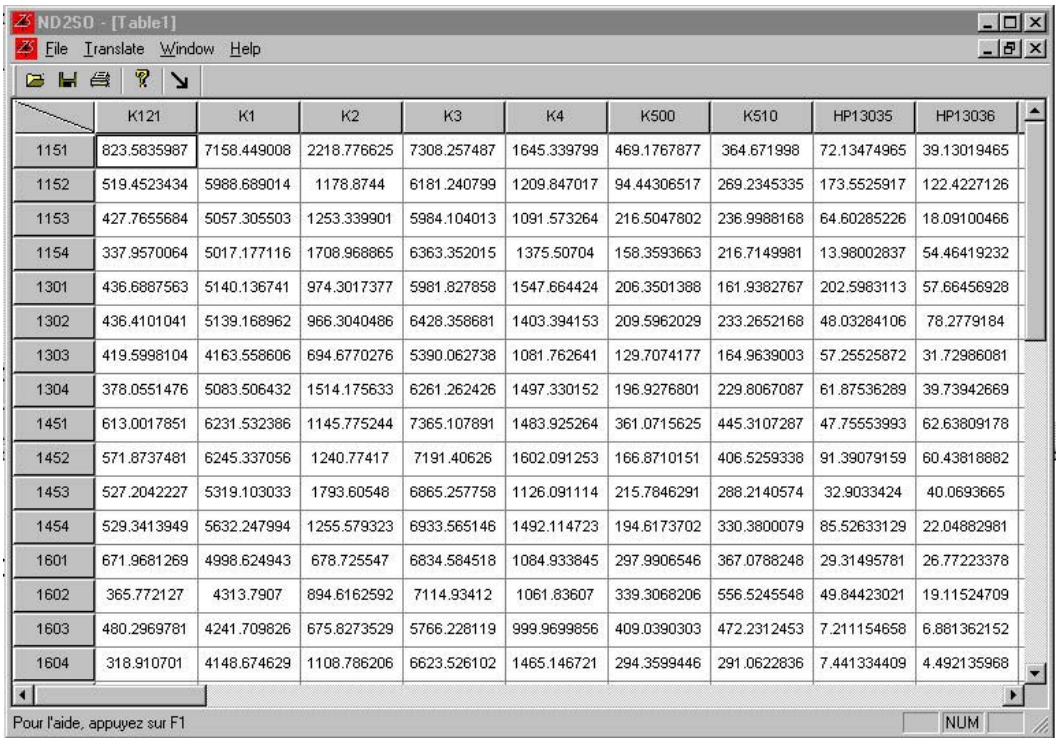
Date: **5/06/2003**

2 ND2SO : From Native Data to Symbolic Objects

2.1 General information

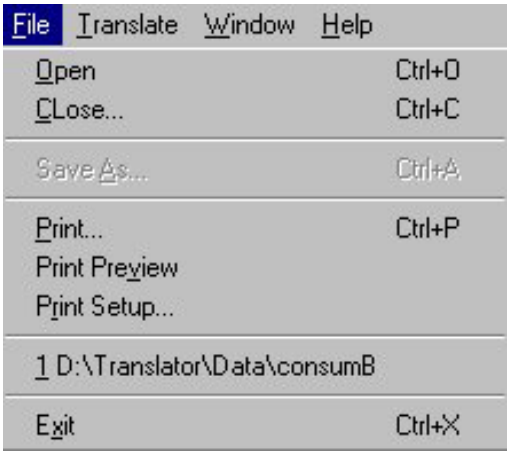
2.2 Typical use

The ND2SO Module allows users to translate ASCII data in symbolic objects data . The following figures present the process used to translate ASCII variables into symbolic variables.



	K121	K1	K2	K3	K4	K500	K510	HP13035	HP13036
1151	823.5835987	7158.449008	2218.776625	7308.257487	1645.339799	469.1767877	364.671998	72.13474965	39.13019465
1152	519.4523434	5988.689014	1178.8744	6181.240799	1209.847017	94.44306517	269.2345335	173.5525917	122.4227126
1153	427.7655684	5057.305503	1253.339901	5984.104013	1091.573264	216.5047802	236.9988168	64.60285226	18.09100466
1154	337.9570064	5017.177116	1708.968865	6363.352015	1375.50704	158.3593663	216.7149981	13.98002837	54.46419232
1301	436.6887563	5140.136741	974.3017377	5981.827858	1547.664424	206.3501388	161.9382767	202.5983113	57.66456928
1302	436.4101041	5139.168962	966.3040486	6428.358681	1403.394153	209.5962029	233.2652168	48.03284106	78.2779184
1303	419.5998104	4163.558606	694.6770276	5390.062738	1081.762641	129.7074177	164.9639003	57.25525872	31.72986081
1304	378.0551476	5083.506432	1514.175633	6261.262426	1497.330152	196.9276801	229.8067087	61.87536289	39.73942669
1451	613.0017851	6231.532386	1145.775244	7365.107891	1483.925264	361.0715625	445.3107287	47.75553993	62.63809178
1452	571.8737481	6245.337056	1240.77417	7191.40626	1602.091253	166.8710151	406.5259338	91.39079159	60.43818882
1453	527.2042227	5319.103033	1793.60548	6865.257758	1126.091114	215.7846291	288.2140574	32.9033424	40.0693665
1454	529.3413949	5632.247994	1255.579323	6933.565146	1492.114723	194.6173702	330.3800079	85.52633129	22.04882981
1601	671.9681269	4998.624943	678.725547	6834.584518	1084.933845	297.9906546	367.0788248	29.31495781	26.77223378
1602	365.772127	4313.7907	894.6162592	7114.93412	1061.83607	339.3068206	556.5245548	49.84423021	19.11524709
1603	480.2969781	4241.709826	675.8273529	5766.228119	999.9699856	409.0390303	472.2312453	7.211154658	6.881362152
1604	318.910701	4148.674629	1108.786206	6623.526102	1465.146721	294.3599446	291.0622836	7.441334409	4.492135968

Main ND2SO window



File menu

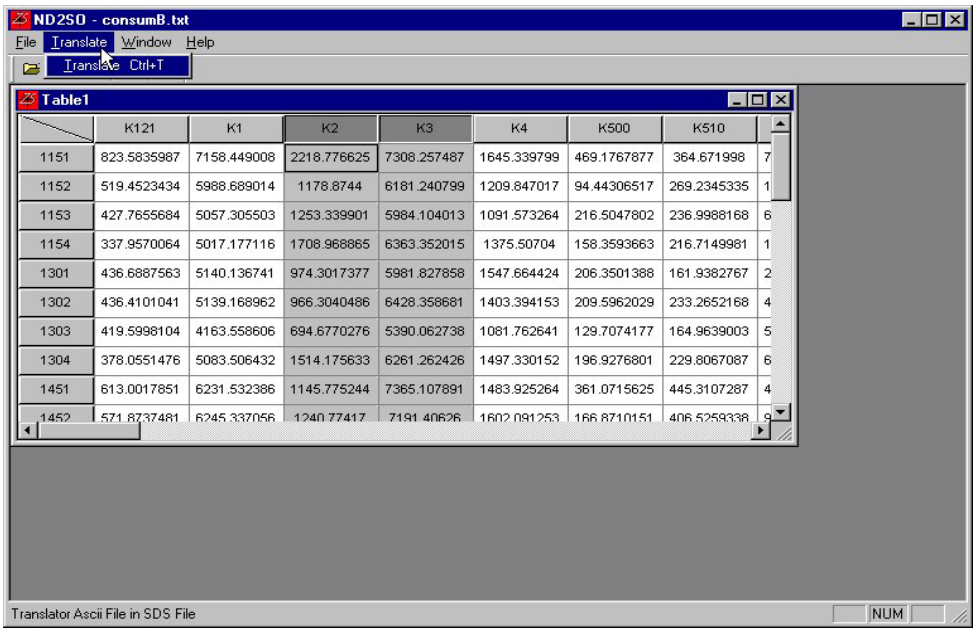
Translate menu

Window menu

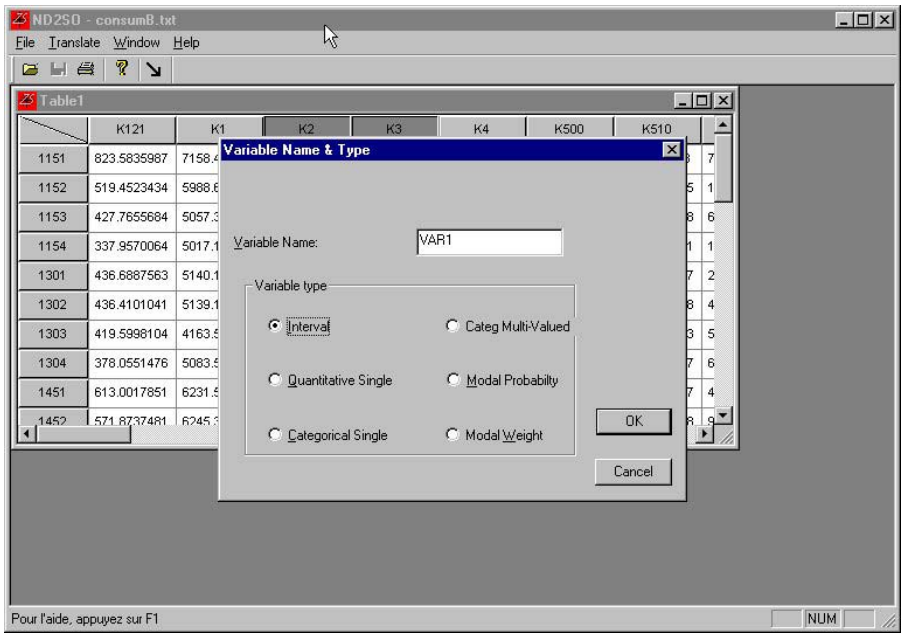
2.3 The « Translate » menu

When a user wants to translate a set of ASCII variables into symbolic variables:

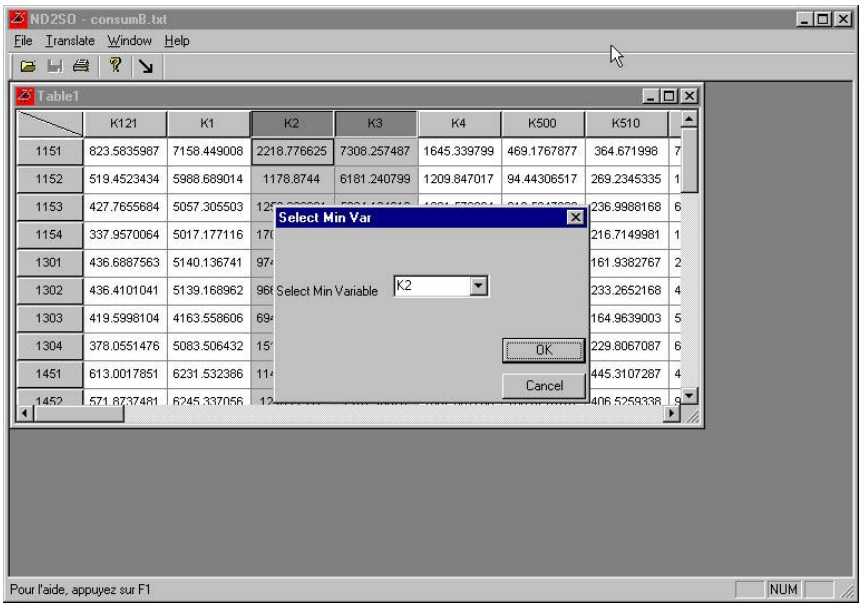
- Select one or more variables according to the type (see Figures)
- Set the name and the type of the new symbolic variable (see Figures)
- If the type is “interval” set the name of the ASCII variable that will be the min (see Figures). If an ASCII variable was already used to make a symbolic variable, you will have a message, which informs you that the variable was already selected (see Figures)
- To make “interval variable”, ASCII values must be numeric and ***min<=max*** for all individuals.
- To make “Quantitative Single”, ASCII values must be numeric.
- To make “Categorical Single”, ASCII values must be integers.
- To make “Categorical Multi-Valued”, ASCII values must be “1” or “0”
- If selected columns are incompatible with the desired type, user will have a message which informs him that selected variable and type are incompatible (see Figures).



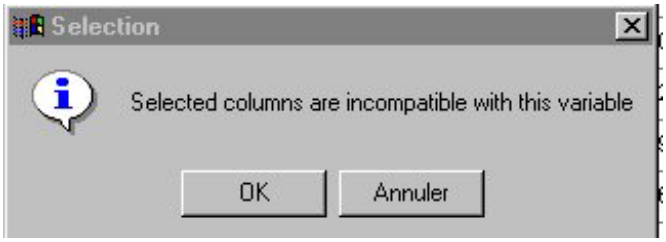
Selection of ASCII variables



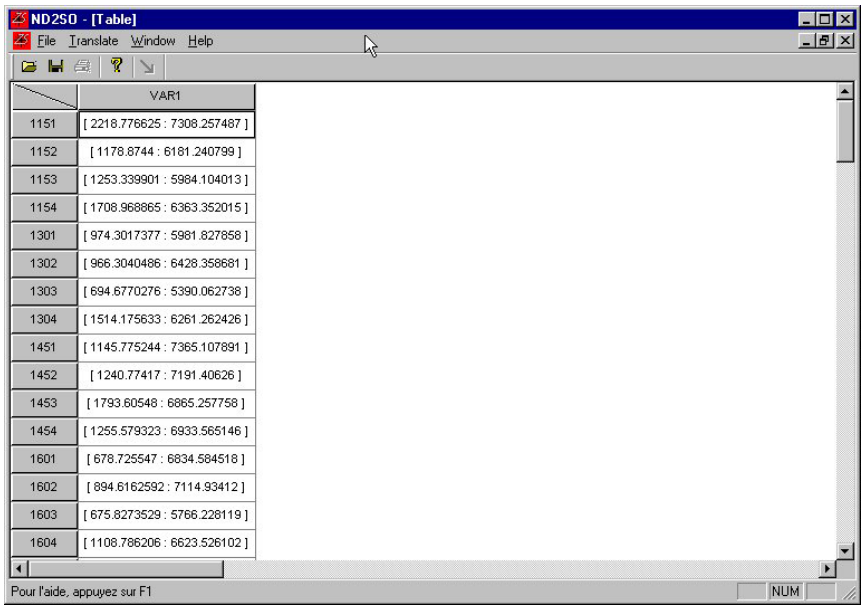
Dialog Box used to select the type of symbolic variables



Dialog box used to select the "Min" value



Message used to inform user that selected variable and type are incompatible



Symbolic object table after translation

Export

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SO2DB Help Guide

From Symbolic Objects to DataBase

Edited by DIB

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **6/06/2003**

3 SO2DB : From Symbolic Objects to DataBase

3.1 Menus

File menu
View menu
Help menu

3.1.1 File menu commands

The File menu offers the following commands:

Open	Opens a Sodas or Xml File.
Exit	Exits So2db

3.1.1.1 Open command (File menu)

Use this command to open an existing sodas/xml file. Use the Window menu to switch among the multiple open documents.

Shortcuts

Toolbar: 

Keys: CTRL+O

File Open dialog box

The following options allow you to specify which file to open:

File Name

Type or select the filename you want to open. This box lists files with the extension you select in the List Files of Type box.

List Files of Type

Select the type of file you want to open:

<< List your application's file types here. >>

Drives

Select the drive in which So2db stores the file that you want to open.

Directories

Select the directory in which So2db stores the file that you want to open.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

3.1.1.2 Exit command (File menu)

Use this command to end your So2db session. You can also use the Close command on the application Control menu. So2db prompts you to save documents with unsaved changes.

Shortcuts

Mouse: Double-click the application's Control menu button.

Keys: ALT+F4

3.1.2 View menu commands

The View menu offers the following commands:

Toolbar	Shows or hides the toolbar.
Status Bar	Shows or hides the status bar.

3.1.2.1 Toolbar command (View menu)

Use this command to display and hide the Toolbar, which includes buttons for some of the most common commands in So2db, such as File Open. A check mark appears next to the menu item when the Toolbar is displayed.

See Toolbar for help on using the toolbar.

Toolbar

The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in So2db,

To hide or display the Toolbar, choose Toolbar from the View menu (ALT, V, T).

<< Add or remove toolbar buttons from the list below according to which ones your application offers. >>

Click To

Open an existing document. So2db displays the Open dialog box, in which you can locate and open the desired file.

3.1.2.2 Status Bar command (View menu)

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed.

See Status Bar for help on using the status bar.

Status Bar

The status bar is displayed at the bottom of the So2db window. To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them. If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right areas of the status bar indicate which of the following keys are latched down:

Indicator	**Description**
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.

3.1.3 Help menu commands

The Help menu offers the following commands, which provide you assistance with this application:

Help Topics	Offers you an index to topics on which you can get help.
About	Displays the version number of this application.

3.1.3.1 Index command (Help menu)

Use this command to display the opening screen of Help. From the opening screen, you can jump to step-by-step instructions for using So2db and various types of reference information.

Once you open Help, you can click the Contents button whenever you want to return to the opening screen.

3.1.3.2 Using Help command (Help menu)

Use this command for instructions about using Help.

3.1.3.3 About command (Help menu)

Use this command to display the copyright notice and version number of your copy of So2db.

Context Help command

Use the Context Help command to obtain help on some portion of So2db. When you choose the Toolbar's Context Help button, the mouse pointer will change to an arrow and question mark. Then click somewhere in the So2db window, such as another Toolbar button. The Help topic will be shown for the item you clicked.

Shortcut

Keys: SHIFT+F1

Title Bar

<< Show your application's title bar here. >>

The title bar is located along the top of a window. It contains the name of the application and document.

To move the window, drag the title bar. Note: You can also move dialog boxes by dragging their title bars.

A title bar may contain the following elements:

- Application Control-menu button
- Document Control-menu button
- Maximize button
- Minimize button
- Name of the application
- Name of the document
- Restore button

Scroll bars

Displayed at the right and bottom edges of the document window. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the document. You can use the mouse to scroll to other parts of the document.

<< Describe the actions of the various parts of the scrollbar, according to how they behave in your application. >>

Size command (System menu)

Use this command to display a four-headed arrow so you can size the active window with the arrow keys.

After the pointer changes to the four-headed arrow:

1. Press one of the DIRECTION keys (left, right, up, or down arrow key) to move the pointer to the border you want to move.
2. Press a DIRECTION key to move the border.
3. Press ENTER when the window is the size you want.

Note: This command is unavailable if you maximize the window.

Shortcut

Mouse: Drag the size bars at the corners or edges of the window.

Move command (Control menu)

Use this command to display a four-headed arrow so you can move the active window or dialog box with the arrow keys.

Note: This command is unavailable if you maximize the window.

Shortcut

Keys: CTRL+F7

Minimize command (application Control menu)

Use this command to reduce the So2db window to an icon.

Shortcut

Mouse: Click the minimize icon  on the title bar.

Keys: ALT+F9

Maximize command (System menu)

Use this command to enlarge the active window to fill the available space.

Shortcut

Mouse: Click the maximize icon  on the title bar; or double-click the title bar.

Keys: CTRL+F10 enlarges a document window.

No Help Available

No help is available for this area of the window.

No Help Available

No help is available for this message box.

<< If you wish to author help specific to each message box prompt, then remove the AFX_HIDP_xxx values from the [ALIAS] section of your .HPJ file, and author a topic for each AFX_HIDP_xxx value. For example, AFX_HIDP_INVALID_FILENAME is the help topic for the Invalid Filename message box. >>

3.2 SO2DB

3.2.1 Introduction

The main goal of SO2DB module is that of retrieving individuals with some characteristics described by a set of SO's in a relational database. Retrieved individuals are stored in a database table.

The SO2DB module interfaces a database where individuals to be retrieved are stored. SOs describing the characteristics of the individuals of interest are stored in a SOML file.

A GUI supports users in the tasks of selecting an input SODAS (or XML) file, from which users select one or more SO's, and retrieving a table for the description of individuals.

The problem of matching individuals against a set of SOs goes beyond what can be done with SQL queries, that is, the problem cannot be solved by simply transforming the description of a SO into a SQL query. The use of canonical/flexible matching operators is required. In particular, the canonical (flexible) matching operator permits the retrieval of individuals, which exactly (approximately) match against some SOs.

3.2.2 Input

SODAS or XML file

3.2.3 Method

SO2DB is based on the relational data model, which provides a simple, yet rigorously defined, description of how users perceive data.

In the relational model, a database is a collection of relational tables. The organisation of data into relational tables is known as the logical view of the database.

A relational table is composed of a set of named columns and an arbitrary number of unnamed rows. The columns of the tables contain information about the attributes. The rows of the table represent occurrences of what is represented by the table. A data value is stored in

the intersection of a row and column. Each named column has a domain, which is the set of values that may appear in that column.

Since the input SOML file does not convey information on the individuals covered by SOs, the database tables used to generate SOs are required. Therefore, individuals covered by a set of SOs are found by matching a set of SOs, named **referents**, against individuals described by one database table, named **subjects**. Matching individuals are stored in one single database table, whose attributes are denoted with the names of the input symbolic variables.

Another application of this method is the propagation on data base, here intended as the possibility of matching some SOs against a set of individuals different from those used to generate the same input SOs. In this way, it is possible to discover new individuals, stored in other databases, such that they show the same properties described by the input SOs.

The problem that we want to solve is the following:
Let B be a SO, whose extention is computed on a set and described by a set of symbolic variables (nominal, interval, etc.). Let ' be another set of individuals (eventually coincident with): we want to calculate the extention of B on ' as well. This operation is not always possible, since variables used to describe the individuals on ' should permit the system to compute the extention of B. An example is reported below.

Let ' be:

maker_name	fuel_type	aspiration	num_of_doors	body_style	drive_wheels
alfa-romeo	Gas	Std	two	convertible	Rwd
alfa-romeo	Gas	Std	two	convertible	Rwd
alfa-romeo	Gas	Std	two	hatchback	Rwd
Audi	Gas	Std	four	sedan	4wd
Audi	Gas	Std	four	wagon	Fwd
Audi	Gas	Turbo	four	sedan	Fwd
Audi	Gas	Turbo	two	hatchback	4wd
Bmw	Gas	Std	two	sedan	Rwd
Bmw	Gas	Std	four	sedan	Rwd

If B is:

[**maker_name** {Audi,Bmw}][**type_fuel** {Gas, Diesel}][**body_style** {wagon,sedan}]

then, we can calculate the extension of B on ' because there exists a logical association between the variables of the SO and some columns of '. The extension of B on ' is:

maker_name	fuel_type	aspiration	num_of_doors	body_style	drive_wheels
Audi	Gas	Std	four	sedan	4wd
Audi	Gas	Std	four	wagon	Fwd
Audi	Gas	Turbo	four	sedan	Fwd
Bmw	Gas	Std	two	sedan	Rwd
Bmw	Gas	Std	four	sedan	Rwd

obtained through the SQL query

SELECT *
FROM Car
WHERE make = ''Audi'' **OR** make = ''Bmw'' **AND** fueltype = ''Gas'' **OR**
fueltype = ''Diesel'' **AND** bodystyle = ''wagon'' **OR** bodystyle = ''sedan''**;**

generated from B upon associating of variable names to column names.

If the set ' is the following:

car name	cylinders	displacement	horsepower	weight
"chevrolet chevelle malibu"	8	307	130	3504
"dodge challenger se"	8	383	170	3563
"chevrolet monte carlo"	8	400	150	3761
"buick estate wagon (sw)"	8	455	225	3086
"toyota corona mark ii"	4	113	95	2372
"plymouth duster"	6	198	95	2833
"amc hornet"	6	199	97	2774
"ford maverick"	6	200	85	2587

There is no correspondence between attribute names and the computation of the extension of B is not possible. The operation that allows to compute the extension of a SO on a set of individuals is called Matching, defined as the process of comparing two or more structures to discover their likeness or differences.

Let us consider a symbolic object A and a tuple b retrieved through a SQL query.

Match(**A**,**b**) testing whether conditions expressed in A are satisfied by b (but not viceversa).

A [**colour** = {black, white}] [**height** =[170, 200]]

b≡

colour	height
black	180

The symbolic object A represents a class description and plays the role of the referent in the matching process, while the row b of the relational table corresponds to the description of an individual plays the role of subject: the problem consists of establishing whether the individual described by b can be considered as an instance of the class described by A.

In its simplest form, matching compares two structures of patterns just for equality. More formally, given a symbolic object and a tuple of relational table, the former describing a class of individuals, the latter an individual, we want to check whether there is a match or not. Such a test is called canonical matching and returns either 0 (failure) or 1 (success).

The system outputs those individuals stored in the database whose canonical matching against selected SO returns 1. This result can be obtained also with an SQL query generated from an SO.

However, this definition might be too strict for real-world problems, because of their inherent vagueness. The presence of symbolic variables, obtained by summarization process, often causes the matching test to fail. Indeed, such a variable is generated through the aggregation operators of SQL (Average, Max, Min, etc.) and could correspond to no tuple in the tables.

For this reason, it becomes necessary to rely on a more flexible definition of matching that aims to compare two descriptions in order to identify their similarities rather than their equality. The result of the flexible matching is a number, within the interval [0, 1], that indicates a degree of matching between a symbolic object and a tuple: in particular, flexible matching between A and b is equal to 1 if the canonical matching returns 1; otherwise, it is a value in [0, 1[. The user may introduce a **threshold** T[0, 1[that represents the degree of flexible matching. The system outputs the individuals of database whose degree of match against the selected SO's is greater than or equal to T.

It is noteworthy that it is not possible to generate an SQL query from an SO when the variables are summarized, or the SO is probabilistic, or the matching is flexible. We see an example, for the flexible matching, that shows the problem above:

Let us consider a SO:

A = [**au_lname** = {Green, Bennet}] [**title** ={ The Busy Executive's Database }] [**pub_name** = { Algodata Infosystem, New Moon Books }]

and the individuals (''):

au_lname	Title	pub_name
Green	The Busy Executive's Database	Algodata Infosystem
Green	You Can Combat Computer Stress!	New Moon Books
Carson	But Is It User Friendly?	Algodata Infosystem
O'Leary	Cooking with Computers	Algodata Infosystem
Straight	Straight Talk About Computers	Algodata Infosystem
Bennet	The Busy Executive's Database	Algodata Infosystem
Dull	Secretes of Silicon Valley	Algodata Infosystem

The query, obtained by A, will retrieve the tuples:

au_lname	Title	Pub_name
Green	The Busy Executive's Database	Algodata Infosystem
Bennet	The Busy Executive's Database	Algodata Infosystem

When we decide to use the flexible matching function, fm, and a threshold T = 0.85, the system retrieves those tuples such that fm > T. Therefore, in the example, we will retrieve the 2nd tuple because fm(A, 2nd tuple) = 0.91 > 0.85 = T and so on. The following table shows the obtained tuples:

au_lname	Title	Pub_name
Green	The Busy Executive's Database	Algodata Infosystem
Green	You Can Combat Computer Stress!	New Moon Books
Bennet	The Busy Executive's Database	Algodata Infosystem

Hence, retrieving of individuals in a database is a more complex operation than simple transforming of an SO in an SQL query. Hence, it is necessary to help the user of the system to retrieve this set of individuals through an appropriate method, that is SO2DB.

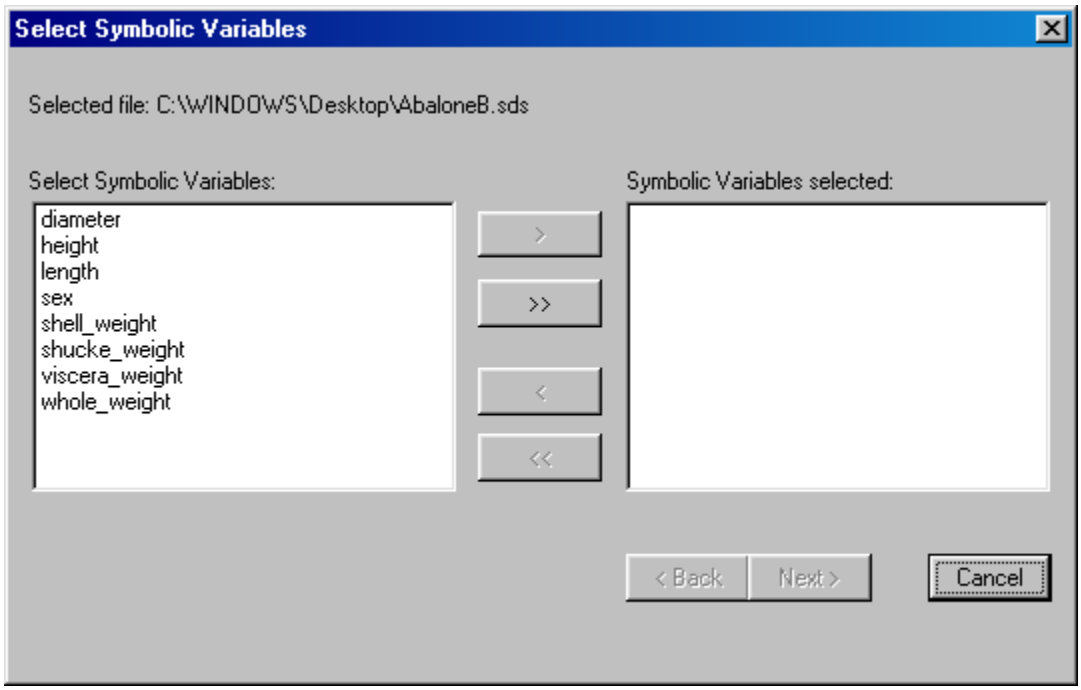
3.2.4 Output

A relational data base table describing the matching individuals.

3.3 GUI

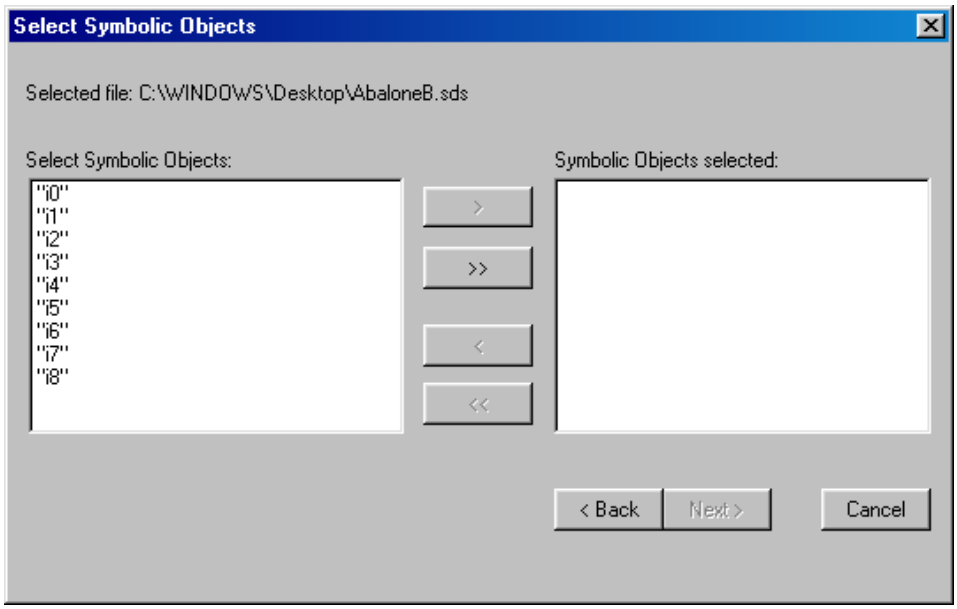
3.3.1 Select Symbolic Variables (GUI)

"Select Symbolic Variables" shows Symbolic Variables in SODAS/XML file selected. The user can select one or more Symbolic Variables.



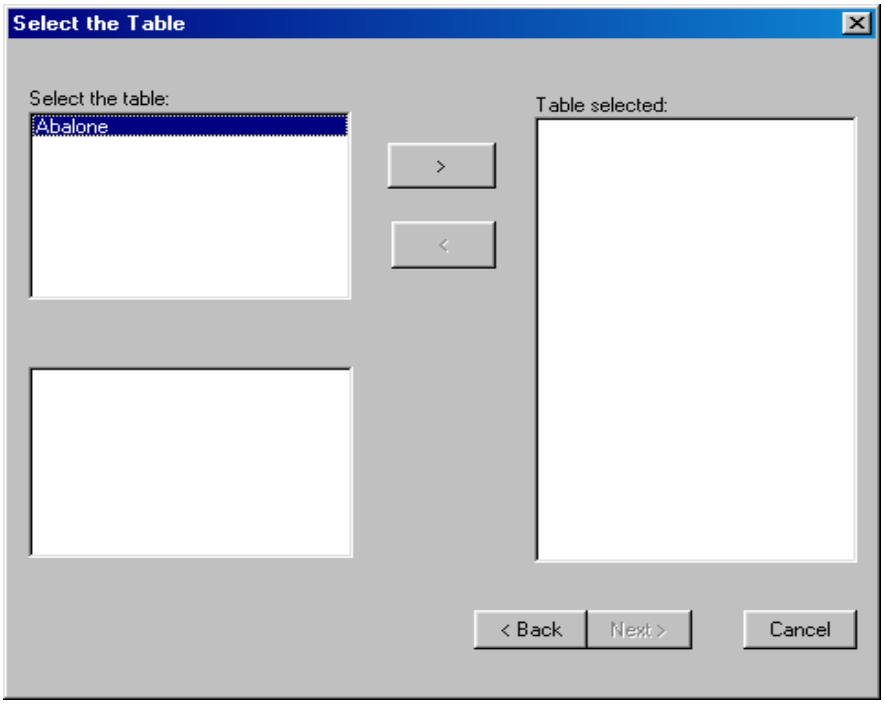
3.3.2 Select Symbolic Objects (GUI)

"Select Symbolic Objects" shows Symbolic Objects in SODAS/XML file selected. The user can select one or more Symbolic Objects that plays a role of referent in matching function.



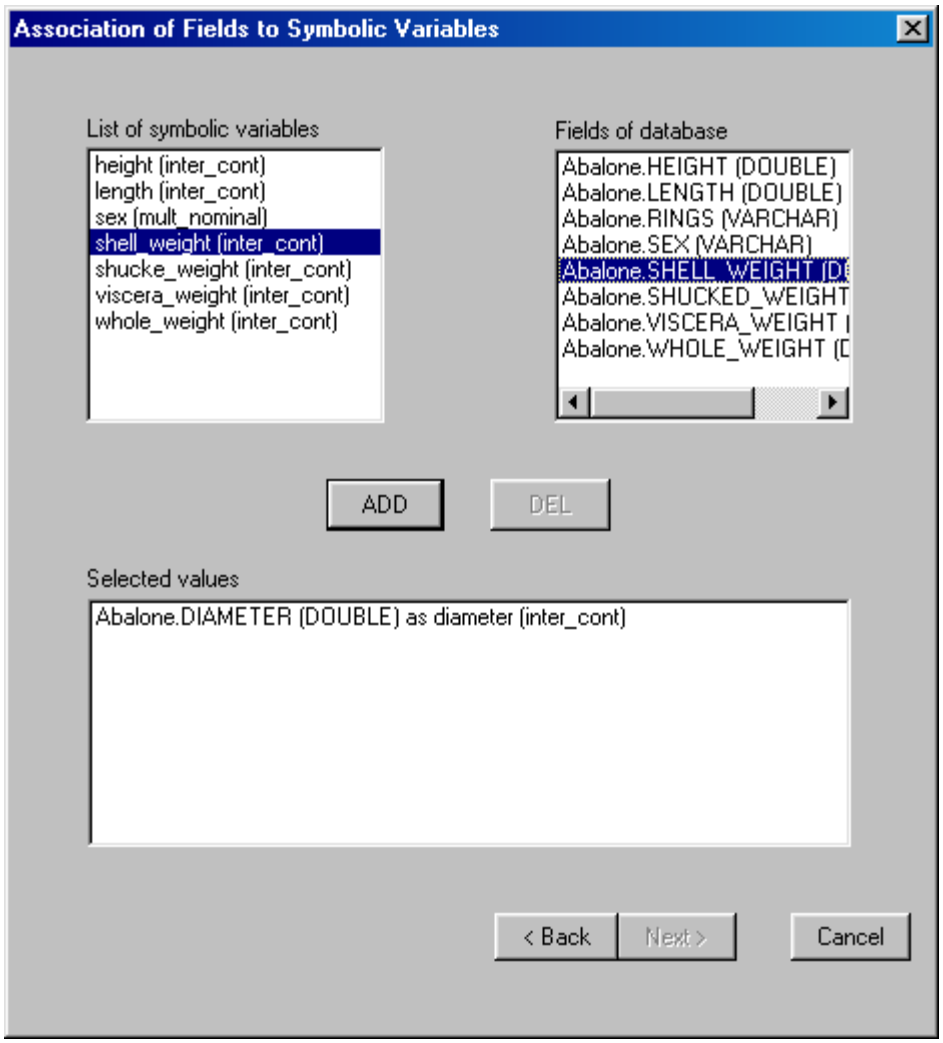
3.3.3 Select the Table (GUI)

The user can select a Table from the connected Database. Only one Table can be selected.



3.3.4 Association of Fields to Symbolic Variables (GUI)

Any Symbolic Variables on the left can be associated with only a Field on the right. The Next push button is Enabled when all the Variables are associated.



3.3.5 Type of Matching (GUI)

Let a, b be two BSO's:

$$a = [x1 = A1] \quad [xn = An]$$

$$b = [x1 = B1] \quad [xn = Bn]$$

There are two type of matching functions:

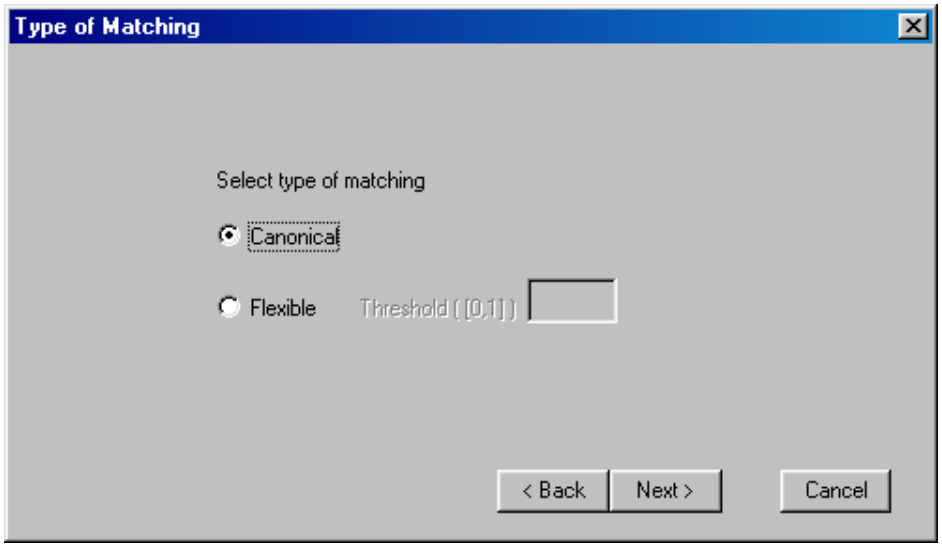
- Canonical Matching
- Flexible Matching

Canonical Matching

Given a symbolic object and a tuple of relational table, the former describing a class of individuals, the latter an individual, we want to check whether there is a match or not. Such a test is called Canonical Matching and returns either 0 (failure) or 1 (success).

canonical-matching(a,b) = true if BiAi for each i=1, 2, , n
canonical-matching (a,b) = false otherwise.

The system outputs those individuals stored in the database whose canonical matching against selected SO returns 1.



Flexible Matching

A more flexible definition of matching aims to compare two descriptions in order to identify their similarities rather than their equality. The result of the Flexible Matching is a number, within the interval [0, 1], that indicates a degree of matching between a symbolic object and a tuple: in particular, flexible matching between A and b is equal to 1 if the canonical matching returns 1; otherwise, it is a value in [0, 1[.

The user may introduce a **Threshold** T[0, 1[that represents the degree of flexible matching. The system outputs the individuals of database whose degree of match against the selected SO's is greater than or equal to T.

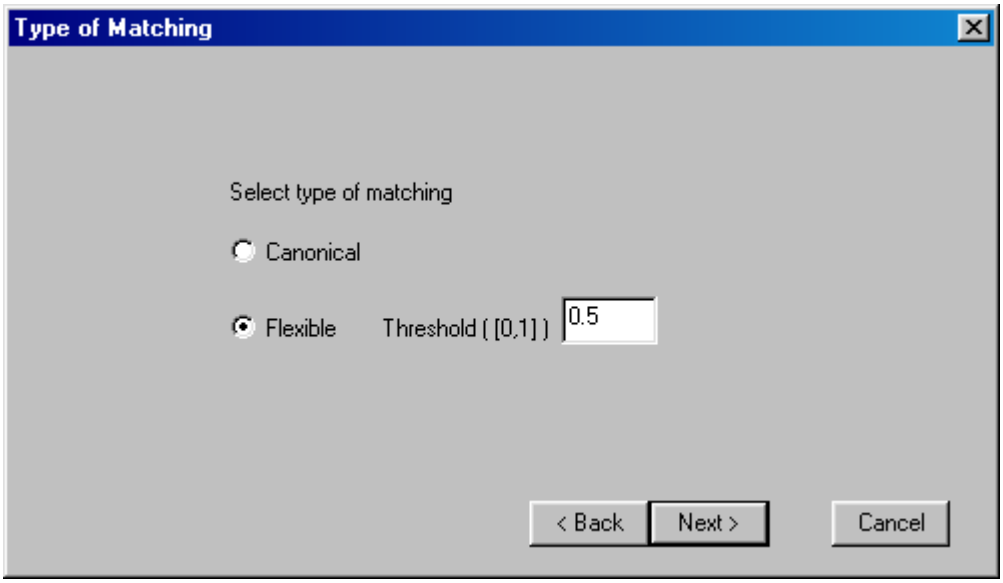
flexible-matching(a,b) =

A further extension of flexible matching is applicable to the case of a pair of two PSO's.
Let a, b be two PSO's; then we can define:

flexible-matching(a,b) =

where $P(b_i')$ is the probability of the symbolic object that plays the role of class while $P(b_{ij})$ is the probability of the other symbolic object.

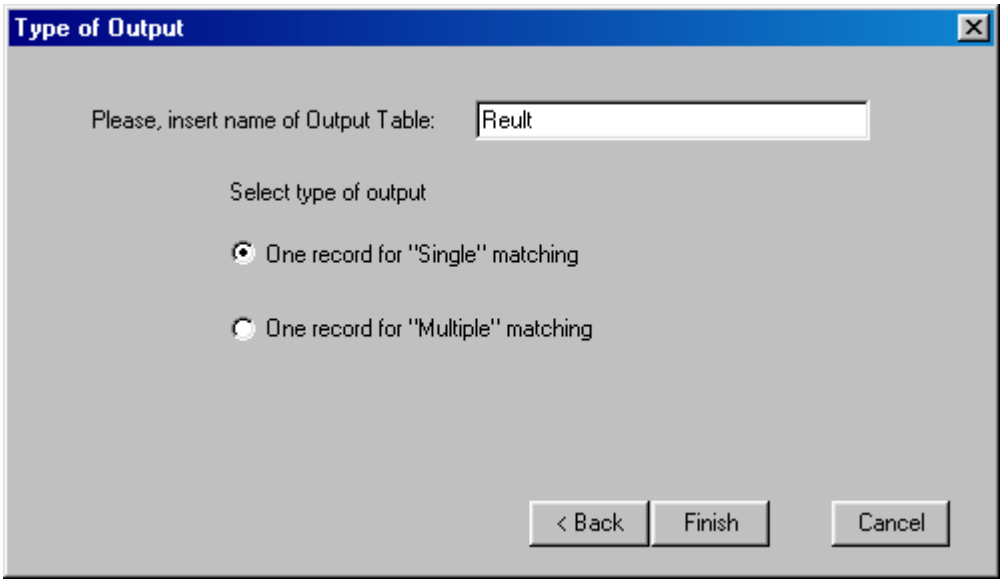
As a particular case, this formula can be also used to compute the flexible matching of a PSO against the description of an individual (i.e., a PSO having single-valued variables).



3.3.6 Type of Output (GUI)

There are two type of matching functions:

- One record for “Single” matching
- One record for “Multiple Matching



3.4 Print

3.4.1 Print command (File menu)

Use this command to print a document. This command presents a Print dialog box, where you may specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.

Shortcuts

Toolbar: 

Keys: CTRL+P

Print dialog box

The following options allow you to specify how the document should be printed:

Printer

This is the active printer and printer connection. Choose the Setup option to change the printer and printer connection.

Setup

Displays a Print Setup dialog box, so you can select a printer and printer connection.

Print Range

Specify the pages you want to print:

All	Prints the entire document.
Selection	Prints the currently selected text.
Pages	Prints the range of pages you specify in the From and To boxes.

Copies

Specify the number of copies you want to print for the above page range.

Collate Copies

Prints copies in page number order, instead of separated multiple copies of each page.

Print Quality

Select the quality of the printing. Generally, lower quality printing takes less time to produce.

Print Progress Dialog

The Printing dialog box is shown during the time that <<YourApp>> is sending output to the printer. The page number indicates the progress of the printing.

To abort printing, choose Cancel.

3.4.2 Print Preview command (File menu)

Use this command to display the active document as it would appear when printed. When you choose this command, the main window will be replaced with a print preview window in which one or two pages will be displayed in their printed format. The print preview toolbar offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job.

Print Preview toolbar

The print preview toolbar offers you the following options:

Print

Bring up the print dialog box, to start a print job.

Next Page

Preview the next printed page.

Prev Page

Preview the previous printed page.

One Page / Two Page

Preview one or two printed pages at a time.

Zoom In

Take a closer look at the printed page.

Zoom Out

Take a larger look at the printed page.

Close

Return from print preview to the editing window.

3.4.3 Print Setup command (File menu)

Use this command to select a printer and a printer connection. This command presents a Print Setup dialog box, where you specify the printer and its connection.

Print Setup dialog box

The following options allow you to select the destination printer and its connection.

Printer
Select the printer you want to use. Choose the Default Printer; or choose the Specific Printer option and select one of the current installed printers shown in the box. You install printers and configure ports using the Windows Control Panel.

Orientation
Choose Portrait or Landscape.

Paper Size
Select the size of paper that the document is to be printed on.

Paper Source
Some printers offer multiple trays for different paper sources. Specify the tray here.

Options
Displays a dialog box where you can make additional choices about printing, specific to the type of printer you have selected.

Network...
Choose this button to connect to a network location, assigning it a new drive letter.

Edit/New

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SOEDIT Help Guide

Symbolic Objects Edition

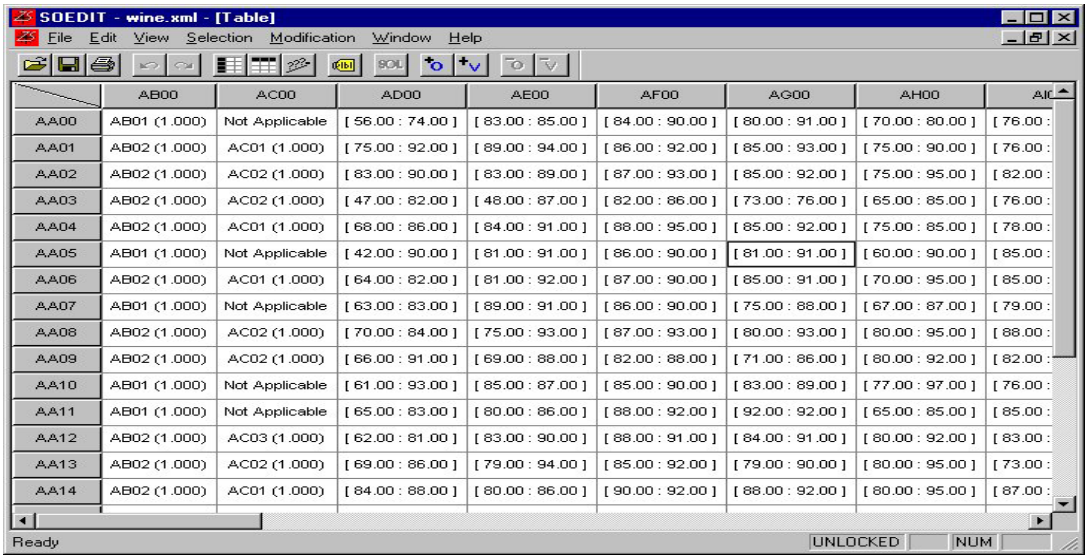
Edited by FUNDP

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

4 SOEDIT : Symbolic Objects Edition

4.1 Execution Procedure

The Symbolic Objects Editor (see Figure 1) allows users to view in a table all the symbolic objects present in a SODAS file and to perform some basic modifications on Symbolic Data file. The editor also provides functionality for viewing the SOL representation of each symbolic object present in the table. We will present each functionality menu by menu an item by item.



	AB00	AC00	AD00	AE00	AF00	AG00	AH00
AA00	AB01 (1.000)	Not Applicable	[56.00 : 74.00]	[83.00 : 85.00]	[84.00 : 90.00]	[80.00 : 91.00]	[70.00 : 80.00]
AA01	AB02 (1.000)	AC01 (1.000)	[75.00 : 92.00]	[89.00 : 94.00]	[86.00 : 92.00]	[85.00 : 93.00]	[75.00 : 90.00]
AA02	AB02 (1.000)	AC02 (1.000)	[83.00 : 90.00]	[83.00 : 89.00]	[87.00 : 93.00]	[85.00 : 92.00]	[75.00 : 95.00]
AA03	AB02 (1.000)	AC02 (1.000)	[47.00 : 82.00]	[48.00 : 87.00]	[82.00 : 86.00]	[73.00 : 76.00]	[65.00 : 85.00]
AA04	AB02 (1.000)	AC01 (1.000)	[68.00 : 86.00]	[84.00 : 91.00]	[88.00 : 95.00]	[85.00 : 92.00]	[75.00 : 85.00]
AA05	AB01 (1.000)	Not Applicable	[42.00 : 90.00]	[81.00 : 91.00]	[86.00 : 90.00]	[81.00 : 91.00]	[60.00 : 90.00]
AA06	AB02 (1.000)	AC01 (1.000)	[64.00 : 82.00]	[81.00 : 92.00]	[87.00 : 90.00]	[85.00 : 91.00]	[70.00 : 95.00]
AA07	AB01 (1.000)	Not Applicable	[63.00 : 83.00]	[89.00 : 91.00]	[86.00 : 90.00]	[75.00 : 88.00]	[67.00 : 87.00]
AA08	AB02 (1.000)	AC02 (1.000)	[70.00 : 84.00]	[75.00 : 93.00]	[87.00 : 93.00]	[80.00 : 93.00]	[80.00 : 95.00]
AA09	AB02 (1.000)	AC02 (1.000)	[66.00 : 91.00]	[69.00 : 88.00]	[82.00 : 88.00]	[71.00 : 86.00]	[80.00 : 92.00]
AA10	AB01 (1.000)	Not Applicable	[61.00 : 93.00]	[85.00 : 87.00]	[85.00 : 90.00]	[83.00 : 89.00]	[77.00 : 97.00]
AA11	AB01 (1.000)	Not Applicable	[65.00 : 83.00]	[80.00 : 86.00]	[88.00 : 92.00]	[92.00 : 92.00]	[65.00 : 85.00]
AA12	AB02 (1.000)	AC03 (1.000)	[62.00 : 81.00]	[83.00 : 90.00]	[88.00 : 91.00]	[84.00 : 91.00]	[80.00 : 92.00]
AA13	AB02 (1.000)	AC02 (1.000)	[69.00 : 86.00]	[79.00 : 94.00]	[85.00 : 92.00]	[79.00 : 90.00]	[80.00 : 95.00]
AA14	AB02 (1.000)	AC01 (1.000)	[84.00 : 88.00]	[80.00 : 86.00]	[90.00 : 92.00]	[88.00 : 92.00]	[80.00 : 95.00]

Figure 1: *The symbolic objects editor window*

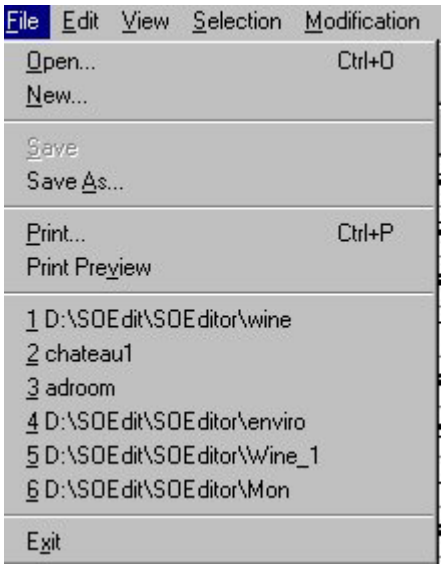


Figure 2: *"File" Menu*

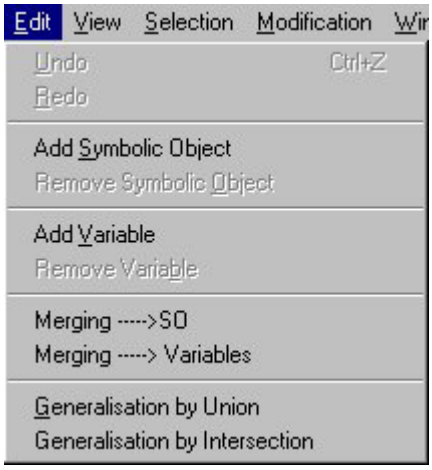


Figure 3: *"Edit" Menu*

Figure 4: *"View" Menu*

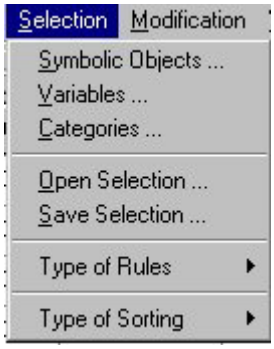


Figure 5: *"Selection" Menu*

Figure 6: *"Window" Menu*

Figure 7: *"Help" Menu*

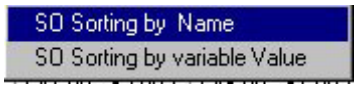


Figure 8:*"Variables Selection Rule by Type" and "SO sorting by Name" menu*

4.2 The «File» menu

The «File» menu corresponds to a classical Windows [TM] File menu. It allows SODAS files to be opened (), closed and saved ().

4.2.1 New

The «NEW» item is used to create a new SOXML file. When user executes this function, he initially defines the number of objects (see Figure 9), the codes and labels for all symbolic objects (see Figure 10).

He must also define code, label and type for the variables (see Figure 11). If variable type is "Categorical Single", "Modal" or "Categorical Multi-Valued", he must also define the number, codes and labels of the categories (see Figure 12 and Figure 13).

After having initialised all the new Symbolic variables objects and modalities, the user must set the values for all SOs, variable by variable.

- For Interval variable, he must give the Min (numeric) and the Max (numeric) value (Min<=Max).
- For Quantitative Single, he must give a numeric value.
- For Modal, he must give a float which is the proportion of the category in the variable (between 0 and 1). The sum of proportions must be equal to 1. If it is not correct, an error message will appear.
- For Categorical Single, he must set the index of the present category.
- For Categorical Multi-Valued, he must set 1 if a category is present and 0 if not.

4.2.2 Save as

The «Save as ...» item saves the content of the table in a SOXML file. It implies that the symbolic objects and the variables present in the table will be the only one to be written in the file (in this case, a warning message will be given to the user), and the order defined in the table will be respected (rules and taxonomies are also updated according to the selected variables). If a new order has been defined for categories, this order will also be taken into account. The user can also save only selected symbolic objects and variable. In this case, only a part of the table is saved in another file (see Figure 14).

4.2.3 Print

The software also allows the content of any window (Table, LOS) to be printed (). The program checks if the selected printer is a colour printer. If not, colours providing distinct greys will be automatically selected by the program. These colours correspond to the colours recommended in the style guide. When printing a star, it is necessary to reduce the size of label characters.

For example, choose 5 for variable and categories labels.

The «File» menu also contains the list of the last opened files.

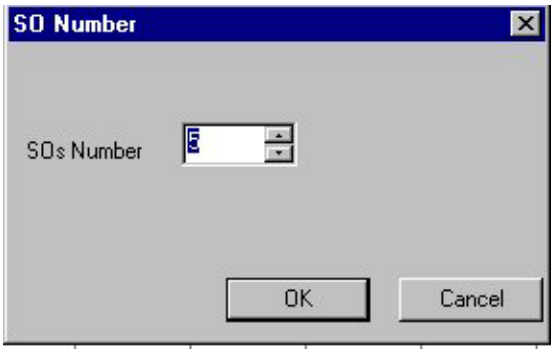


Figure 9: *Dialog box used to set number of SO*

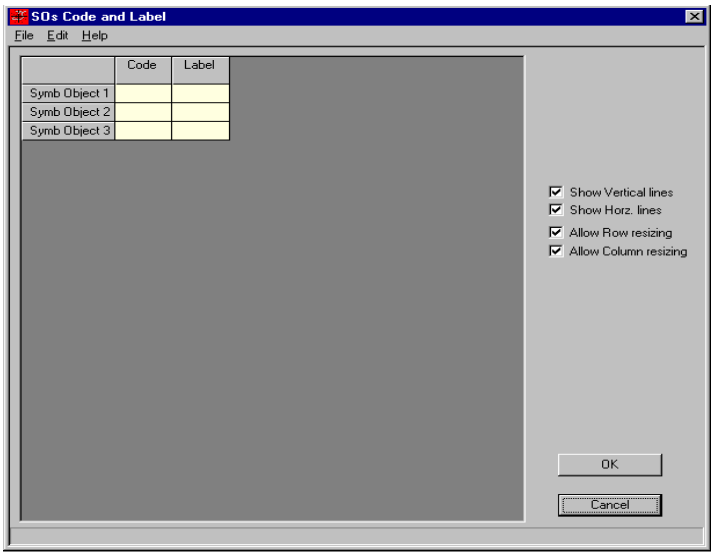


Figure 10: *Dialog box used to set codes and labels SO*

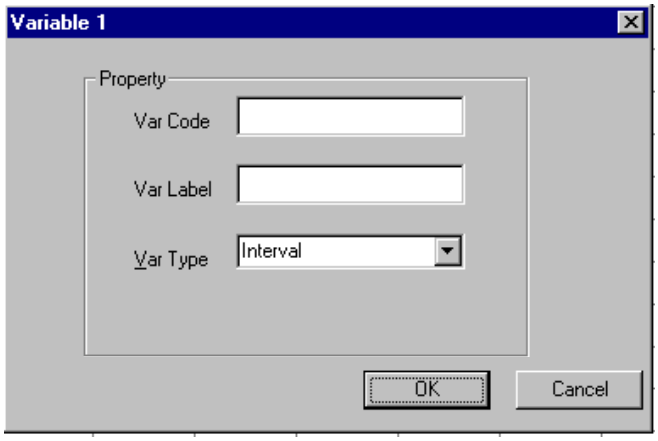


Figure 11: *Dialog box used to set variable code, label and type*

Figure 12: *Number of categories*

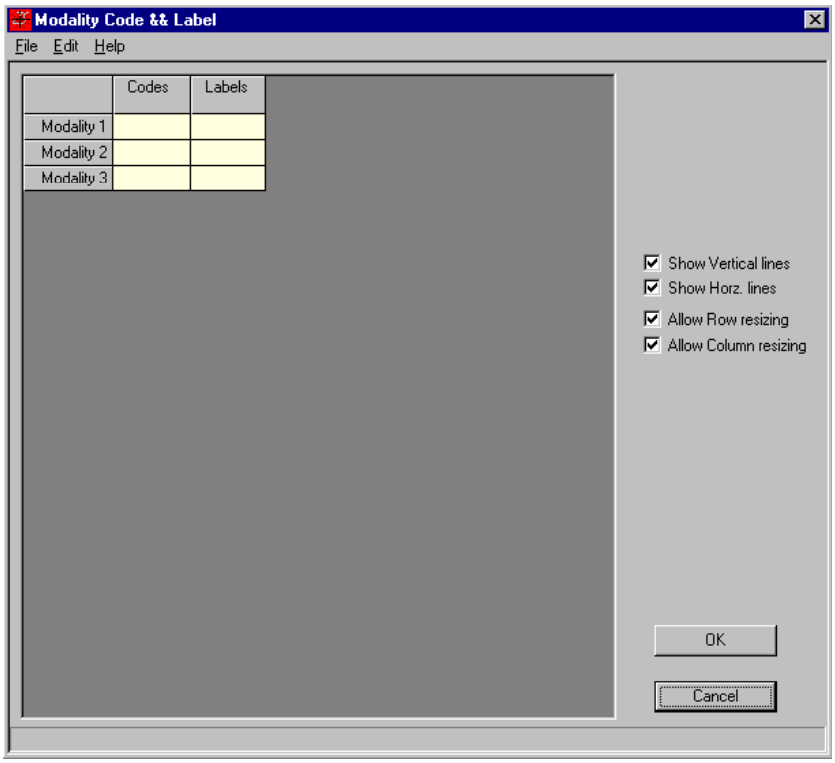


Figure 13: *Modalities code and label*

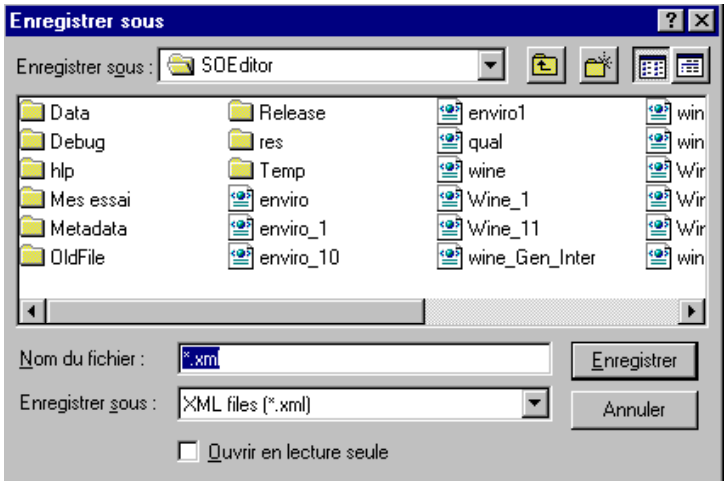


Figure 14: *"Save as" Dialog box*

4.3 The «Edit» menu

The «Edit» menu contains the «Undo» ( ) and «Redo» ( ) functions. These functions only apply for modifications done inside the table.

4.3.1 Copy

The «Copy» item allows the content of any window to be copied to the clipboard. So, the image can be paste inside any other application (word processor, image processor ...).

4.3.2 Remove Symbolic Object

The «***Remove Symbolic Object***» («Remove variables») item removes all selected symbolic objects (all selected variables). These elements are also removed from the original SODAS file. User is invited to create a new SODAS File.

4.3.3 Add Symbolic Object

The «***Add Symbolic Object***» item adds a new Symbolic Object in the displayed table. When this function is executed, a new Symbolic object is added in the table, and the user is asked to modify values of the added Symbolic Object.

The user has to give the label and code of the new symbolic object in the corresponding field. In order to help the user in the fulfilment of the cells values of the first SO of the table are given. User has to modify them accordingly to what he wants but respecting the format.

4.3.4 Add variable

The «***Add Variable***» item is used to add a new Variable in the displayed table. When this function is executed, user is asked to give name, label and type of the new variable. At end,

his is asked to give values of the added variable (see function «**new**») and to create a new SODAS file.

1. Give the type of the new variable(see figure 15)
2. Give the code and label of the new variable (see figure 16)
3. Set the values for the new variable(see figure 17)

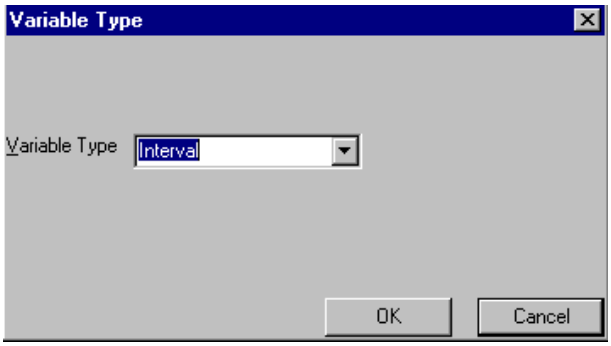


Figure 15: *Variable type dialog box*

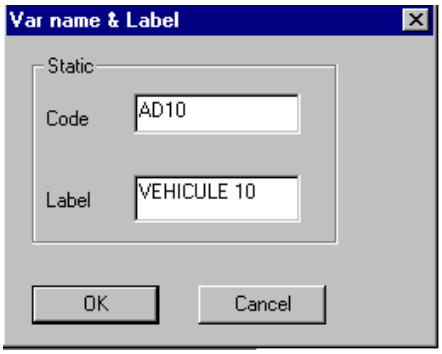


Figure 16: *Variable label and code dialog box*

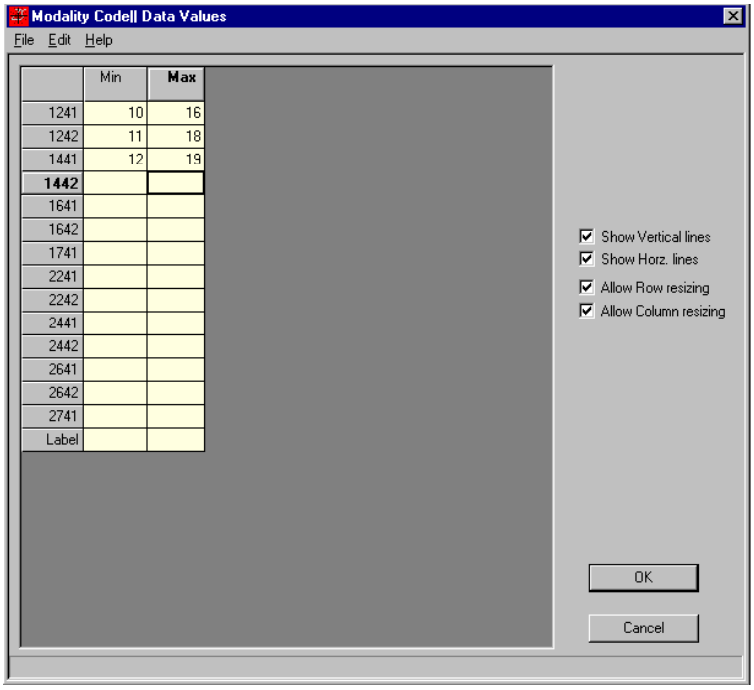


Figure 17: *Set values for the new variable*

4.3.5 Generalisation

The «***Generalisation***» items (by union and by intersection) are used to generalize selected symbolic objects. To activate this function user must select at least two Symbolic Objects.

4.3.6 Merging

The ***«Merging SO»*** (***«Merging Variable»***) allows the merging of SODAS files and makes a new SODAS File. The two files must contain same variables for **Merging SO** and same Symbolic Objects for **Merging Variable**.

4.3.7 Modify

To modify the content of a cell, you have to click on the cell. If the variable is Interval, you have just to modify the value in the cell like in EXCEL. If the variable is Categorical or Modal, the type of categories is reminded in a separate window.

4.4 The «View» menu

In the SO Editor application, two distinct types of window are available. The table containing the Symbolic Objects and the variables located in the SOXML file and the SOL (Symbolic Object Language) description of a Symbolic Object. The first items of the «View» menu correspond to these different windows.

4.4.1 The «Table» item

The «Table» item sends the table window to the front, i.e. it displays the table when the window is hidden by other windows.

4.4.2 The «SOL» item

The «SOL» item displays the SOL description of all Symbolic objects selected in the table. The active window must be the table and objects have to be selected (see Figure 18) to enable this item.

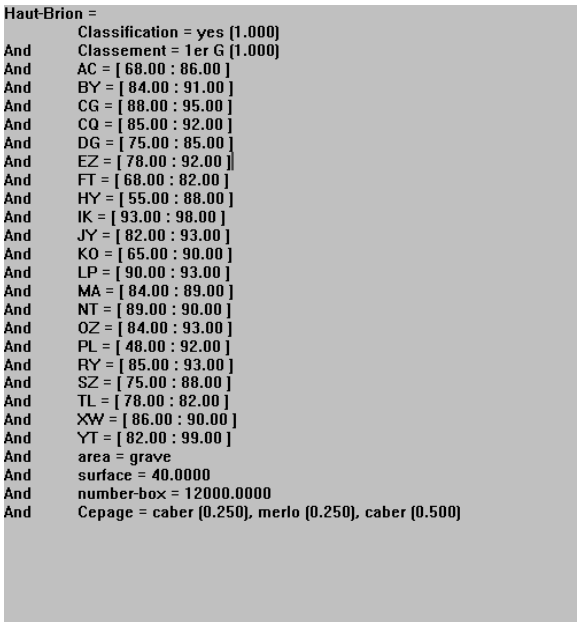


Figure 18: *SOL for a SO*

4.4.3 The «Tool bar» and «Status bar» items

The «Tool bar» and «Status bar» items allow display/hide respectively the status bar located in the bottom of the screen and the tool bar containing all shortcut icons.

4.4.4 The «Labels ...» item

The «Labels...» item makes appear a dialogue box (see Figure 19) which provides the user with the opportunity to work with the first letters of labels instead of the identifiers for Symbolic Objects, variables, and categories. The maximum length of labels can be defined for each element. The chosen options are saved.

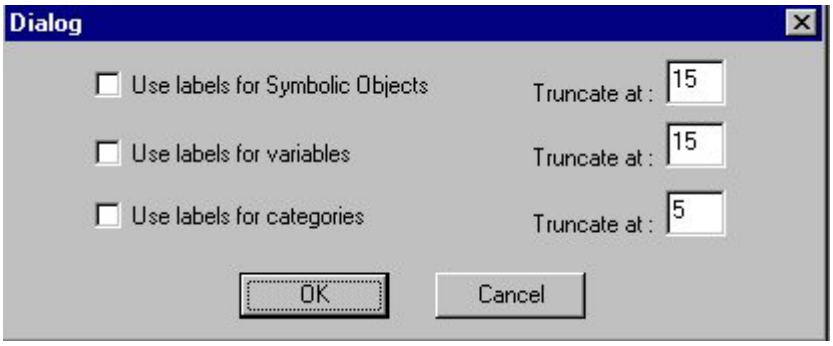


Figure 19: *The "Label Dialog Box"*

4.4.5 The Taxonomy item

The Taxonomy item displays the list of taxonomical variables. When selecting one of these variables, the corresponding taxonomy is displayed.

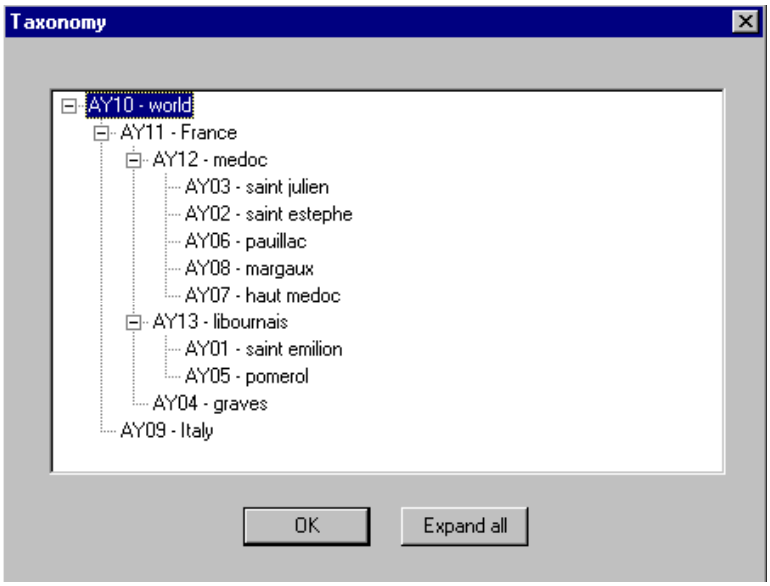


Figure 20: *Example of taxonomy display*

4.4.6 The Hierarchical Dependency item

The Hierarchical Dependency item displays the variables with hierarchical relationship.

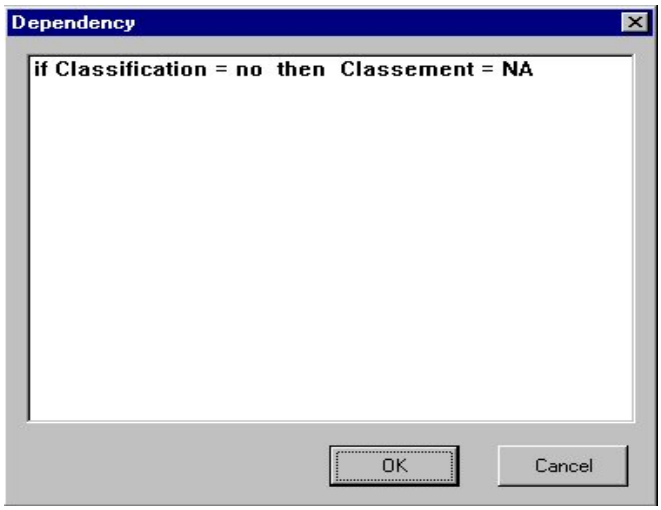


Figure 21: *Example of dependency display*

4.5 The «Selection» menu

The «Selection» menu allows the user to select the Symbolic Objects, the variables and the categories that he wants to see in the table. It also allows a configuration to be saved/loaded. The selection is made by way of a dialog box containing three « tags ». The first one corresponds to Symbolic Objects (see Figure 22), the second one corresponds to variables (see Figure 23), and the third one corresponds to Categories (see Figure 24).

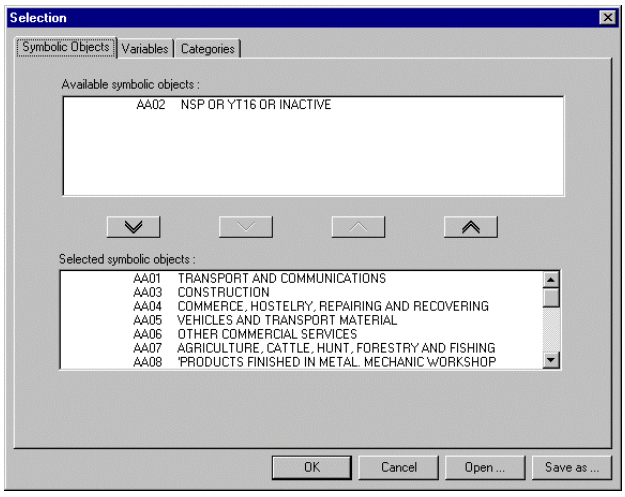


Figure 22: *Symbolic objects selection*

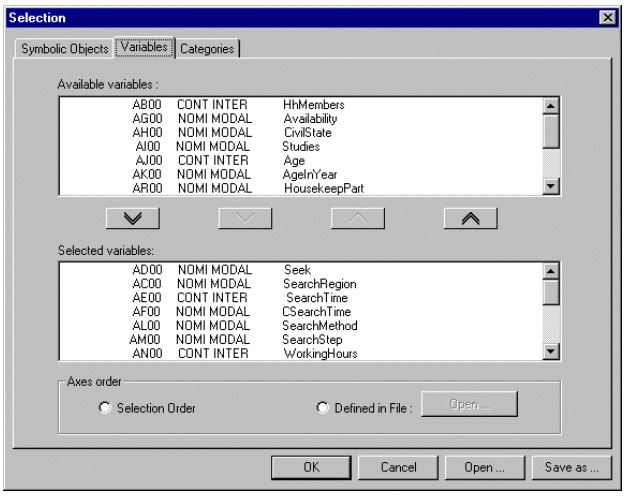


Figure 23: *Variables selection*

In the case of symbolic objects and variables, each «tag» contains two lists. The list located on the bottom of the «tag» contains the selected elements. Elements can be added and/or removed by using the four buttons. The order defined in the list will be respected in the table.

On the bottom of the dialog box, there are two buttons which can be used to open and to save a configuration. The configuration corresponds to choices made for symbolic objects, variables and categories.

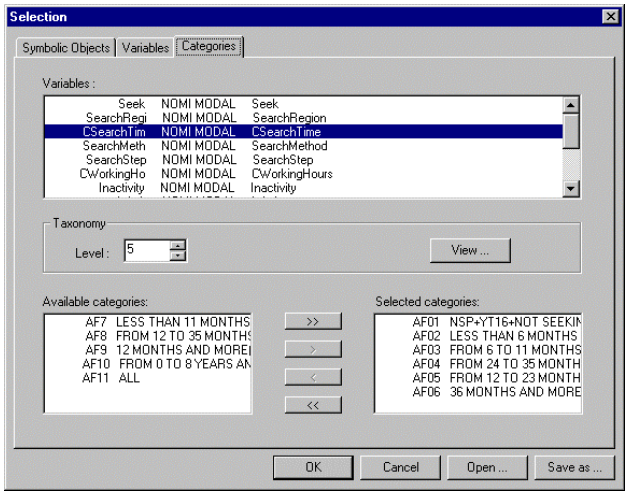


Figure 24: *Categories selection*

In the «tag» corresponding to categories selection, all categorical variables which are currently selected are placed in the list located on the top, and the selected categories are located in the lower right list. Categories, of the selected variable, can be added and/or removed by using the four buttons. The order defined in the list will be respected in the SO table.

4.5.1 The «Select So...» item

The «Select So...» item () displays the selection dialog box and activates the «tag» corresponding to Symbolic Objects selection (see Figure 22).

4.5.2 The «Select Variables...» item

The «Select Variables...» item () displays the selection dialog box and activates the «tag» corresponding to variables selection (see Figure 23).

4.5.3 The «Select Categories...» item

The «Select Categories...» item () displays the selection dialog box and activates the «tag» corresponding to categories selection (see Figure 24).

4.5.4 The «Open Selection...» item

The «Open Selection...» item displays a dialog box which allows a file containing a configuration to be opened. This file has a «.ovc» extension.

4.5.5 The «Save Selection...» item

The «Save Selection...» item displays a dialog box which allows the current configuration to be saved in a file. This file has a «.ovc» extension.

4.5.6 The «Variable Selection rule by type» item

This item displays a dialog box which allows to create another file with all the variables of a chosen type (see Figure 25).

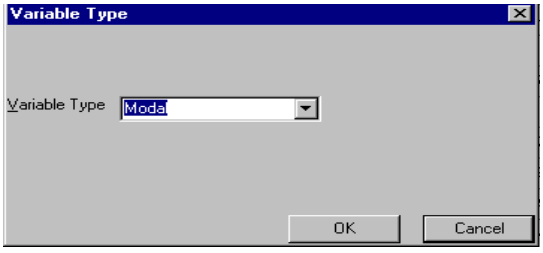


Figure 25: *Variable selection by type*

4.5.7 The «Objects Selection Rule by Value» item

This item displays a dialog box which allows user to give parameters (see Figure 26) for the selection of Symbolic objects according to values of one variable (Quantitative Single or Interval). This item is activated only if the file contains at least one Quantitative single variable or one Interval variable. Per default min of max and max of max is displayed.

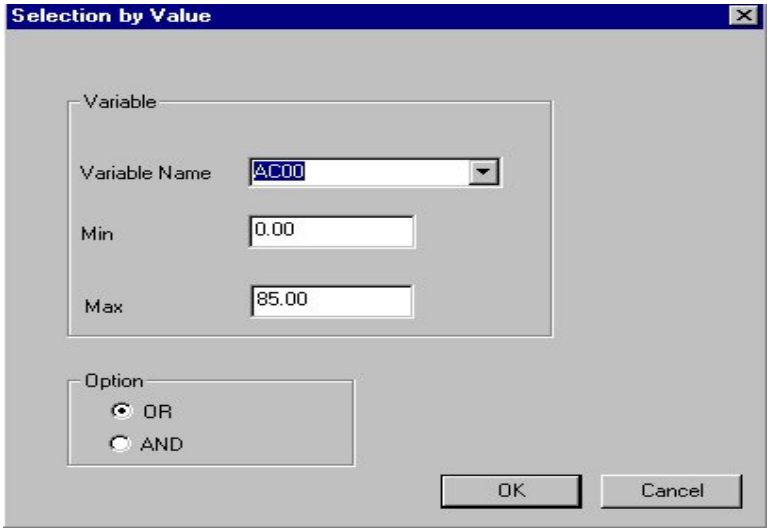


Figure 26: *"SO Selection rule by value" Dialog Box*

4.5.8 The «Objects Selection Rule by Category» item

This item displays a dialog box which allows user to give parameters (see Figure 28) used to select Symbolic object according to present category of one variable (Categorical single and Categorical Multi-valued). This item is activated only if the file contains at least one

Categorical Single variable or one Categorical Multi-Valued. The example of figure 28 selects all the objects such that variable AY00 has categories AY03, AY05, AY07.

4.5.9 The «SO Sorting by Label» item

This item allows sorting Symbolic object by SO Labels.

4.5.10 The «SO Sorting by Value» item

This item displays a dialog box which allows user to give parameters used to sort Symbolic objects according to values of one variable (Quantitative Single). This item is activated only if the file contains at least one Quantitative Single variable (see Figure 27).

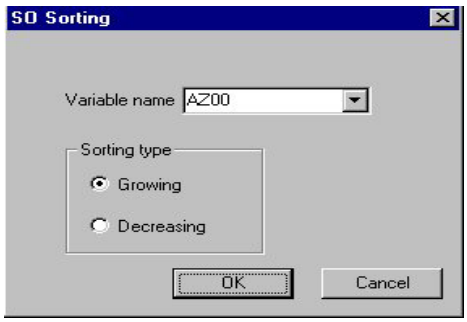


Figure 27: *"SO Sorting by value" Dialog Box*

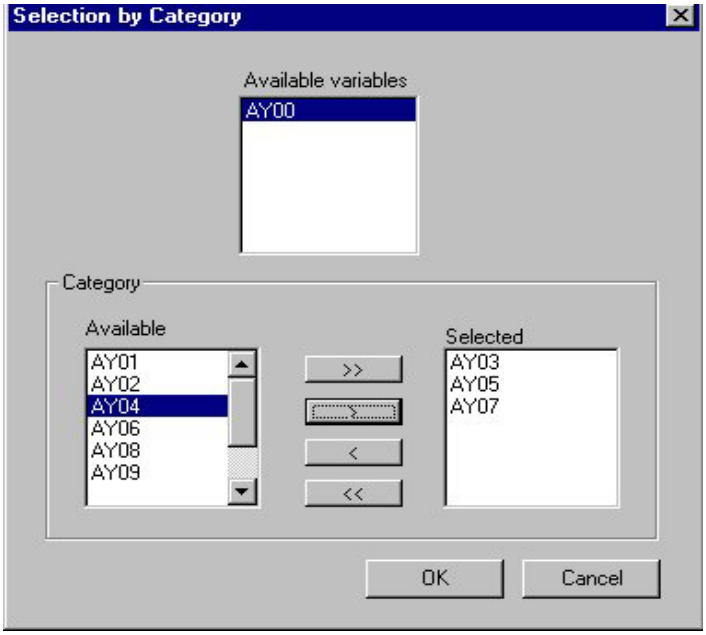


Figure 28: *SO Selection rule by category*

4.6 The «Window» menu

The «Window» menu is a standard Windows™ menu making the windows managing easier.

4.6.1 The «Cascade» item

The «Cascade» item displays windows the one behind the others.

4.6.2 The «Tile» item

The «Tile» item displays windows so that they are all completely visible (see Figure 29).

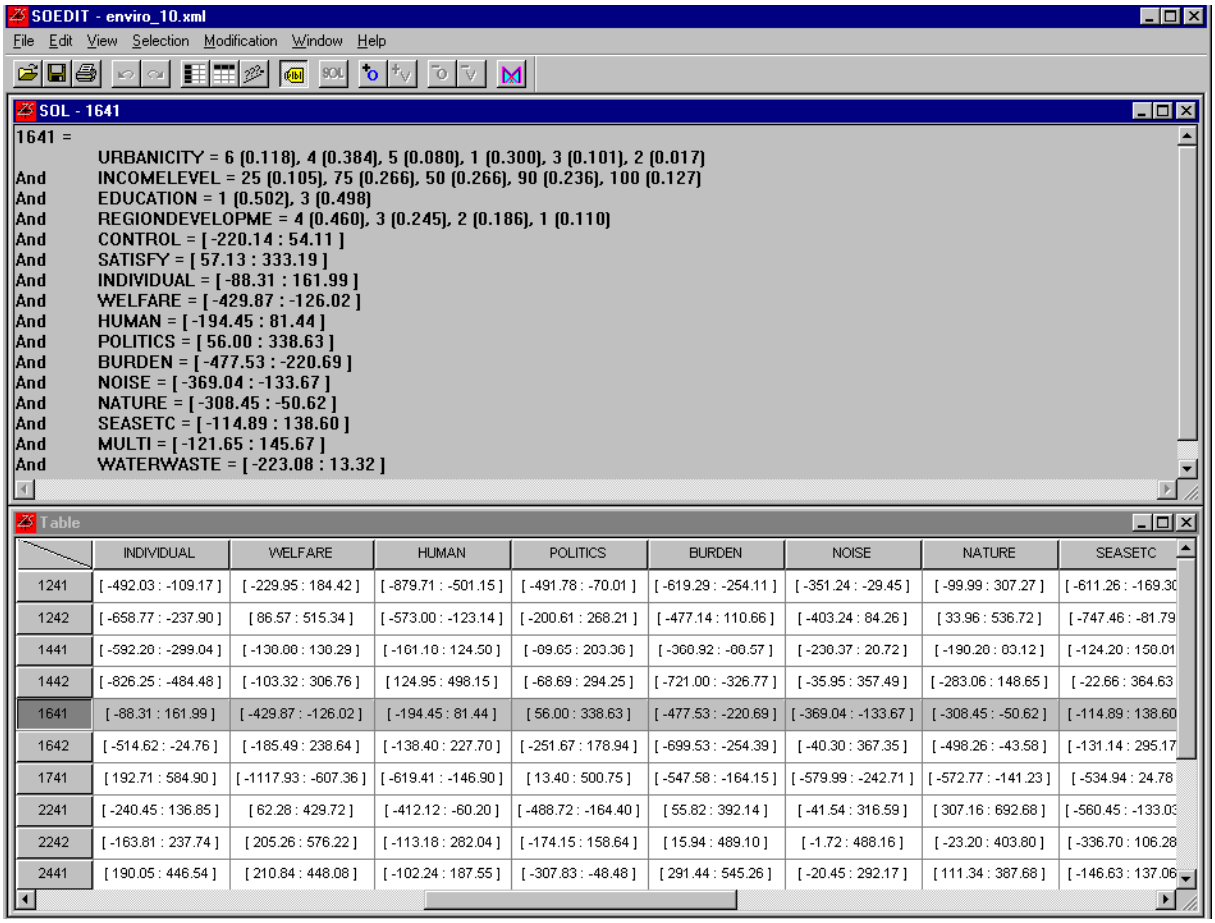
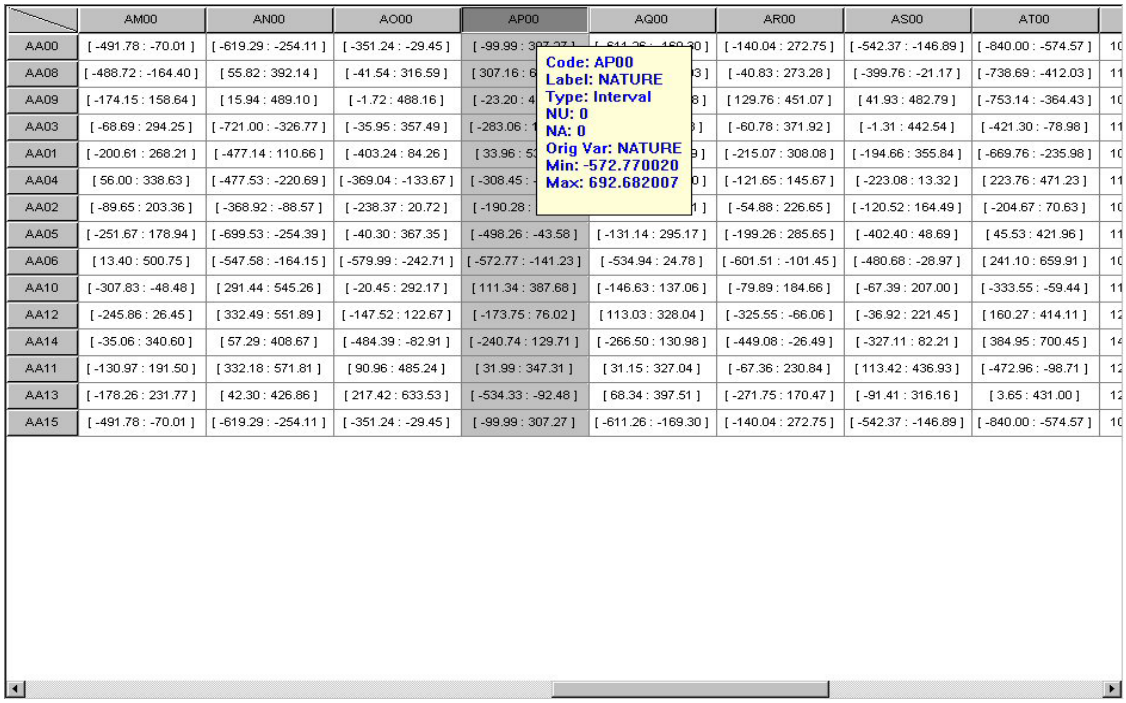


Figure 29: *Example of Windows in tile*

4.7 Metadata display

The metadata can be displayed. They concern the metadata which have been recorded for the Symbolic Data file.

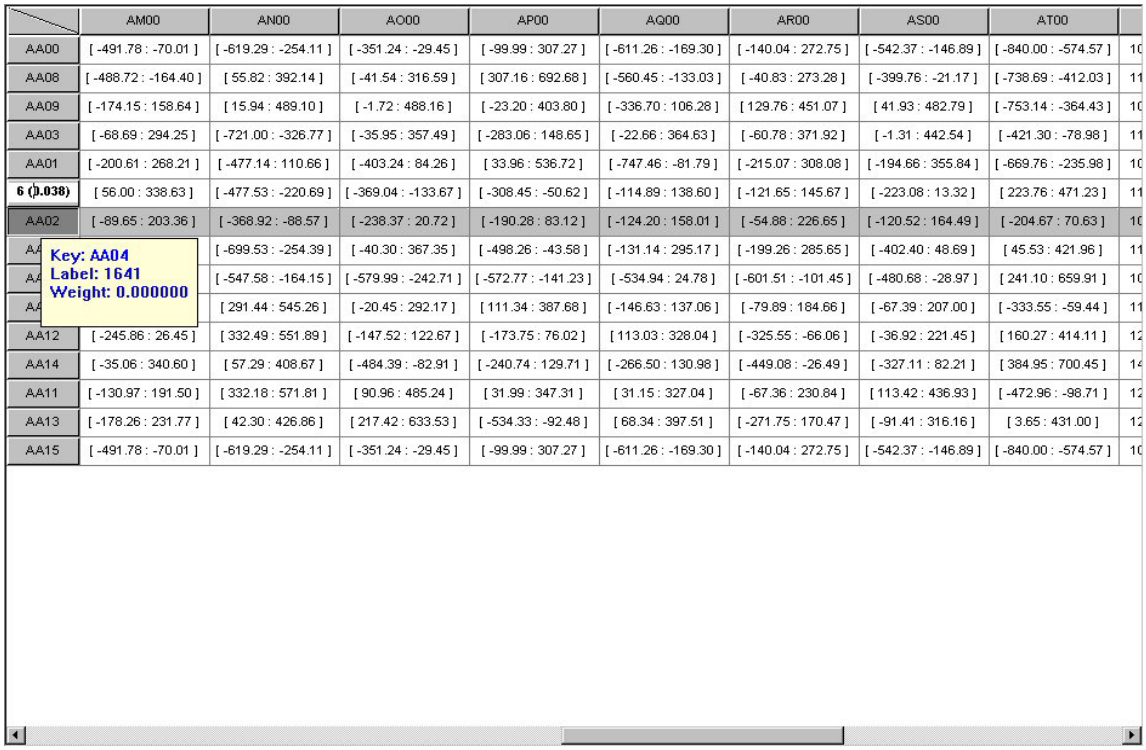
Clicking with the mouse right button on the top cell of a column in the table will display the characteristics of the corresponding variable (see Figure 30).



	AM00	AN00	AO00	AP00	AQ00	AR00	AS00	AT00	
AA00	[-491.78 : -70.01]	[-619.29 : -254.11]	[-351.24 : -29.45]	[-99.99 : 307.27]	[-611.26 : -169.30]	[-140.04 : 272.75]	[-542.37 : -146.89]	[-840.00 : -574.57]	10
AA08	[-488.72 : -164.40]	[55.82 : 392.14]	[-41.54 : 316.59]	[307.16 : 692.68]	[-560.45 : -133.03]	[-40.83 : 273.28]	[-399.76 : -21.17]	[-738.69 : -412.03]	11
AA09	[-174.15 : 158.64]	[15.94 : 489.10]	[-1.72 : 488.16]	[-23.20 : 403.80]	[-336.70 : 106.28]	[129.76 : 451.07]	[41.93 : 482.79]	[-753.14 : -364.43]	10
AA03	[-68.69 : 294.25]	[-721.00 : -326.77]	[-35.95 : 357.49]	[-283.06 : 148.65]	[-22.66 : 364.63]	[-60.78 : 371.92]	[-1.31 : 442.54]	[-421.30 : -78.98]	11
AA01	[-200.61 : 268.21]	[-477.14 : 110.66]	[-403.24 : 84.26]	[33.96 : 536.72]	[-747.46 : -81.79]	[-215.07 : 308.08]	[-194.66 : 355.84]	[-669.76 : -235.98]	10
AA04	[56.00 : 338.63]	[-477.53 : -220.69]	[-369.04 : -133.67]	[-308.45 : -50.62]	[-114.89 : 138.60]	[-121.65 : 145.67]	[-223.08 : 13.32]	[223.76 : 471.23]	11
AA02	[-89.65 : 203.36]	[-368.92 : -88.57]	[-238.37 : 20.72]	[-190.28 : 83.12]	[-124.20 : 158.01]	[-54.88 : 226.65]	[-120.52 : 164.49]	[-204.67 : 70.63]	10
AA05	[-251.67 : 178.94]	[-699.53 : -254.39]	[-40.30 : 367.35]	[-498.26 : -43.58]	[-131.14 : 295.17]	[-199.26 : 285.65]	[-402.40 : 48.69]	[45.53 : 421.96]	11
AA06	[13.40 : 500.75]	[-547.58 : -164.15]	[-579.99 : -242.71]	[-572.77 : -141.23]	[-534.94 : 24.78]	[-601.51 : -101.45]	[-480.68 : -28.97]	[241.10 : 659.91]	10
AA10	[-307.83 : -48.48]	[291.44 : 545.26]	[-20.45 : 292.17]	[111.34 : 387.68]	[-146.63 : 137.06]	[-79.89 : 184.66]	[-67.39 : 207.00]	[-333.55 : -59.44]	11
AA12	[-245.86 : 26.45]	[332.49 : 551.89]	[-147.52 : 122.67]	[-173.75 : 76.02]	[113.03 : 328.04]	[-325.55 : -66.06]	[-36.92 : 221.45]	[160.27 : 414.11]	12
AA14	[-35.06 : 340.60]	[57.29 : 408.67]	[-484.39 : -82.91]	[-240.74 : 129.71]	[-266.50 : 130.98]	[-449.08 : -26.49]	[-327.11 : 82.21]	[384.95 : 700.45]	14
AA11	[-130.97 : 191.50]	[332.18 : 571.81]	[90.96 : 485.24]	[31.99 : 347.31]	[31.15 : 327.04]	[-67.36 : 230.84]	[113.42 : 436.93]	[-472.96 : -98.71]	12
AA13	[-178.26 : 231.77]	[42.30 : 426.86]	[217.42 : 633.53]	[-534.33 : -92.48]	[68.34 : 397.51]	[-271.75 : 170.47]	[-91.41 : 316.16]	[3.65 : 431.00]	12
AA15	[-491.78 : -70.01]	[-619.29 : -254.11]	[-351.24 : -29.45]	[-99.99 : 307.27]	[-611.26 : -169.30]	[-140.04 : 272.75]	[-542.37 : -146.89]	[-840.00 : -574.57]	10

Figure 30: *Metadata for a variable*

Clicking on the left cell of a line in the table will display the characteristics of the corresponding Symbolic Object (see Figure 31).



	AM00	AN00	AO00	AP00	AQ00	AR00	AS00	AT00	
AA00	[-491.78 : -70.01]	[-619.29 : -254.11]	[-351.24 : -29.45]	[-99.99 : 307.27]	[-611.26 : -169.30]	[-140.04 : 272.75]	[-542.37 : -146.89]	[-840.00 : -574.57]	10
AA08	[-488.72 : -164.40]	[55.82 : 392.14]	[-41.54 : 316.59]	[307.16 : 692.68]	[-560.45 : -133.03]	[-40.83 : 273.28]	[-399.76 : -21.17]	[-738.69 : -412.03]	11
AA09	[-174.15 : 158.64]	[15.94 : 489.10]	[-1.72 : 488.16]	[-23.20 : 403.80]	[-336.70 : 106.28]	[129.76 : 451.07]	[41.93 : 482.79]	[-753.14 : -364.43]	10
AA03	[-68.69 : 294.25]	[-721.00 : -326.77]	[-35.95 : 357.49]	[-283.06 : 148.65]	[-22.66 : 364.63]	[-60.78 : 371.92]	[-1.31 : 442.54]	[-421.30 : -78.98]	11
AA01	[-200.61 : 268.21]	[-477.14 : 110.66]	[-403.24 : 84.26]	[33.96 : 536.72]	[-747.46 : -81.79]	[-215.07 : 308.08]	[-194.66 : 355.84]	[-669.76 : -235.98]	10
6 (0.038)	[56.00 : 338.63]	[-477.53 : -220.69]	[-369.04 : -133.67]	[-308.45 : -50.62]	[-114.89 : 138.60]	[-121.65 : 145.67]	[-223.08 : 13.32]	[223.76 : 471.23]	11
AA02	[-89.65 : 203.36]	[-368.92 : -88.57]	[-238.37 : 20.72]	[-190.28 : 83.12]	[-124.20 : 158.01]	[-54.88 : 226.65]	[-120.52 : 164.49]	[-204.67 : 70.63]	10
AA		[-699.53 : -254.39]	[-40.30 : 367.35]	[-498.26 : -43.58]	[-131.14 : 295.17]	[-199.26 : 285.65]	[-402.40 : 48.69]	[45.53 : 421.96]	11
AA		[-547.58 : -164.15]	[-579.99 : -242.71]	[-572.77 : -141.23]	[-534.94 : 24.78]	[-601.51 : -101.45]	[-480.68 : -28.97]	[241.10 : 659.91]	10
AA		[291.44 : 545.26]	[-20.45 : 292.17]	[111.34 : 387.68]	[-146.63 : 137.06]	[-79.89 : 184.66]	[-67.39 : 207.00]	[-333.55 : -59.44]	11
AA12	[-245.86 : 26.45]	[332.49 : 551.89]	[-147.52 : 122.67]	[-173.75 : 76.02]	[113.03 : 328.04]	[-325.55 : -66.06]	[-36.92 : 221.45]	[160.27 : 414.11]	12
AA14	[-35.06 : 340.60]	[57.29 : 408.67]	[-484.39 : -82.91]	[-240.74 : 129.71]	[-266.50 : 130.98]	[-449.08 : -26.49]	[-327.11 : 82.21]	[384.95 : 700.45]	14
AA11	[-130.97 : 191.50]	[332.18 : 571.81]	[90.96 : 485.24]	[31.99 : 347.31]	[31.15 : 327.04]	[-67.36 : 230.84]	[113.42 : 436.93]	[-472.96 : -98.71]	12
AA13	[-178.26 : 231.77]	[42.30 : 426.86]	[217.42 : 633.53]	[-534.33 : -92.48]	[68.34 : 397.51]	[-271.75 : 170.47]	[-91.41 : 316.16]	[3.65 : 431.00]	12
AA15	[-491.78 : -70.01]	[-619.29 : -254.11]	[-351.24 : -29.45]	[-99.99 : 307.27]	[-611.26 : -169.30]	[-140.04 : 272.75]	[-542.37 : -146.89]	[-840.00 : -574.57]	10

Figure 31: *Metadata for an object*

The use of the  icon of the bar menu will display all the metadata of the Symbolic Data Table. They are organised within the following paragraphs:

- File general characteristics
- Survey description;
- Symbolic objects;
- Symbolic variables
- Original variables
- Original Files

For viewing the metadata of a SO/variable, the label of the SO/variable has to be selected (see Figure 32).

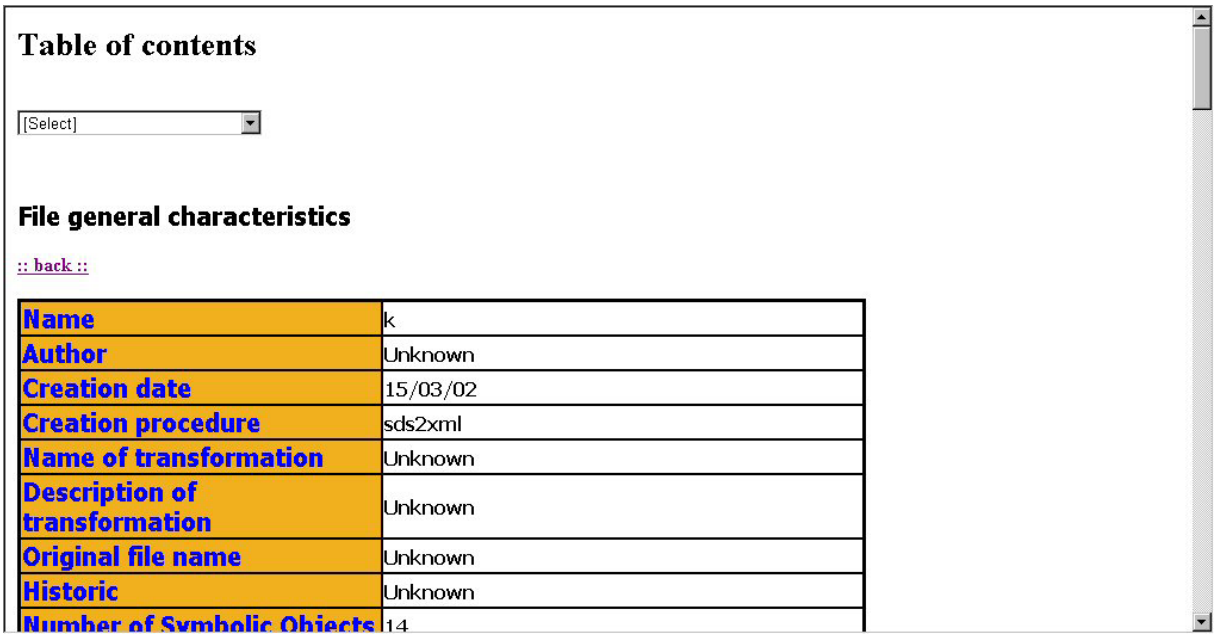


Figure 32: *Metadata "File general characteristics"*

TREATMENT METHODS

Descriptive Statistics and Visualisation

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

DSTAT Help Guide

Descriptive Statistics

Edited by DAUPHINE

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

5 DSTAT : Descriptive Statistics

5.1 Introduction

DSTAT stands for “Descriptive STATistics”.

The DSTAT method is actually a set of methods (hereafter called sub-methods), which aim to extend to the symbolic variables statistical methods usually applied to conventional variables. These sub-methods build a report listing, and some of them have an interactive graphics output.

	Variable type			Graph
Sub-method	Categorical multi-valued	Interval	Modal	
Frequencies for categorical multi-valued	X			X
Frequencies for interval		X		X
Biplot		X		X
Capacities			X	X
Numeric and symbolic characteristics		X		

Central object: just deals with the individuals (symbolic objects).

N.B.: there is no computing associated to Biplot, which uses DSTAT in a pass-through mode, taking advantage of its facilities for handling the interval variables in the workbench environment.

5.2 Input

The input of DSTAT is an ASSO symbolic objects file of type .SDS or .XML.

Three variable types may be processed by DSTAT, as detailed above.

For the Central Object submethod, the file must contain the distances matrix.

5.3 Parameters

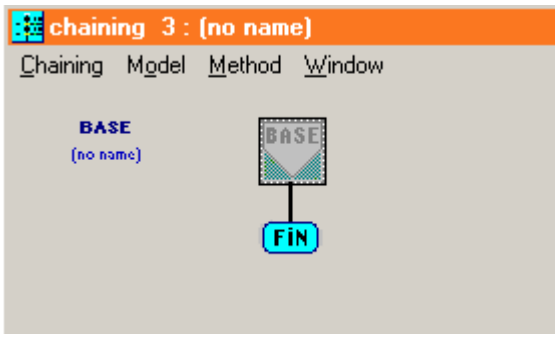
DSTAT is inserted in a workbench chaining, as a single method (regardless of which sub-method will be selected).

It may be alone in the chaining, or follow another method, the output of which is the input to DSTAT. No other method may follow it, since DSTAT has no ASSO data file output.

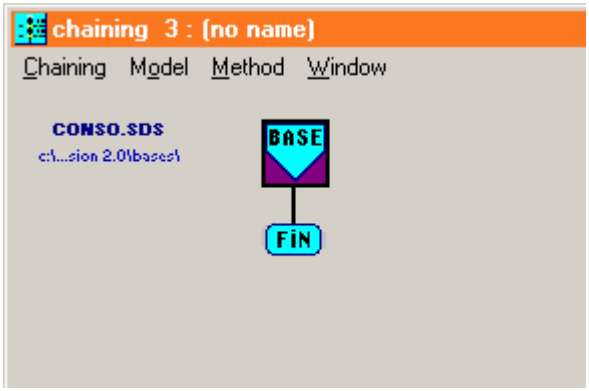
5.3.1 Create a chaining for DSTAT

The following description assumes that the user builds a chaining for DSTAT from the beginning. But if DSTAT follows another method in a chaining being built, steps 1 and 2 have to be skipped.

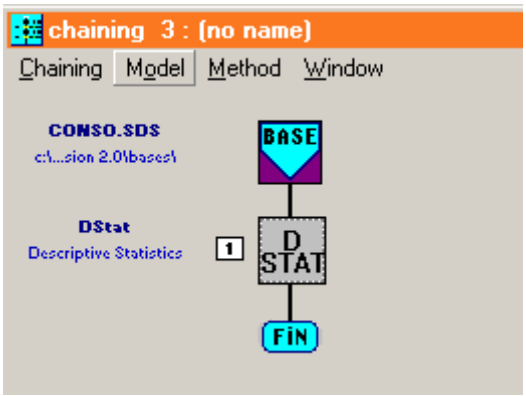
1. Select, in the main menu, "File" then "New chaining"



2. Double-click on the BASE icon and select the data base (either .SDS or .XML file type).

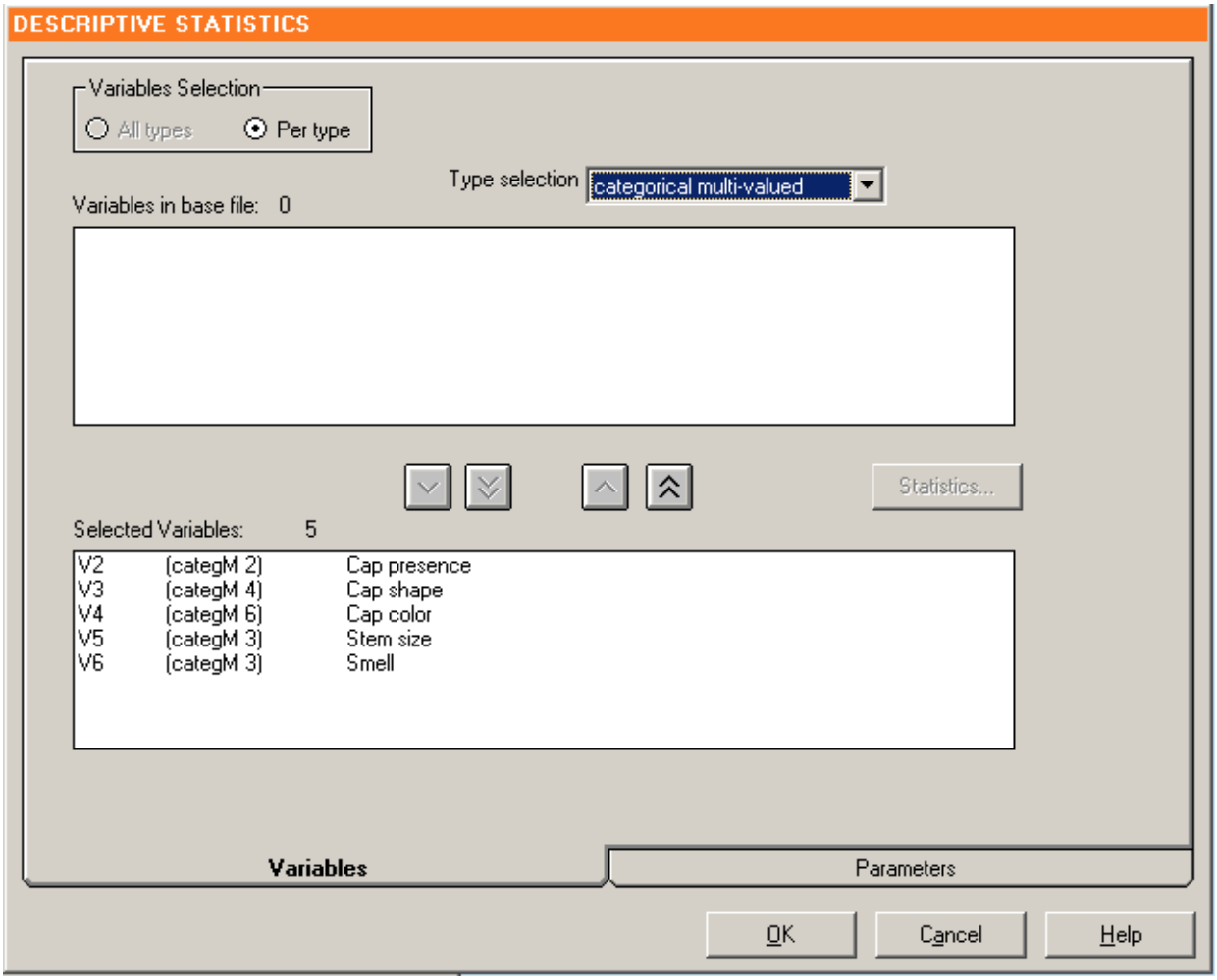


3. Select, in the chaining menu, "Method" then "Insert method"; an empty icon is added to the chaining.
   From the "Methods" window on the left, drag the DSTAT icon into the empty icon.
   The chaining is now ready for parametrization.



4. Double-click on the DSTAT icon to call the parametrization dialog.

5.3.2 Variables panel

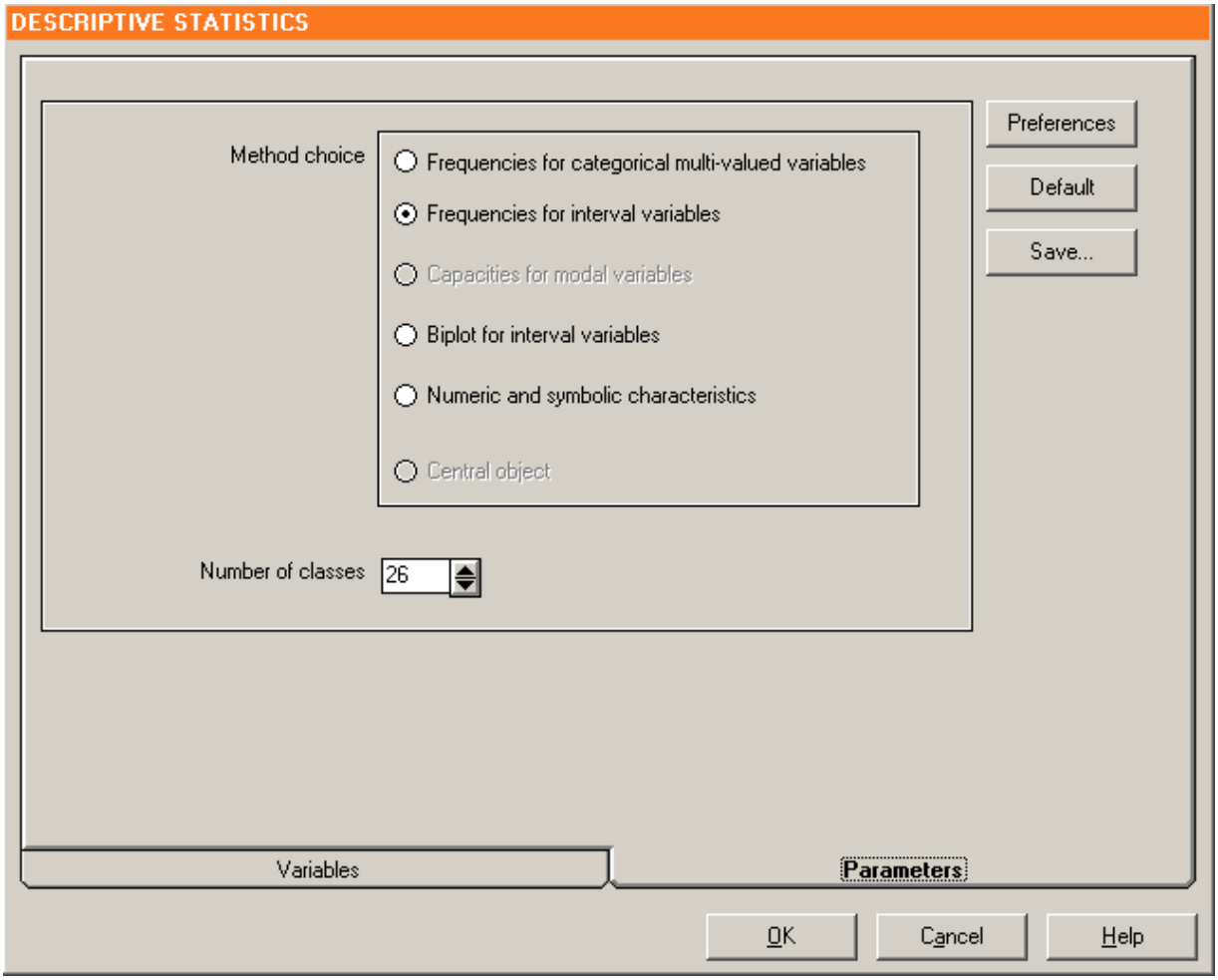


You have to select the variables type matching the sub-method you intend to select (or have already selected) in the Parameters panel.

You may select all or part of the variables inside this type.
It is advisable to select all of them, because DSTAT will later let you select those you want to effectively process, in an interactive way; i.e., you may change the selection at will while staying inside DSTAT (no need to exit and return to the workbench).
Selecting part of the variables here is useful only if you wish to restrict the set of variables that will later be offered by DSTAT for selection.

In case of "frequencies for categorical multi-valued variables", DSTAT will take all the applicable variables regardless of whether the user selected all of them or part of them; it is a must for this sub-method, because of the rules.

5.3.3 Parameters panel



You have to choose the optional sub-method to run.

The options are enabled according to the different variable types contained in the data file, not according to the variable type selected in the other panel, because it is possible to select the option before the variables type. It is up to the user to match both; in case of mismatch the execution does not start and an explanatory warning message shows up.

The applicability is as follows:

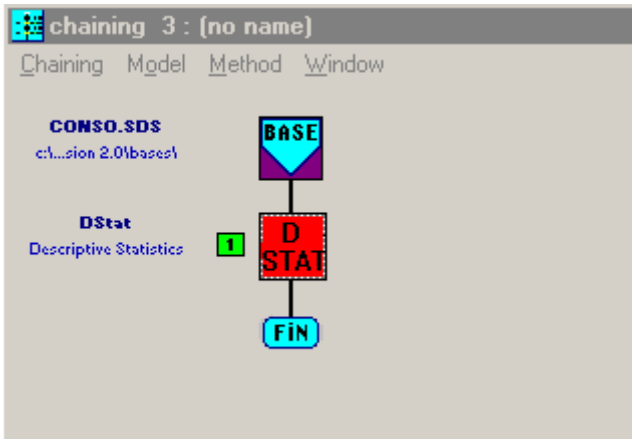
	Types of variables		
	Categorical multi-valued	Interval	Modal
Frequencies for categorical multi-valued	X		
Frequencies for interval		X	
Capacities			X
Biplot		X	
Numeric and symbolic characteristics		X	

Central object: just deals with the individuals (symbolic objects), so that it is not type

dependant ; but it requires that the data file contains the distances matrix (it is disabled if not).

The **Number of classes** parameter only applies to the "frequencies for interval variables"; it is disabled for the other options.
It tells DSTAT in how many "classes", i.e. sub-intervals, it should divide the overall interval for each variable.

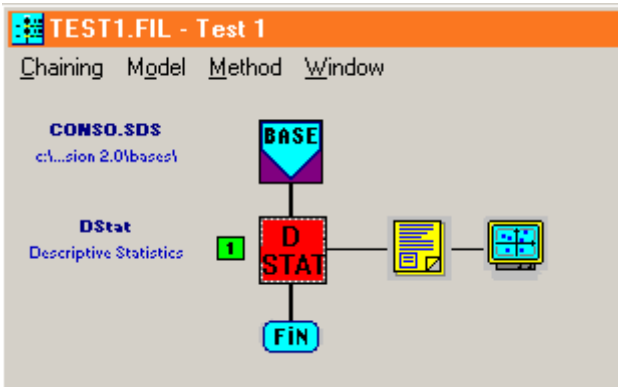
When you click on OK the DSTAT icon turns red; the chaining can now be executed.



It should be noted that DSTAT is always the only or last method in a chaining, because it does not create any data file that might be processed by another method.

5.3.4 Execute the chaining

Click on F5 to execute the chaining.
At the end, icons will be added to the right of the DSTAT icon.
The first one is for the listing of the results and the second one, if applicable, for the graph (see Output applicability below).
If an error occured during the execution, the graph icon will not show up, the listing icon will be grayed and the listing will contain the error description.



5.4 Output applicability

	Listing	Graph
Frequencies for categorical multi-valued	X	X
Frequencies for interval	X	X
Capacities	X	X
Biplot		X
Numeric and symbolic characteristics	X (*)	
Central object	X (*)	

(*) The results are also dynamically displayed in a dialog or message window at execution time.

5.5 Variables selection for the graphs

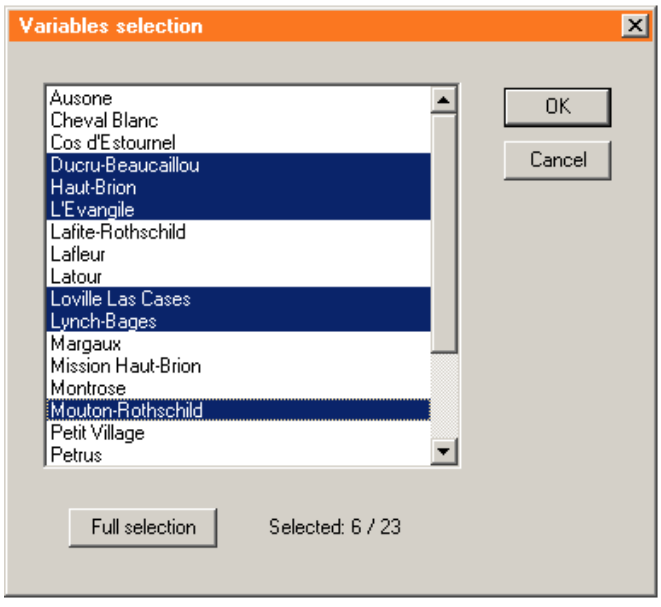
If several variables have been selected in the workbench, the module handling the graph will need to know which variable (or variables) it must process first, so that in such a case what shows up after clicking on the graph icon is a selection dialog for the variables.

The selectable variables are limited to the set defined in the workbench.

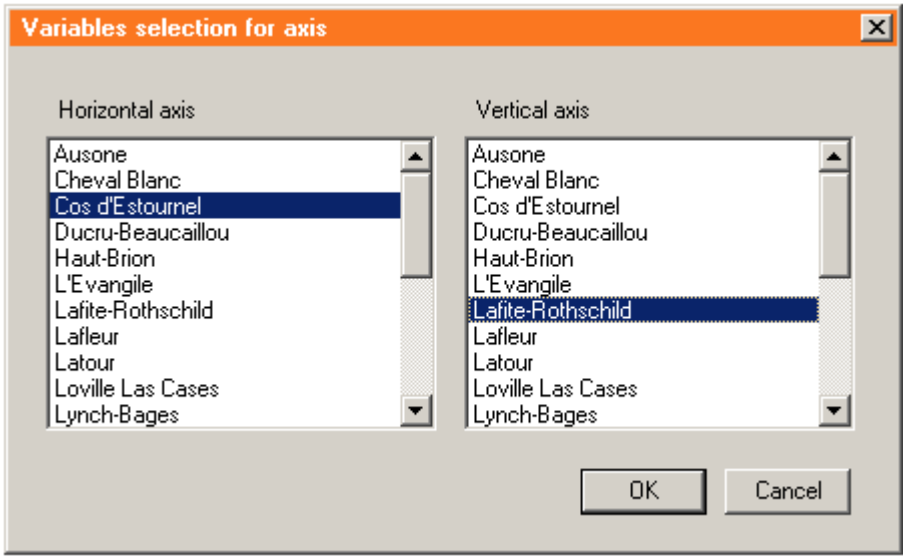
Apart from the biplot which needs two and only two variables -that it takes as X and Y axis-, the other sub-methods are basically related to a single variable; but it is possible to select more than one, what will result in several smaller graphs in parallel, in order to ease comparisons between variables.

Whatever the sub-method, it is possible to later change the variable(s) selection at will using the relevant graph menu command (or associated toolbar button).

Selection for all graphs but the Biplot :



Selection for the Biplot :



INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

VIEW Help Guide

Viewer

Edited by FUNDP

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

6 VIEW : Viewer

6.1 Introduction

The module VIEW has been developed in order to run the symbolic objects module VSTAR independently from a statistical treatment method on any ASSO base file.

6.2 The input to the VIEW module

The input of VIEW is an ASSO symbolic objects file of type .sds or .xml. All the types of the variables are processed.

6.3 The parameters of VIEW

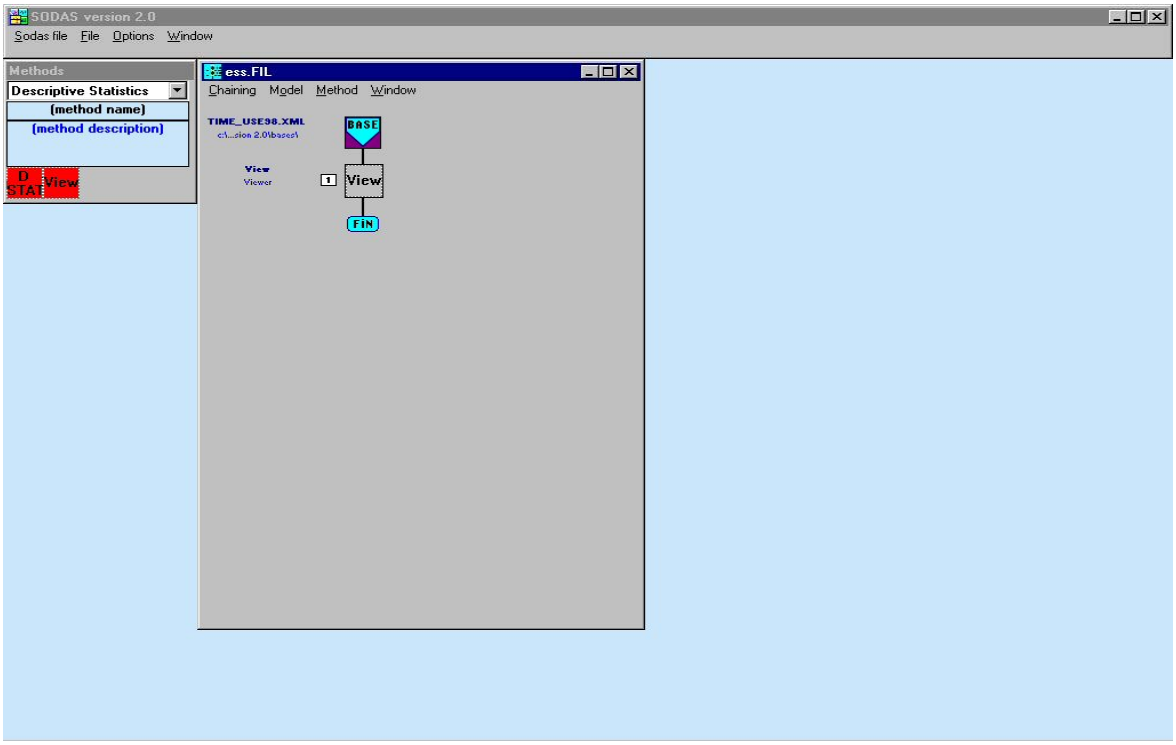


Figure 1: Inserting the module VIEW

To run VIEW module:

- insert the VIEW icon in the chain (figure 1);
- click on the icon and select the symbolic objects and the variables which are necessary for the analysis (figure 2).

When you will enter to VSTAR, you will be able to access to all the objects and the variables, but your selection will be given by default (in dark grey in the table).

The number of selected variables must be at least 3 and the number of selected objects is limited to 30, for memory reason.

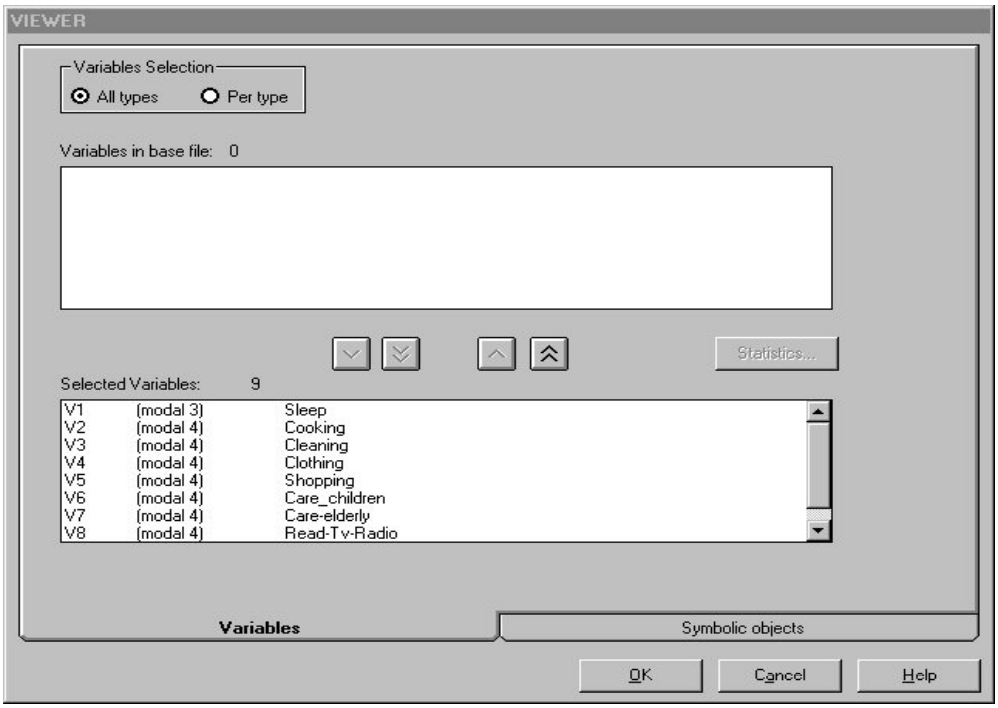


Figure 2: Variable selection in the workbench for VIEW

Save the chaining and run VIEW in clicking on the icon.

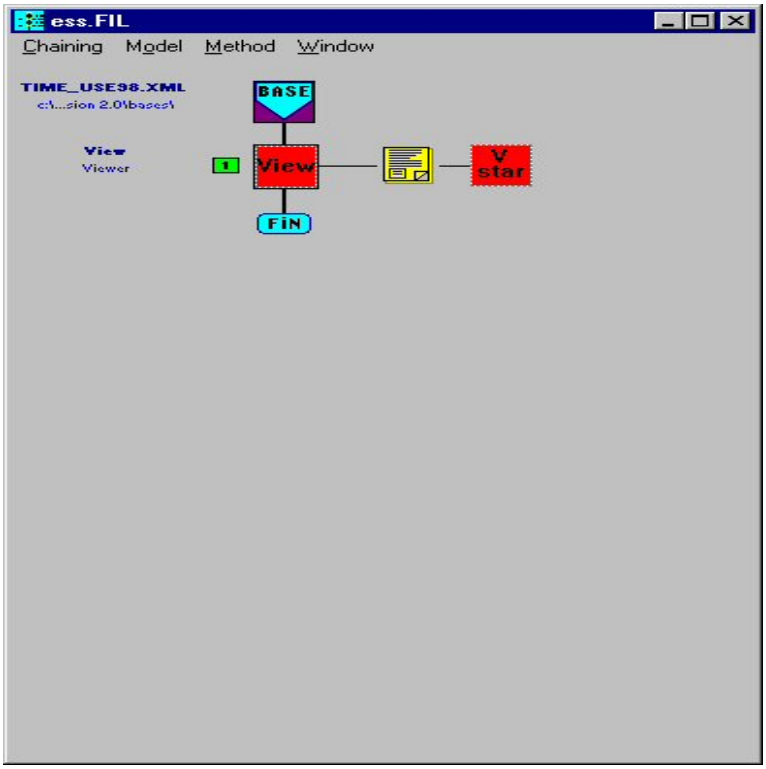


Figure 3: VIEW results

The result of VIEW is a report and an access to VSTAR (figure 3).

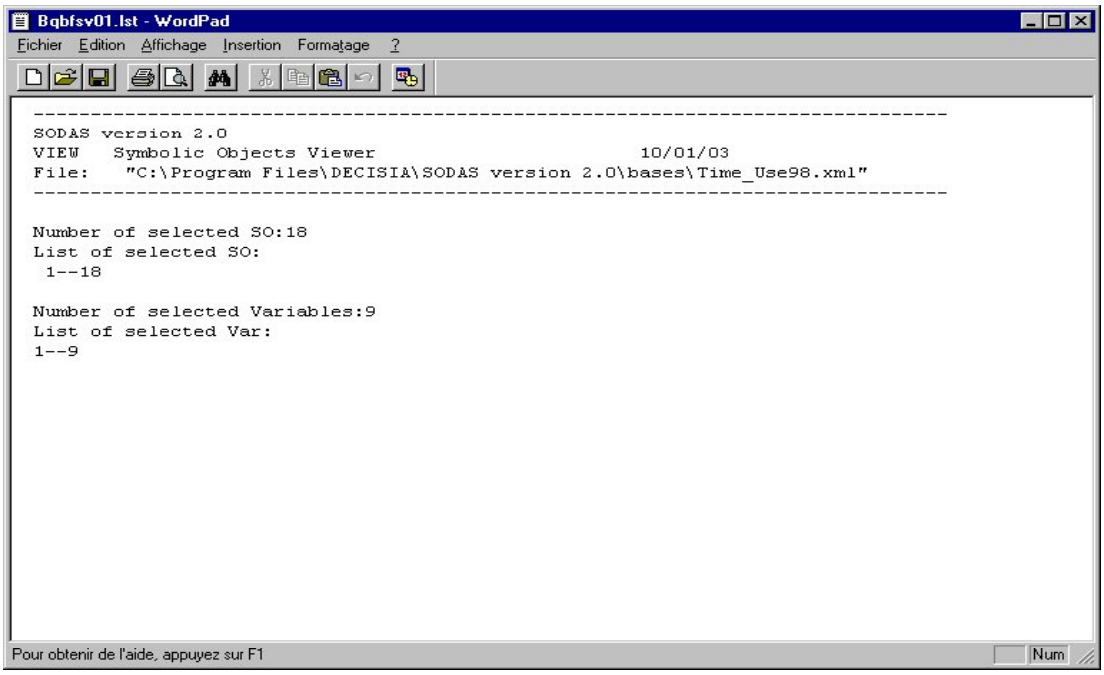


Figure 4: VIEW report

The report gives the number of selected SOs and the number of selected variables (figure 4). When clicking on VSTAR icon, you run the visualisation VSTAR module.

Dissimilarity and Matching

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

DISS Help Guide

Descriptive Measures

Edited by DIB

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **01/12/2003**

7 DISS : Descriptive Measures

7.1 Introduction

The module DISS has been developed for the method "Dissimilarity and Matching". This method is finalized to compare SOs in order to quantify the existing correlations, to cluster or to discriminate between them. The results of this sort of examination, aiming to explicitly understand, measure and identify groups of SOs, can be called "Symbolic Data Patterns" and may be applied to other statistical methods or some Data Mining tasks.

Particularly DISS module aims to compare SOs in order to evaluate their dissimilarity and uses directly a visualization module, named VDISS, to represent graphically these results.

7.2 The input to the DISS module

The input to DISS module is a set of SOs in a selected ASSO file. It is possible to evaluate the dissimilarity among them either described by the complete set of symbolic variables or by a restricted selected set. On the contrary, no selection of SOs from the SOs table can be performed: since all pairs of SOs are compared for dissimilarity evaluation. Metadata are only read by the modules, but they are not managed.

The SOs may be described by the following symbolic variables:

set-valued variables

categorical single (i.e. *town(w)=London* where '*town*' is the variable and *w* the individual)

categorical multi-value (i.e. *town(w)={London, Paris, Rome}*)

quantitative single valued (i.e. *height(w)=3.5*)

interval (i.e. *height(w)=[3,7]*)

modal variables

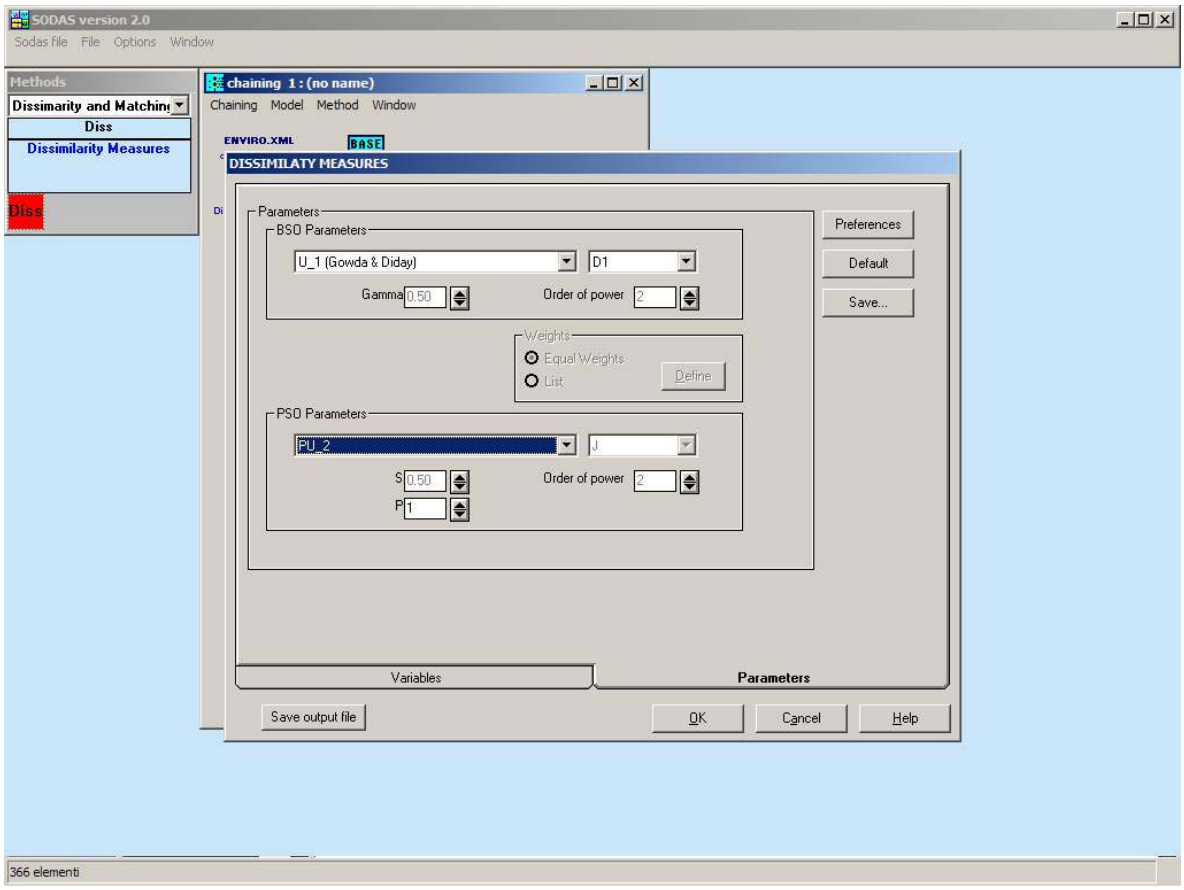
probability distribution (i.e. town(w)={London(0.2), Paris(0.7), Rome(0.1)})

Constraints (hierarchical dependencies and logical dependencies) are allowed for some dissimilarity measures. Missing values (i.e., unknown values) are also allowed.

According the mixture of these variables, the dissimilarity measures can be defined for BSOs, PSOs, mixed BSOs and PSOs. The last kind of SOs are described by both set-valued and modal variables (i.e. w=[age(w)=[20,50] ∧ sex(w)={F(0.4), M(0.5), I(0.1)}]).

7.3 The parameters of DISS

After selecting the relevant symbolic variables for the computation of dissimilarities among SOs, the parameter window of DISS module looks as follows:



The several options and parameters are explained below in the same order as they appear in the window. For a precise explanation, please look at the tutorial of this module ("DISS tutorial").

This window is divided in three main sections: one for BSOs, one for the weights of each selected symbolic variables (in the middle), and one for PSOs. Each section, excepted that one in the middle, are activated or not-activated according the type of selected SOs (i.e. in the case of mixed SOs they must be both activated).

BSO Parameters

In this section is possible to select the dissimilarity measure which will treat all those set-valued variables without a probability distribution. The possible dissimilarity measures for BSOs with the linked parameters are shown in the following table:

Dissimilarity measure	***Parameters***	***Constraints***	***Default***
U_1 (Gowda & Diday)	none		
U_2 (Ichino & Yaguchi)	Gamma Order of power	[0 .. 0.5] 1 .. 10	0.5 2
U_3 (Normalized Ichino & Yaguchi)	Gamma Order of power	[0 .. 0.5] 1 .. 10	0.5 2
U_4 (Weighted Normalized Ichino & Yaguchi)	Gamma Order of power List of weights per variable	[0 .. 0.5] 1 .. 10 Sum(weights) = 1.0	0.5 2 Equal weights
C_1 (Normalized De Carvalho)	Comparison function Order of power	D_1, D_2, D_3, D_4, D_5 1 .. 10	D_1 2
SO_1 (De Carvalho)	Comparison function Order of power List of weights per variable	D_1, D_2, D_3, D_4, D_5 1 .. 10 Sum(weights) = 1.0	D_1 2 Equal weights
SO_2 (De Carvalho)	Gamma Order of power	[0 .. 0.5] 1 .. 10	0.5 2
SO_3 (De Carvalho)	Gamma	[0 .. 0.5]	0.5
SO_4 (Normalized De Carvalho)	Gamma	[0 .. 0.5]	0.5
SO_5 (Normalized De Carvalho)	Gamma	[0 .. 0.5]	0.5
SO_6 ()	None		

PSO Parameters

In this section is possible to select the dissimilarity measure which will treat all those set-valued variables with a probability distribution. The possible dissimilarity measures for PSOs with the linked parameters are shown in the following table:

Dissimilarity measure	***Parameters***	***Constraints***	***Default***
PU_1	Compfun	J,CHI2,CHER,REN,LP	J
	Order of power	1..10	2
	S	(0,1)	0.5
	P	1..10	1
	List of weights per variable	Sum(weights) = 1.0	Equal weights

In particular :

PU_1 (J)	Order of power	1..10	2
	List of weights per variable	Sum(weights) = 1.0	Equal weights
PU_1(CHI2)	Order of power	1..10	2
	List of weights per variable	Sum(weights) = 1.0	Equal weights
PU_1(REN)	S	(0,1)	0.5
	Order of power	1..10	2
	List of weights per variable	Sum(weights) = 1.0	Equal weights
PU_1(CHER)	S	(0,1)	0.5
	Order of power	1..10	2
	List of weights per variable	Sum(weights) = 1.0	Equal weights
PU_1(LP)	P	1..10	1

	Order of power List of weights per variable	1..10 Sum(weights) = 1.0	2 Equal weights
PU_2	P	1..10	2
PU_3	none		

Particularly, the dissimilarity measure “PU_1” combined with the comparison functions “J-CHI2-REN-CHER”, may be used only when the SOs have been generated with KT-estimate (for addition information on this topic, see “DB2SO help guide”).

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

MATCH Help Guide

Matching Operators

Edited by DIB

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **24/11/2003**

8 MATCH : Matching Operators

8.1 Introduction

The module MATCH allows the computation of the matching between a set of symbolic objects (SOs) in order to discover their likeness or differences. Similarity judgments in the matching process are directional. In its simplest form, matching is the process of comparing two structures just for equality. The test fails if the structures differ in at least one aspect. In more complex case, the matching compares the description of a class with the description of an individual in order to establish whether the individual can be considered an instance of the class. In MATCH two forms of matching are implemented: the *canonical* matching and the *flexible* matching. The former checks for an exact match whereas the latter one computes a degree of matching. Matching two structures is crucial in symbolic classification, pattern recognition or expert systems.

The library of matching functions available in MATCH is also used by SO2DB in order to retrieve individuals in a relational database corresponding to some characteristics described by a symbolic object.

8.2 The input to the MATCH module

MATCH allows the computation of the matching between the set of all symbolic objects stored in an input ASSO file. Furthermore, MATCH automatically read the metadata stored in the metadata file named *meta<name of the input asso file>.xml* that is stored in the same path of the input ASSO file. In the case that the metadata file is not available in the required path, MATCH shows a message and it continues the running.

8.3 The output of MATCH module

The output is a square matrix stored in an output ASSO file. In particular, the result of the computation of the matching function between the i-th SO and the j-th SO in the input ASSO file is written in the (i, j)-entry of the matrix. Missing values are handled by simply substituting the whole set of possible values to each occurrence of a missing value.

MATCH module also updates the metadata file associated to the input ASSO file by adding information about matching measures for BSO and/or PSO used in the computation of the matching square matrix and the sub-set of selected symbolic variables.

8.4 The parameters of MATCH

The computation of the matching is performed for all couples of symbolic objects stored in input ASSO file but the users could decide to select only a sub-set of the symbolic variables describing the symbolic objects to be matched. A symbolic object is described by a collection of set-valued variables and/or modal variables, where a set-valued variable could be:

- a categorical single-value variable(i.e. *town(w)=London* where '*town*' is a variable describing the individual *w*),
- a categorical multi-value variable(i.e. *town(w)={London, Paris, Rome}*),

- a quantitative single-value variable(i.e. *height(w)=3.5*),
- an interval variable(i.e. *height(w)=[3,7]*);

while a modal variable describes a probability distribution (i.e. *town(w)={London(0.2), Paris(0.7), Rome(0.1)}*).

The matching measures are defined for both BSOs (described by only set-value variables) and PSOs (described by only modal variables). In the case of mixed SOs described by both set-valued and modal variables (i.e. w=[*age(w)*=[20,50] ∧ *sex(w)*={*F*(0.4), *M*(0.5), *I*(0.1)}]) the users have to choose a matching measure for both BSOs and PSOs.

The users can choose between two forms of matching between BSOs:

- canonical matching (CM) that checks for an exact match,
- flexible matching (FM) that computes a degree of matching.

Only a probabilistic flexible matching (PFM) can be selected for PSOs. In this case the users may also decide to use the *I-divergence, χ2-divergence* or *Hellinger coefficient* to compute the probabilistic flexible matching.

8.5 Method

The module MATCH provides matching measures for both boolean symbolic objects and probabilistic symbolic objects.

Matching measures for boolean symbolic objects

In the case of matching between BSOs, two matching functions are provided in the MATCH module: one for canonical matching (CM) and the other for flexible matching (FM) (see Table 1 for their parameters).

Table 1: Matching functions for BSO's

Dissimilarity measure	***Parameters***	***Constraints***	***Default***
CM	none	none	none
FM	none	none	none

Matching measures for probabilistic symbolic objects

The matching operator for PSOs, implemented in MATCH, is an extension of the flexible matching proposed for BSOs. Details about the parameters are reported in table 2.

Table 2: Matching functions for PSO's

Dissimilarity measure	*Parameters*	*Constraints*	*Default*
PFM	none	none	none
PFM	I	none	none
PFM	CHI2	none	none
PFM	HELL S	none (0,1)	none 0.5

The users may decide to use the *I*-divergence (I) or the χ^2-divergence (CHI2) in computing the matching between PSOs. Both of them are dissimilarity coefficients which are adequately transformed into similarity coefficients. Alternatively the users may decide to involve the *Hellinger* coefficient (that is a similarity-like coefficient) in the computation of flexible matching.

Clustering

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

DIV Help guide

Divisive Classification

Y. Lechevallier and M. Chavent
INRIA

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **20/05/2003**

9 DIV : Divisive Classification

9.1 Introduction

The module DIV is a divisive clustering method. This method is a hierarchical clustering method which starts with all objects in one cluster and proceeds by successive divisions of each cluster. At each step, a cluster is divided into two clusters according to a binary question. This binary question induces the best partition in two clusters according to an extension of the inertia criterion.

The algorithm stops after K-1 divisions, where K is the number of clusters given as input by the user.

9.2 The input to the DIV module

The input of the DIV method is a **classical data matrix** or a **symbolic data matrix**.

We take as an example the symbolic transformation of the waveform recognition data proposed L. Breiman, J.H. Friedman, R.A. Oslhen and C. J. Stone in "Classification and Regression Trees"; Belmont Eds, 1984.

The symbolic data matrix is here composed of 30 symbolic objects described by 21 symbolic interval variables. The “BASE” (see FIG1.) is the file wave30.sds.

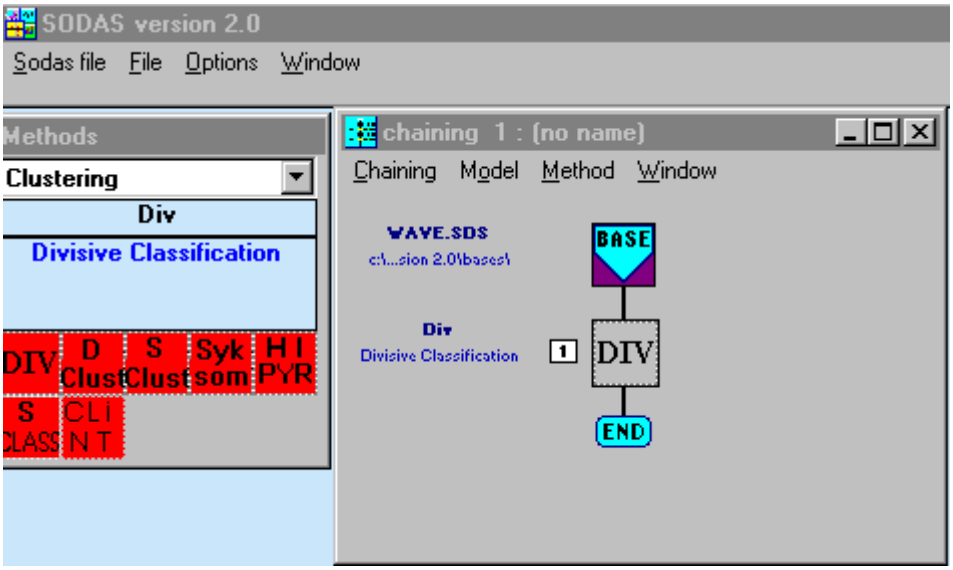


FIG1. Chaining example

9.3 Parameters of DIV

The DIV method takes also as input a list of variables and some parameters we will define now.

9.3.1 List of the selected variables

The user must choose the variables which will be used to compute the dissimilarity matrix and then the extension of the inertia criterion and to define the set of binary questions used to define the split. When you choose a list of variable take care that :

- **the definition domain of the selected variable has to be ordered**. If not, the results you will obtain will be totally wrong.
- it is not possible to mix variables having a continuous definition domain with variables having a discrete definition domain.

In this, in the SODAS DIV's parameters definition window (see FIG 2.), the user has to choose between qualitative and continuous variables and **should not mix them**. In the waveform example, the 21 continuous variables position1 to position21 are selected.

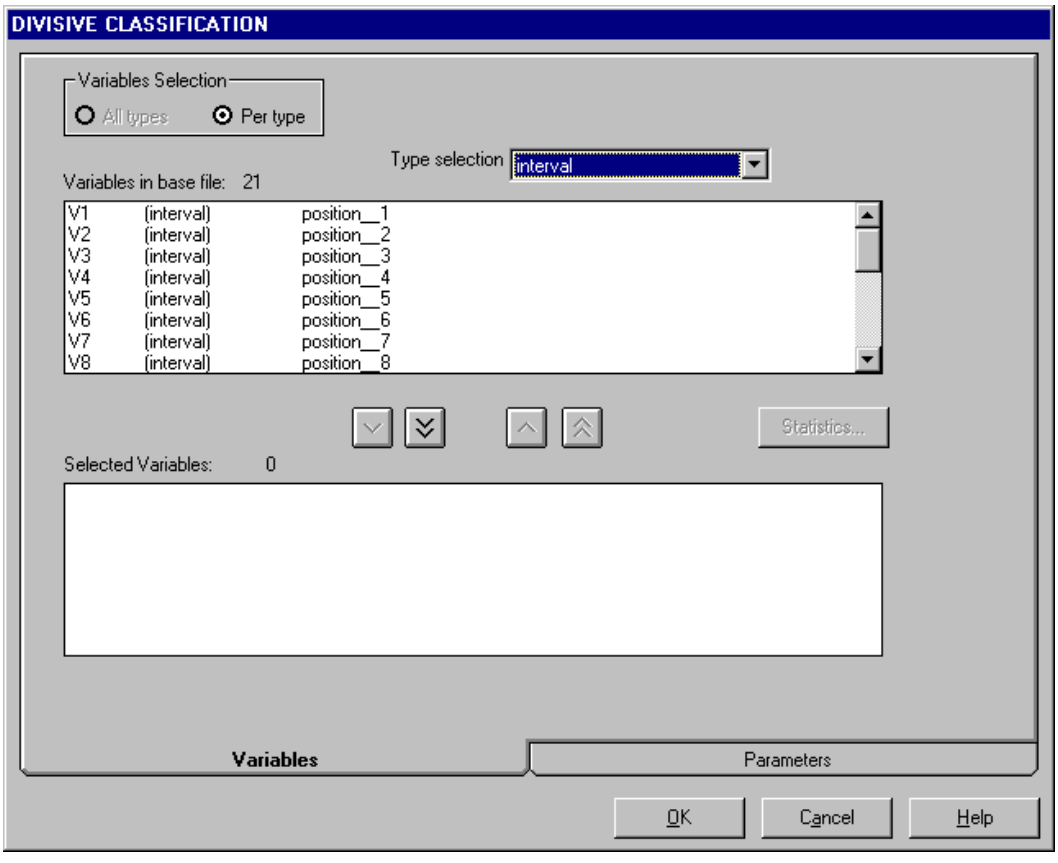


FIG2. The SODAS window for the selection of the variables

9.3.2 The parameter definition

Three parameters have to be defined :

- the dissimilarity between two objects can be normalized or not. It can be normalized by the inverse of dispersion or by the inverse of maximum deviation. The **dispersion** of a variable is here an **extension** to symbolic variables of the **variance**. The **maximum deviation** of a variable is here an **extension** to symbolic variable of the **span** (the biggest minus the smallest observed value).
- The number K of classes of the last partition. The division will stop after K-1 iterations and the DIV method will compute partitions from 2 to K classes.
- It is possible to create a partition file e.g. a text file which contain a matrix (a_{ij}) where each line i (i=1,…,n) corresponds to an object and each row j (j=2,…,K-1) corresponds to a partition in j classes, and where (a_{ij}) = k (k ∈ {1,…,j} means that the object j belongs to the class k in the partition in j classes.

In the waveform example (see FIG3.) we have chosen a dissimilarity with no normalization, the number of classes is 3 which mean that the DIV method will perform a partition in two classes and a partition in 3 classes, and we have chosen not to create a partition file.

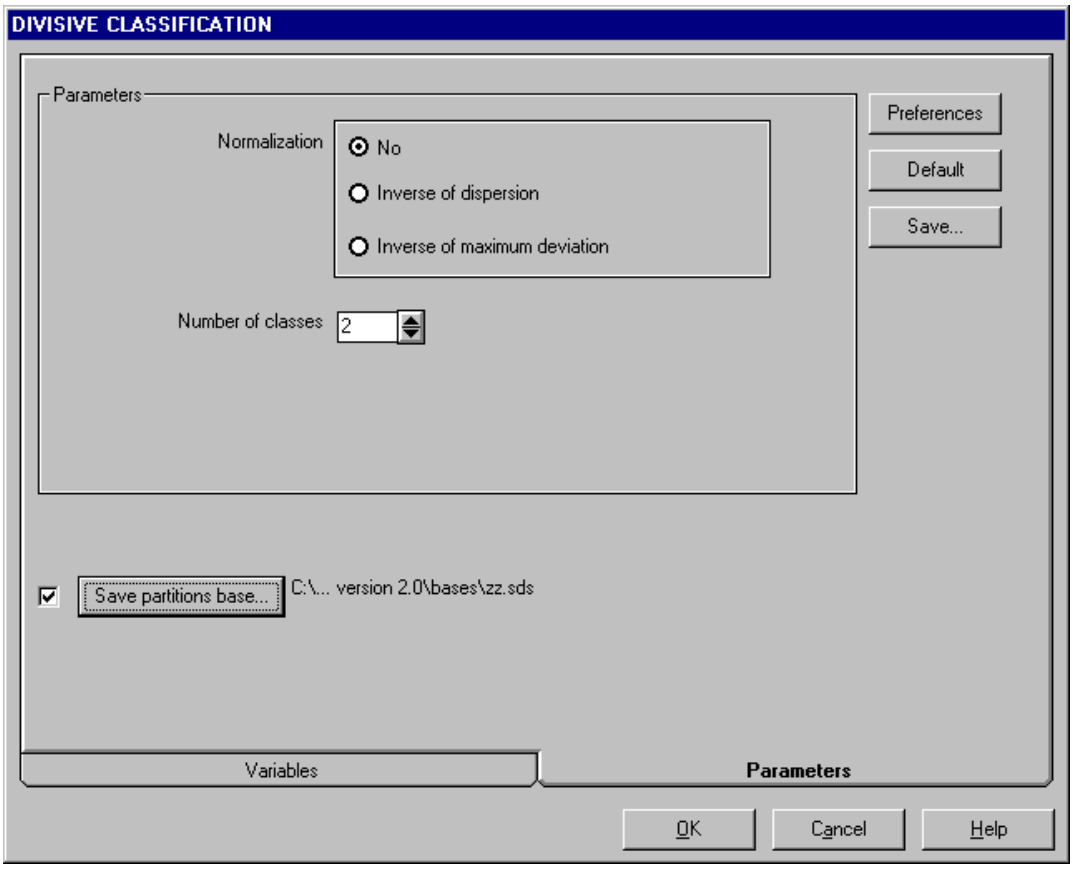


FIG3. The SODAS window for the definition of the parameters

9.4 Additional buttons

9.4.1 Preferences

The user is able to save his own default values for the parameters (see also “Save”!). He gets these parameters by pushing this button.

9.4.2 Default

If this button is pushed, all parameters will be set to their default values.

9.4.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

9.4.4 Save Partition base …

With this option, the partition of n data intervals into m clusters (produced after running the DIV module) is stored in a SODAS file. The user is able to change the name of that file.

9.4.5 Save Node base …

With this option, the symbolic representation of all m clusters produced after running the DIV module are stored in a SODAS file. The user is able to change the name of that file.

9.4.6 The output

After having defined the parameters and run the DIV method, a listing is given as output (see FIG4.)

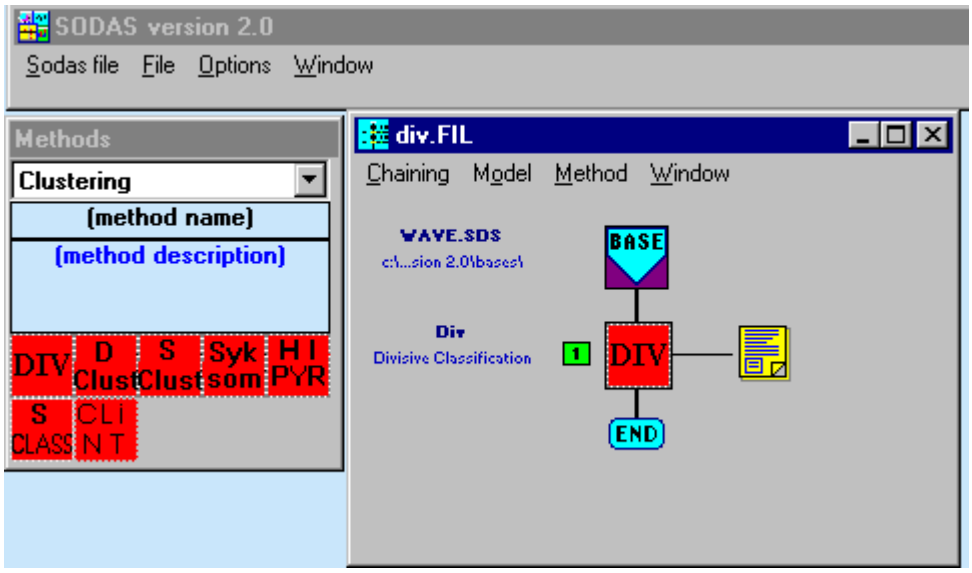


FIG4. The chaining after running the DIV method

The listing contains :

- A list of the "variance" of the selected variables, if those variables are continuous.
- For each partition from 2 to K classes, a list of the objects contained in each class and the "explicated inertia" of the partition.
- The clustering tree

For instance, the end of the listing obtained with the waveform example is the following :

```
PARTITION IN 3 CLUSTERS :
-------------------------:


Cluster 1 (n=14) :

"wave_3_9"   "wave_3_4"   "wave_1_3"   "wave_3_2"   "wave_3_3"
"wave_1_1" "wave_1_5" "wave_3_7" "wave_3_8" "wave_1_4"
"wave_3_0" "wave_1_2" "wave_3_6" "wave_3_5"

Cluster 2 (n=10) :

"wave_1_9"   "wave_2_0"   "wave_1_8"   "wave_1_0"   "wave_1_7"
"wave_2_9" "wave_2_6" "wave_1_6" "wave_2_7" "wave_2_8"


Cluster 3 (n=6) :

"wave_2_5"   "wave_2_1"   "wave_2_3"   "wave_2_2"   "wave_3_1"
"wave_2_4"

Explicated inertia : 62.433255


THE CLUSTERING TREE :
---------------------

   - the number noted at each node indicates
     the order of the divisions
   - Ng <-> yes and Nd <-> no


      +----  Classe 1 (Ng=14)
      !
 !----1- [position  8 <= 2.640000]
```

```
!
!    +----  Classe 2 (Ng=10)
!    !
!----2- [position 11 <= 4.167500]
     !
     +----  Classe 3 (Nd=6)
```

You can read the previous tree as follows :

- The first division has been performed according to the variable position8 and to the cut value 2.64. The first binary question is [position8 ≤ 2.64] ? The objects in classe1 (Ng=14) have answered “yes” to this question. They correspond to the 14 objects of the cluster 1 of the partition in two clusters. The 16 other objects (which will be next divided) have answered “no” to this question and correspond to the objects of the cluster 2 of the partition in two clusters. The 14 objects of cluster 1 and the 16 objects of cluster 2 are listed in the beginning of the listing.
- Similarly, the second division has been performed according to the binary question [position11 ≤ 4.16] ? and this question has induced the two classes (class2 and 3). They correspond to the cluster 2 and the cluster 3 of the partition in 3 clusters.

9.4.7 More details on the waveform example

The example of the waveform symbolic data are presented here with more details. The number of classes given as input is 4. The cluster1 of the partition in 3 clusters is divided in two classes and the partition in 4 clusters is the following :

Cluster 1 (n=6) :Rule [Pos08 > 2.64] AND [po17 <= 2.50]
"wave_3_4" "wave_3_2" "wave_3_3" "wave_1_5" "wave_3_6" "wave_3_5"

Cluster 2 (n=10) :Rule [Pos08 <= 2.64] AND [po11 >4.16]
"wave_1_9" "wave_2_0" "wave_1_8" "wave_1_0" "wave_1_7" "wave_2_9" "wave_2_6" "wave_1_6" "wave_2_7" "wave_2_8"

Cluster 3 (n=6) : Rule [Pos08 <= 2.64] AND [po11 <= 4.16]
"wave_2_5" "wave_2_1" "wave_2_3" "wave_2_2" "wave_3_1" "wave_2_4"

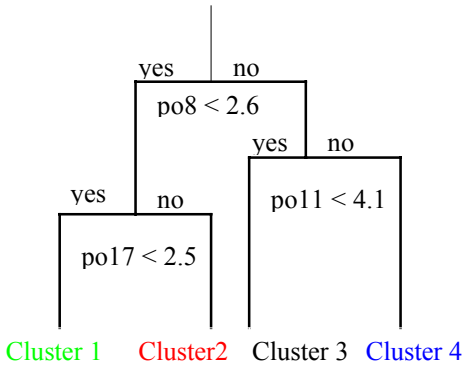
Cluster 4 (n=8) :Rule [Pos08 > 2.64] AND [po17 > 2.50]
"wave_3_9" "wave_1_3" "wave_1_1" "wave_3_7" "wave_3_8" "wave_1_4" "wave_3_0" "wave_1_2"

The explicated inertia is : 71.27

The divisive tree (listing) is the following :

```
          +----  Class 1 (Ng=6)
          !
     !----3- [position 17 <= 2.507500]
     !    !
     !    +----  Class 4 (Nd=8)
     !
!----1- [position  8 <= 2.640000]
     !
     !    +----  Class 2 (Ng=10)
     !    !
     !----2- [position 11 <= 4.167500]
          !
          +----  Class 3 (Nd=6)
```

This tree corresponds to the following dendogramm.



Cluster 1 = [po8 < 2.6] ∧ [po17 < 2.5]
Cluster 2 = [po8 < 2.6] ∧ [po17 > 2.5]
Cluster 3 = [po8 > 2.6] ∧ [po11 < 4.1]
Cluster 4 = [po8 > 2.6] ∧ [po11 > 4.1]

FIG5.

9.4.8 Classical interpretation

According to the four rules corresponding to the four clusters (see FIG5.) , we built a 4-clusters partition of the 3000 classical individuals. This partition is visualized FIG6. on the first plane of the Principal Component Analysis performed on the 30000 classical individuals and the 21 variables.

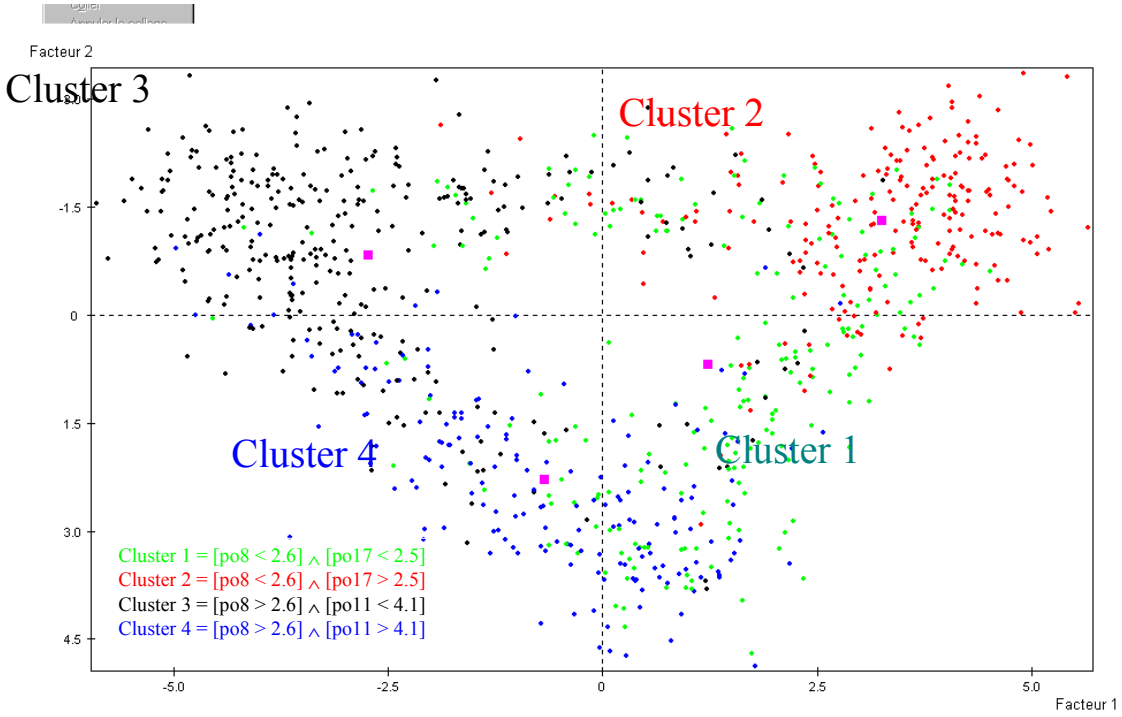


FIG6. Fist plane of the PCA and the 4 clusters

The 21 variables are visualized FIG7.

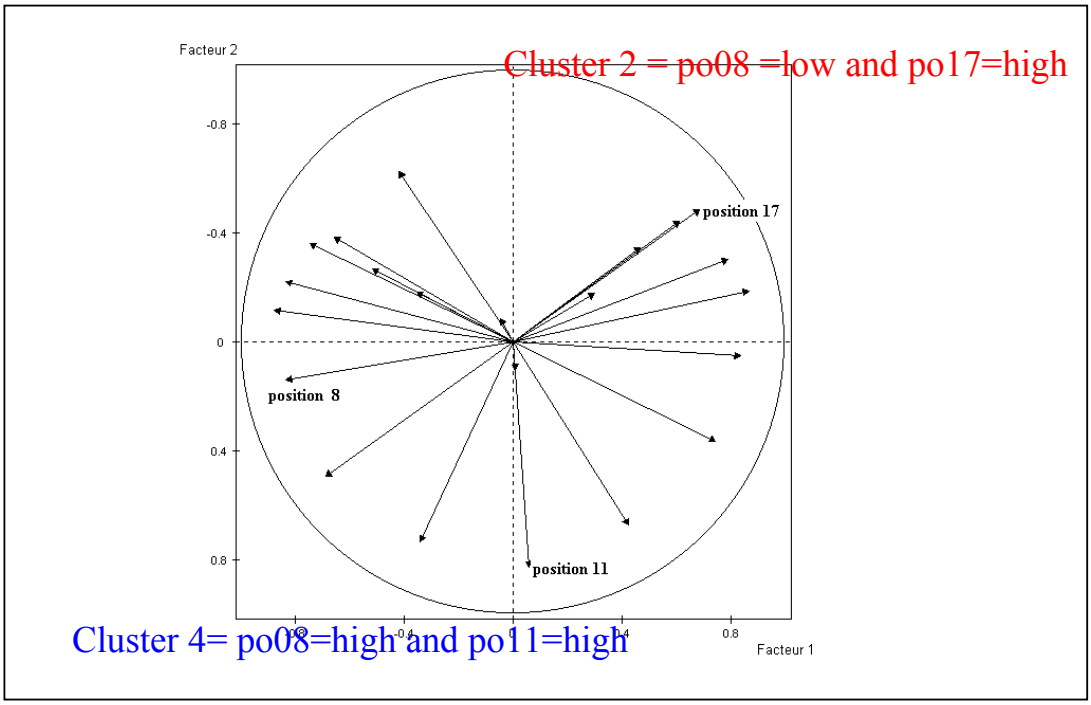


FIG7. First plane of the PCA and the splitting variables

9.4.9 Symbolic interpretation

According to the list of the symbolic objects in each cluster, we built for each cluster a new symbolic object. There is 30 objects which have been clustered (see FIG.8). Each object corresponds to a set of classical individuals. Indeed, each cluster of symbolic objects corresponds to a cluster of classical individuals. For each cluster of individuals, we built with **DB2SO** a new symbolic object which describes the variation on the 21 variables of the individuals of the cluster.

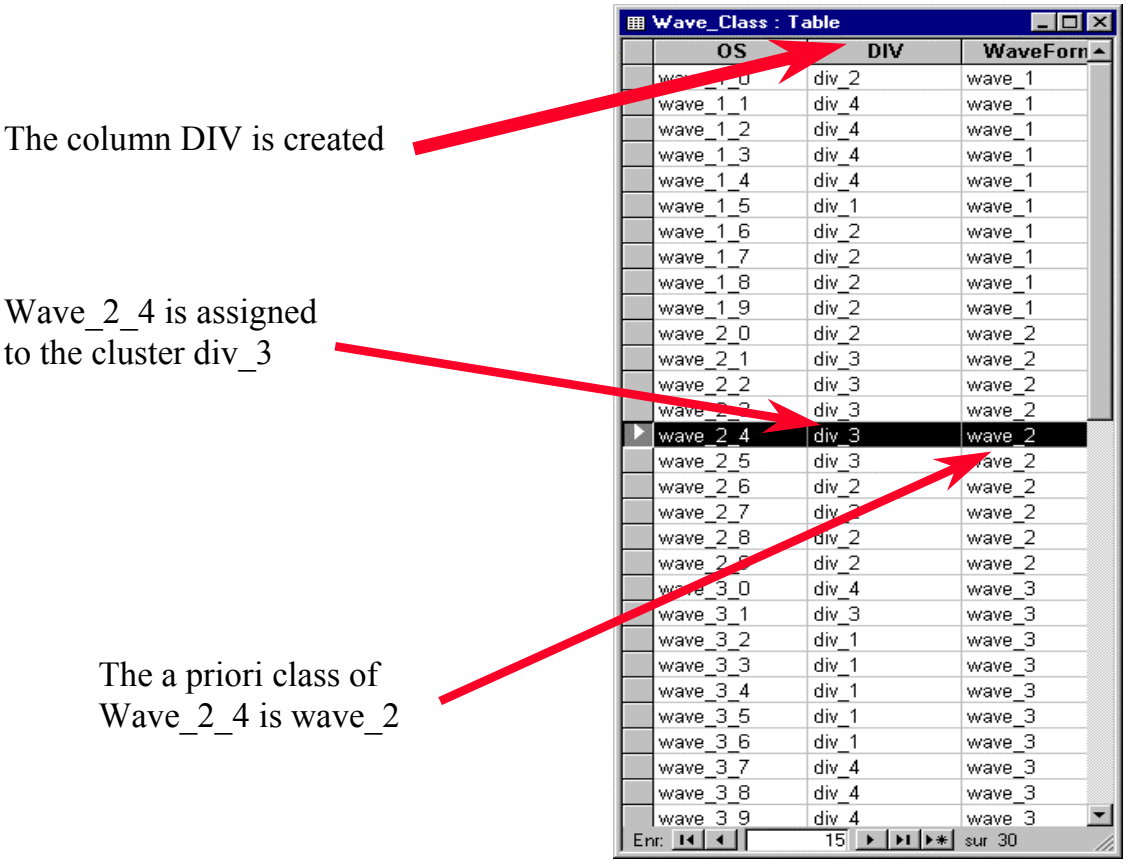


FIG8. The result of the DIV clustering in ACCESS

Then, each cluster can be visualized by using the SOE method. On the FIG9., we can see four stars build with SOE. The symbolic objects described are the four objects constructed below with DB2SO and the 3 variables chosen are the variables used in the clustering tree.

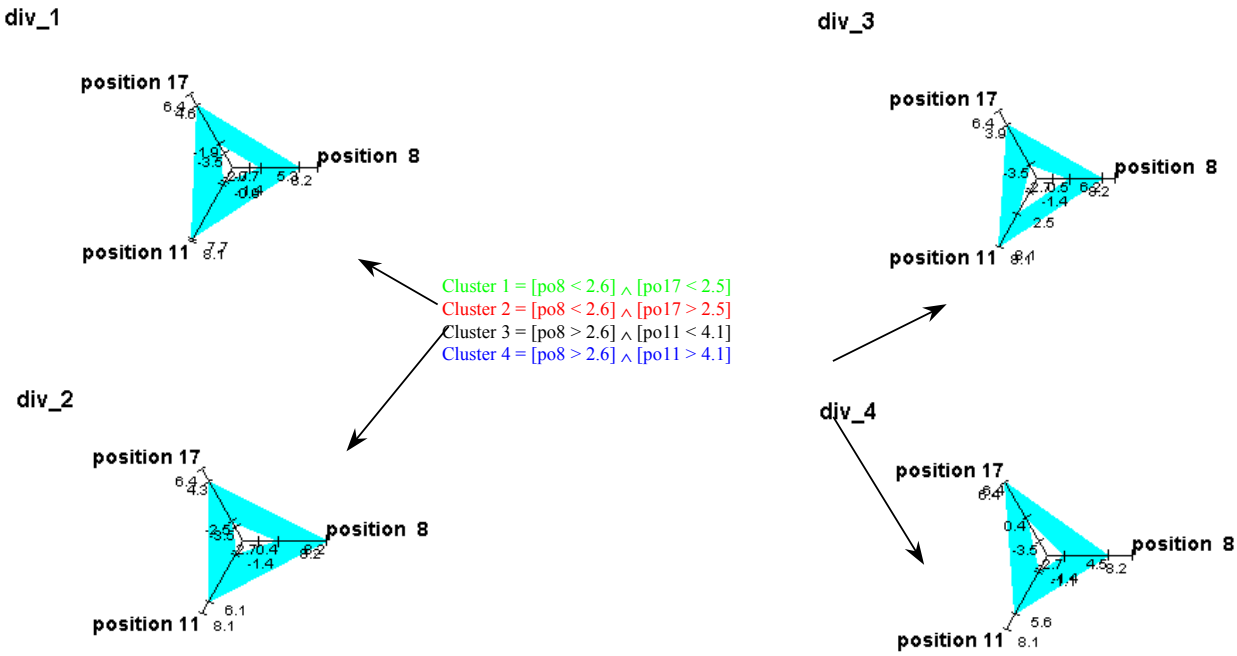


FIG9. Stars of the symbolic interpretation of each cluster

Bibliography

Chavent M. (1997*), Analyse des données symboliques, une méthode divisive de classification*, Thèse de l'Université Paris IX-Dauphine.

Chavent M. (1998), *A monothetic clustering method*, Pattern Recognition Letters 19, pp. 989-996.

Chavent M. (2000), *Criterion-Based Divisive Clustering for Symbolic Objects*, In Analysis of Symbolic Data, Exploratory methods for extracting statistical information from complex data. Bock, H. H., Diday, E. (eds.) Studies in Classification, Data Analysis and Knowledge Organisation, Springer-Verlag.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

HIPYR-VPYR Help guide

Hierarchical and Pyramidal Clustering

Edited by FEP

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

10 HIPYR-VPYR : Hierarchical and Pyramidal Clustering

10.1 Introduction

The module HIPYR is used to perform a hierarchical or pyramidal clustering on a set of symbolic objects. The clustering may be formed on the basis of a dissimilarity matrix (numerical clustering) or using the symbolic data array and imposing that clusters are concepts (symbolic clustering). The module produces as output a listing, with the obtained results and a graphical interactive representation of the obtained structure (hierarchy or pyramid). VPYR is the visualisation module; it is automatically linked to HIPYR.

10.2 The input to the HIPYR module

The input for the module HIPYR is a symbolic data array, where individuals, associated to symbolic objects, are described by quantitative single, interval, nominal, multinominal and /or modal variables (mixed types are allowed). If a dissimilarity matrix is available, it may be used for a numerical clustering.

Taxonomies and hierarchical dependencies defined on variables may be taken into account.

10.3 The Options of HIPYR

10.3.1 The "Variables" window

In this window, the user chooses the variables that will be used to form the classes. HIPYR accepts all types of variables. The appearance and the operation of the window "variables" is the same as in all other ASSO modules.

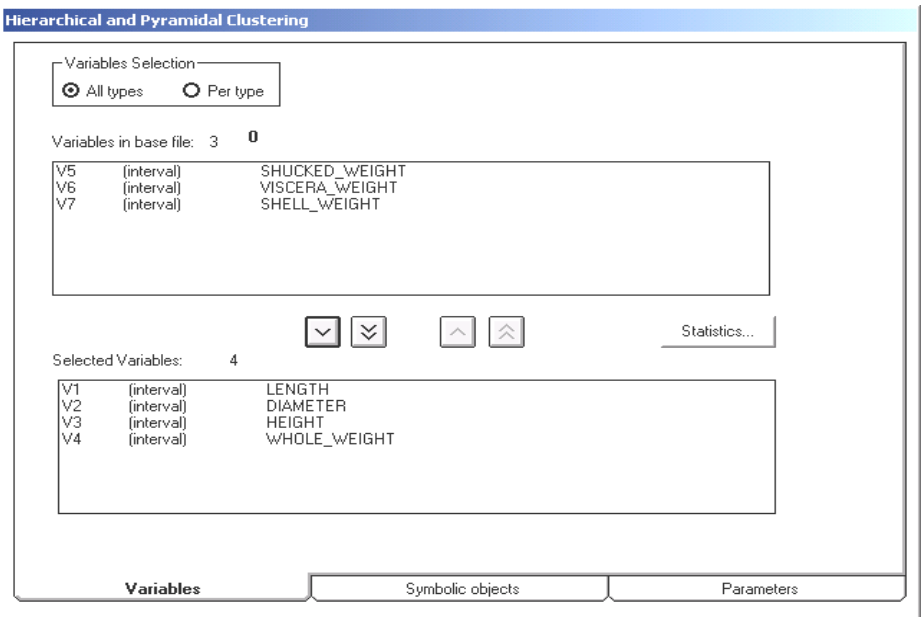


Fig. 1 : Variables Window

10.3.2 The "Symbolic Objects" window

In this window, the user chooses the symbolic objects (rows of the input data array) that will be used to form the classes. The appearance and the operation of the window "symbolic objects" is the same as in all other ASSO modules.

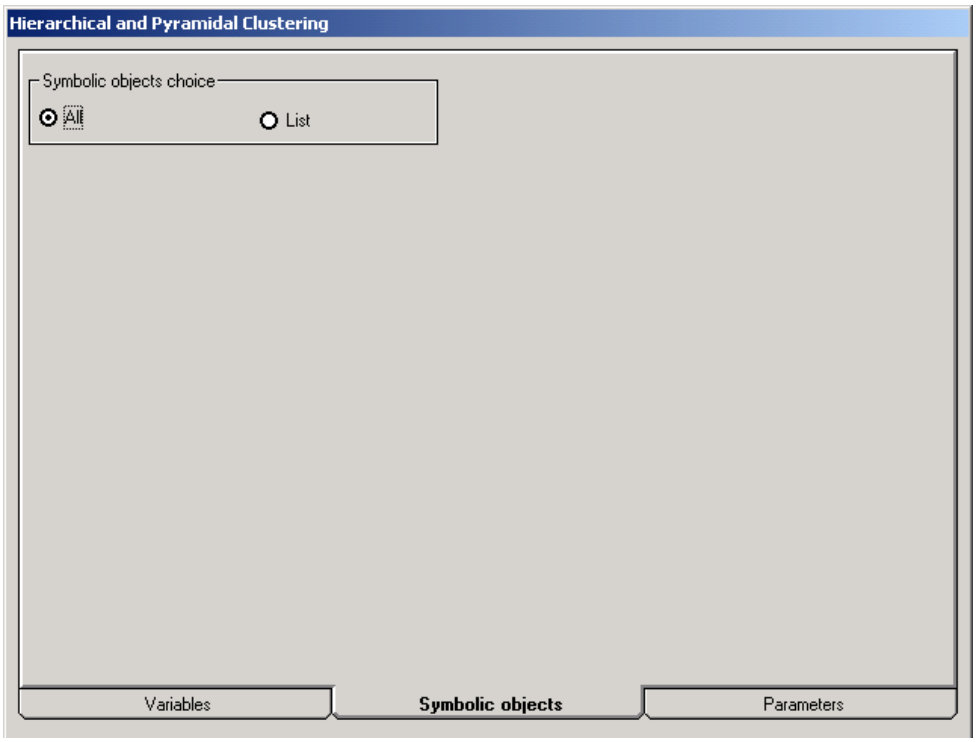


Fig. 2 : Symbolic Objects Window

10.3.3 The "Parameters" window

HIPYR provides several different options for treating data; therefore, the user must specify the values of a variety of parameters which characterize special options.

On the other hand, it is also possible to run HIPYR in a semi-automatic way, i.e., without caring too much about the correct choice of all parameters. In fact, HIPYR provides default (standard) values for all parameters.

The options window of HIPYR looks as follows:

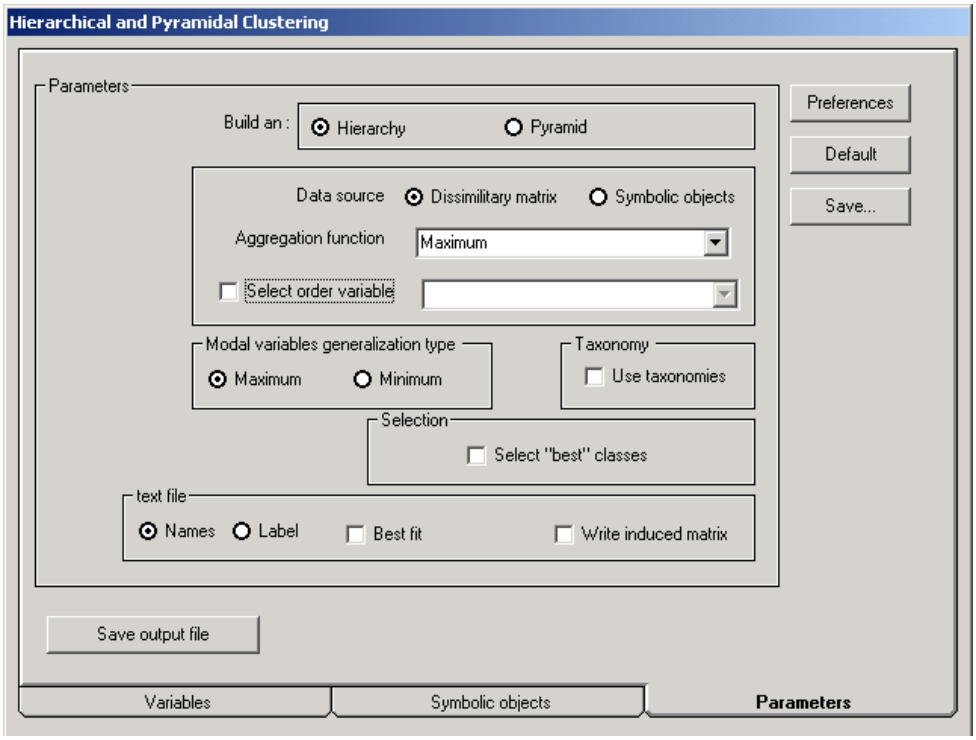


Fig. 3 : HIPYR Options Window

The several options are explained below in the same order as they appear in the window. For a precise explanation of the methods please look at the specification of this module.

Hierarchy / Pyramid

The user must choose to form either a hierarchy or a pyramid and check the corresponding button. By default, a hierarchy is formed.

Data Source

Here, the user must choose whether the clustering is to be made on the basis of a dissimilarity matrix (numerical clustering) or using the symbolic objects (symbolic clustering).

Aggregation Functions

Here, the user must select the aggregation function to be used by the clustering algorithm. For a numerical clustering (data source: dissimilarity matrix) the following options are available: maximum (complete linkage), minimum (single linkage), average linkage and diameter; for a symbolic clustering (data source: symbolic objects) there are two options: generality degree and increment of the generality degree. By default, the maximum is selected for numerical clustering and the generality degree for symbolic clustering.

Select order variable

If the user wants the objects of the data array to be order according to a given order (thus fixing the basis of the pyramid), this button must be checked. Then the user must indicate the variable which contains this order, only quantitative single or ordinal variables are displayed. By default, no order variable is selected.

Modal Variables Generalization Type

If Modal Variables are present, the user may choose whether to perform generalization in a symbolic clustering) by the **Maximum** or by the **Minimum**. Only one of these options may be chosen; by default, generalization is performed by the Maximum.

Taxonomies

If taxonomies are defined for some variables, HIPYR allows the user to use them in the generalization process (symbolic clustering). If this option is checked taxonomies are used, otherwise they are not. By default, taxonomies are not used.

Selection – Select the "best" classes

If this option is checked, the program will automatically produce a set of “best” formed clusters. By default, the selection is not made.

Text File Identification

Here the user must choose whether to use names or labels in the text output file. A “best fit” option is also available. By default, names are used.

Write Induced Matrix

Here the user must indicate whether he wants the induced dissimilarity matrix to be written in the output listing. This is a $n \times n$ triangular matrix. By default, it is not written in the listing.

Save output file

Here the user must indicate the full name of the output file. This file contains the initial data array – top left $n \times p$ array - the description of the clusters formed in the basis of the initial variables (one row for each cluster). Some columns are added to the initial variables: index value, this a quantitative single variable and indicates the height of the cluster (equals 0 for all single individuals); list of cluster members, a multi-nominal variable;

By default, the file is called {<chaining directory>hipyr.xml}. The user is able to change the name and location of that file.

10.4 VPYR

VPYR is the graphical module, which is automatically linked to HIPYR. When opening the graphical representation, the user obtains a window such as the following:

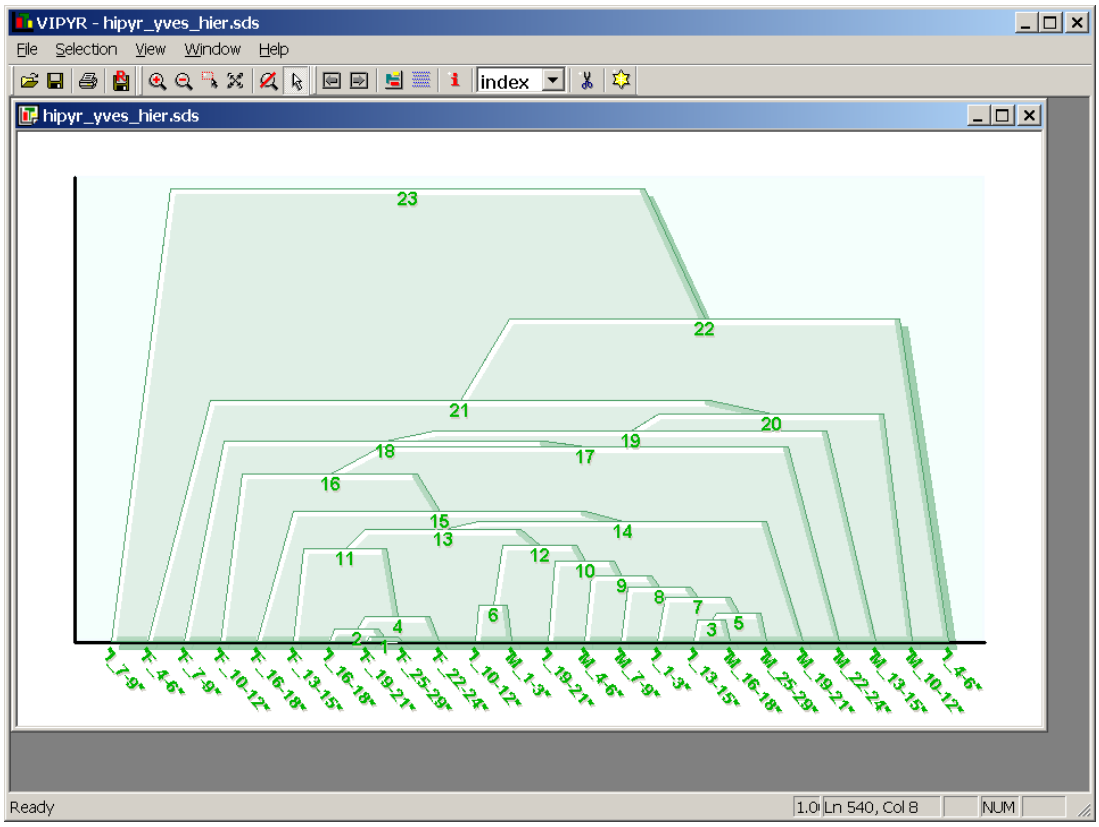


Fig. 4 : Graphical Representation

Several options are then available to explore the structure (hierarchy or pyramid).

10.4.1 Selection

Selection a symbolic object

There is only a way (for now) of selecting a single symbolic object:

In the menu ***Selection » Select Classes***:

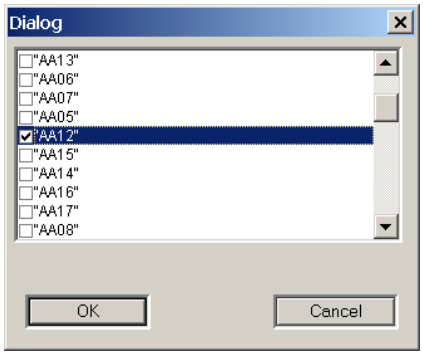


Fig. 5 – Selecting a single object

The user must choose the object (or set of) and then click **OK**.

Selection a class (cluster)

The user can choose a class in the same way as symbolic objects or by clicking in the class area in the graphic, if the arrow button is selected (as in Fig 4).

Getting a Object Information

When an object is selected, click in the **i** button on the menu,

This lunches a variable selection window:

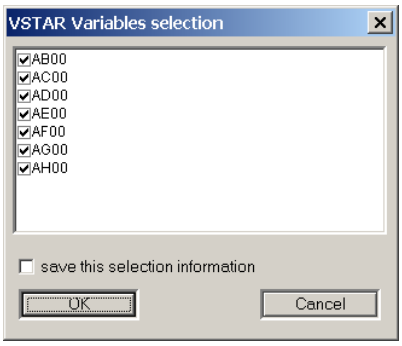


Fig. 6 – Variable selection window

The user must choose the variable he wants to view in the information window and then click **OK**.

This lunches new(s) window(s) with the selected object(s)…

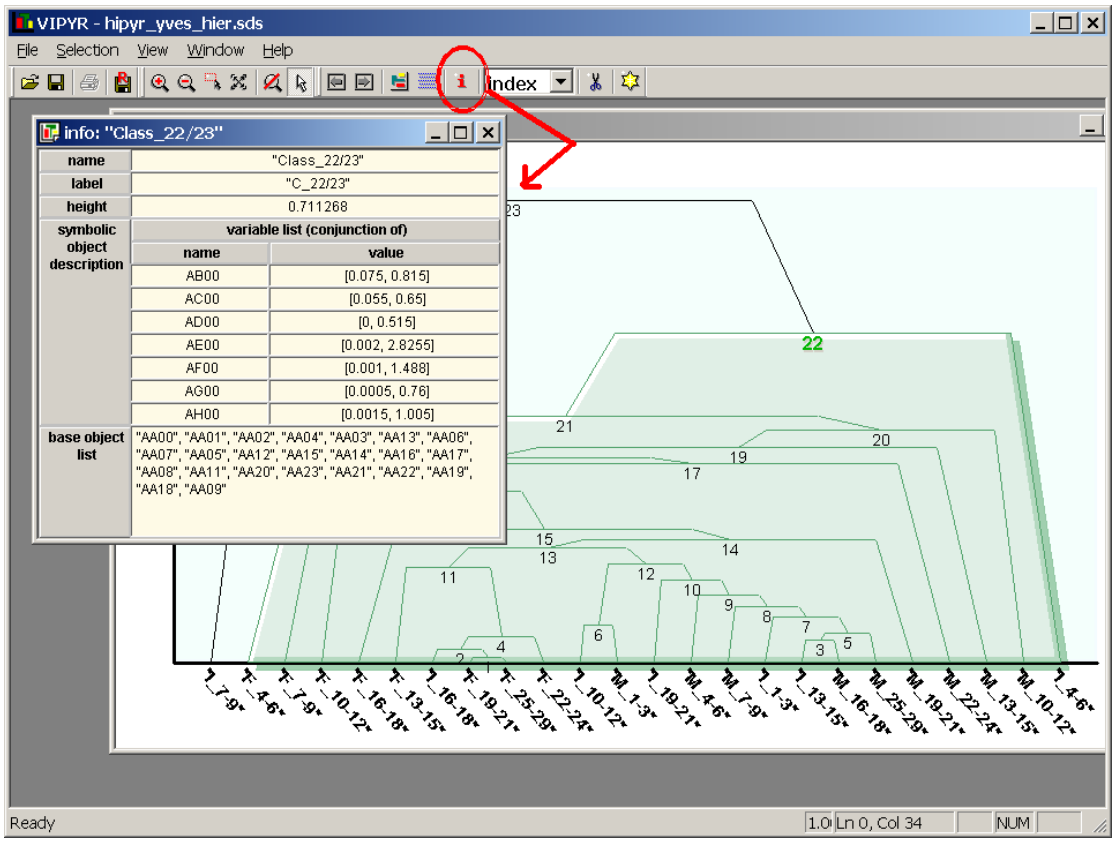


Fig. 7 – Class_22/23 information window (how to lunch)

Graphic Orientation

Selecting the **graphic orientation** button in the menu we can change the way the graphic is represented:

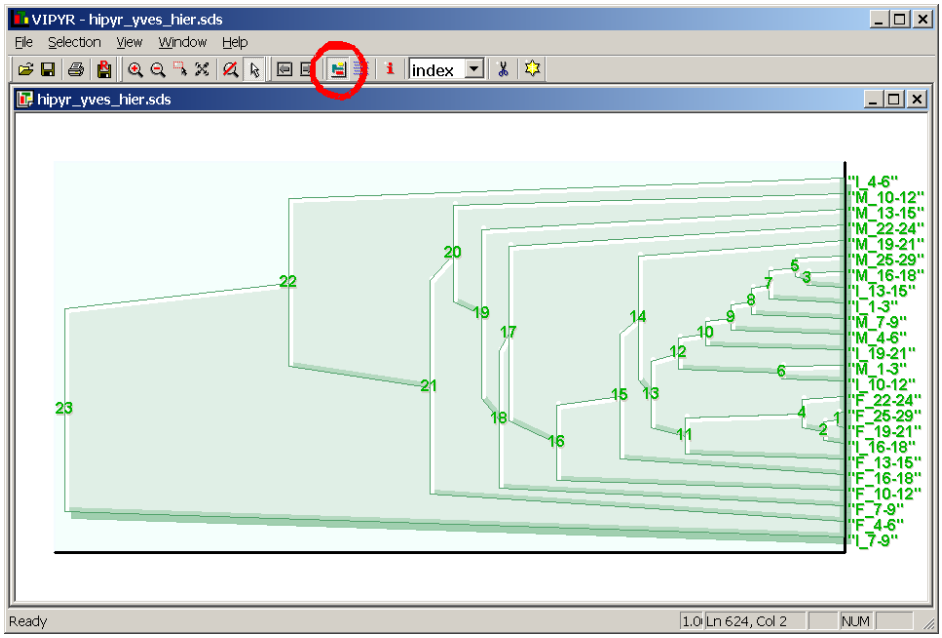


Fig. 8– VIPYR Module (all objects selected in a hierarchy) side image (how to)

Lunching STAR

The user can lunch VSTAR from VPYR, he must have selected object(s) and then click on the **star** button in the menu to lunch the application:

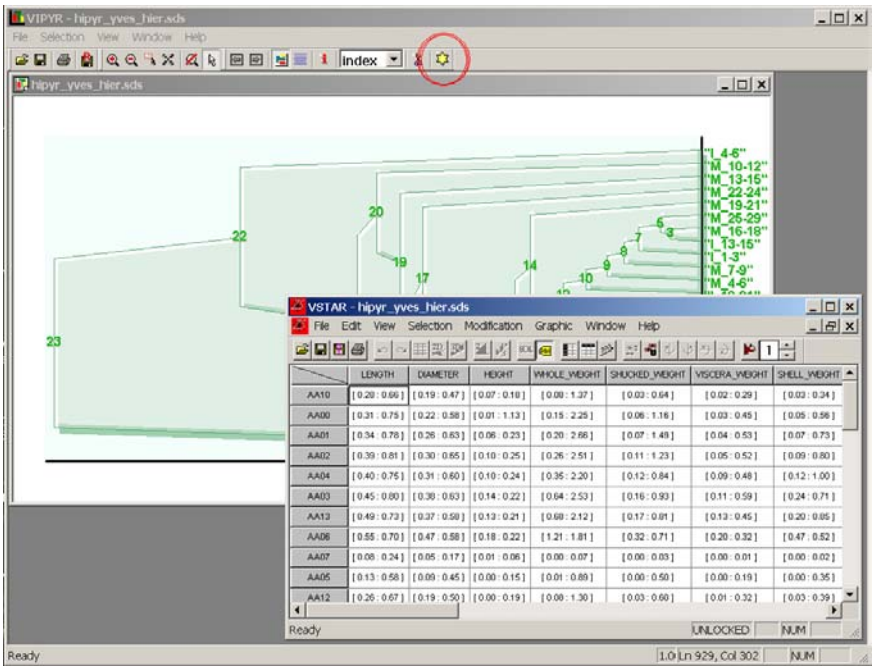


Fig. 9 – Lunching VSTAR from VPYR (how to)

Simplification

The user can make a graphic simplification using the aggregation heights as a criterion.

First, the rate of simplification must be selected, this is done in the options menu:

View »Options

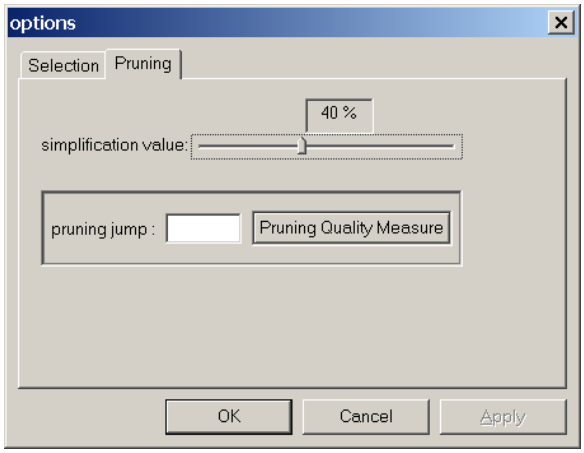


Fig. 10 – The options menu

The simplification value is chosen in the slide bar (in this case 40%).

Then the user must click in the **simplification button** in the menu, this lunches a new graphic window with a simplified graphic

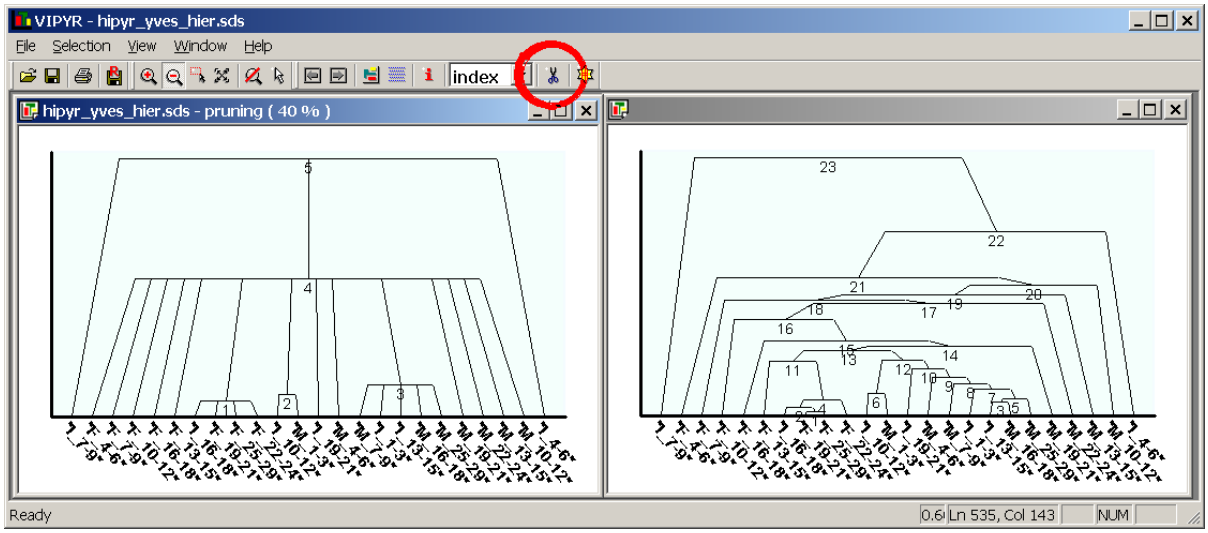


Fig. 11 – The graphic simplification (how to)

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SCLUST Help Guide

Dynamic Clustering

F. A. T. de Carvalho
UFPE
A. Hardy
FUNDP
P. Bertrand and Y. Lechevallier
INRIA
R. Verde
DMS

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **20/05/2003**

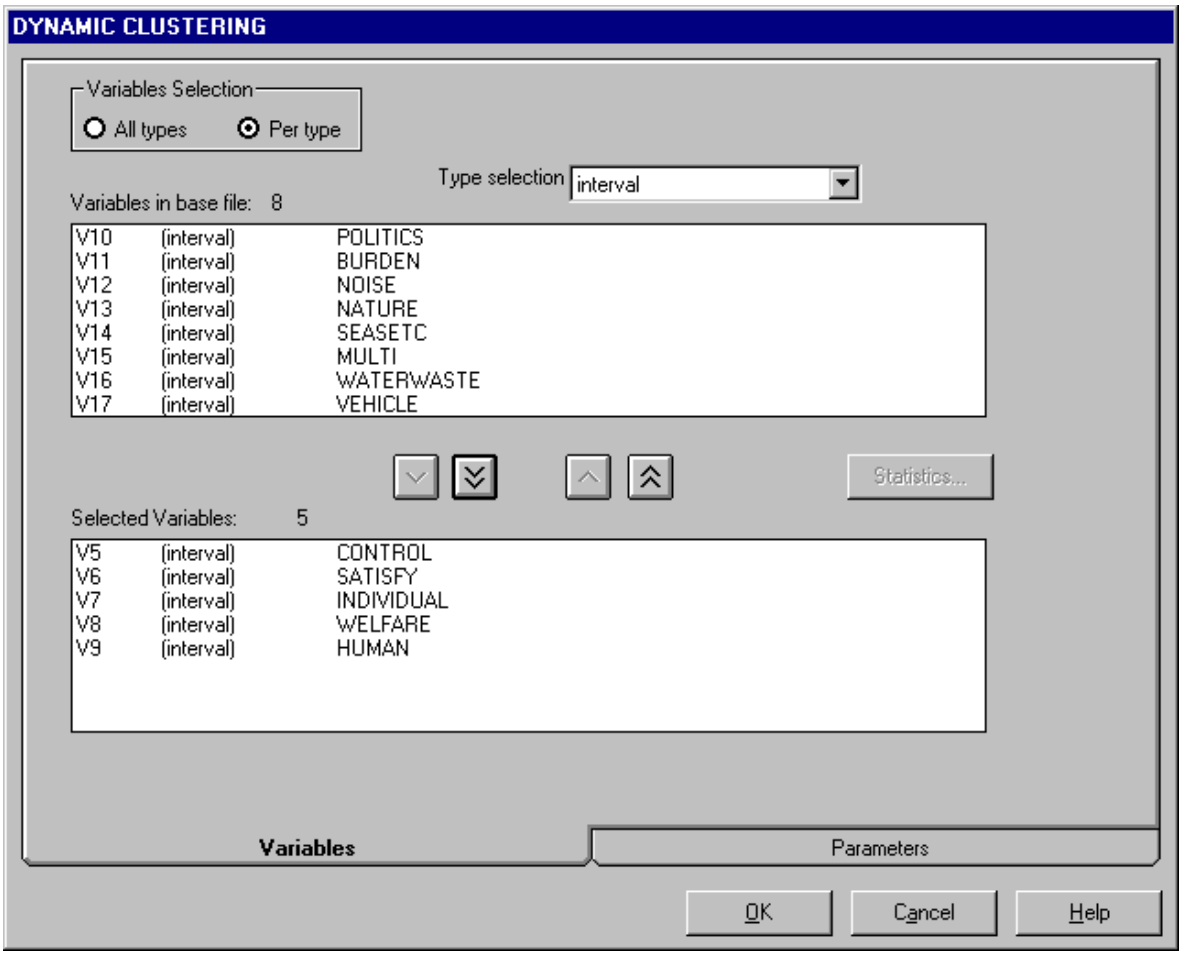
11 SCLUST : Dynamic Clustering

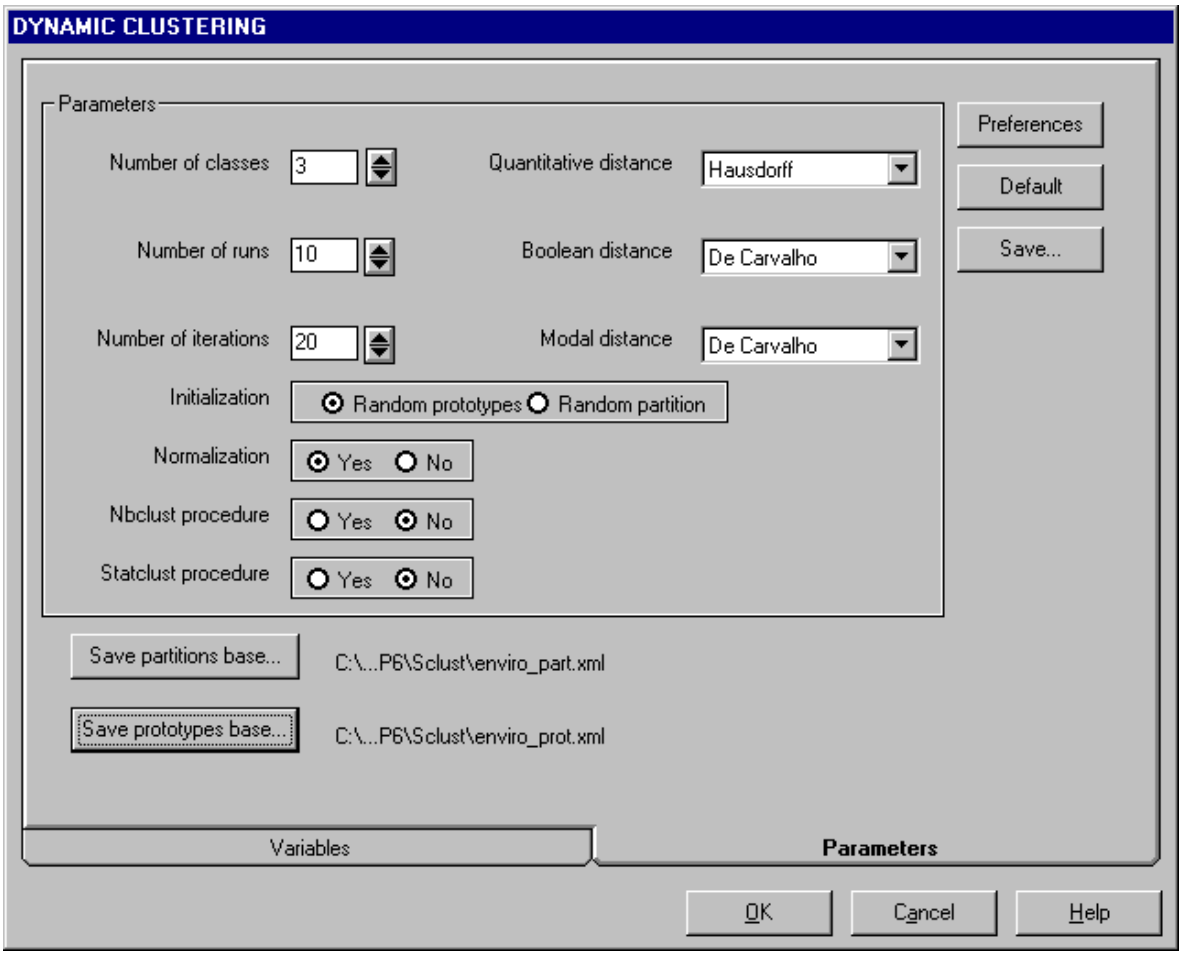
11.1 Introduction

The module SCLUST can be used to partition a set of n p-dimensional symbolic data into a number m of homogeneous clusters.

11.2 Parameters of SCLUST

SCLUST provides several different options for treating data, which differ, e.g., by the definition of dissimilarities, cluster prototypes, update formulas etc. Therefore, the user must specify the values of a variety of parameters which characterize special options. Insofar the user is able to tune the parameters such that the resulting clustering method fits optimally his special application problem.





The several options and parameters are explained below in the same order as they appear in the window.

11.2.1 Number of classes

The number *k* of classes must be fixed, but a different number of classes can be request in order to look for the best partition in *k* classes. That can be performed by moving *k* between two integers corresponding to the minimum and maximum number of classes.

11.2.2 Number of runs

DCLUST treats the n data table in a sequential way. That means, it takes the first object, assigns it to a cluster, takes the second object, assigns it to a cluster, etc. When all n objects are assigned to their clusters DCLUST checks if a local optimal partition is attained. If this is not yet the case, the module DCLUST starts a new cycle in which all *n* objects are processed again in the previous order.

11.2.3 Number of iterations

To prevent an endless loop DCLUST stops after at most a maximal number of iterations or cycles which is determined by this parameter. Increasing this parameter can result in a linear increase in running time.

11.2.4 Initialization

As every clustering method DCLUST assigns each object to that class, which has the “most fitting” prototype, initial class prototypes are needed to start the clustering method. The module offers two different options (“Random prototypes”, “Random partition”).

If the first option is selected, the method will initially create *k* prototypes, whose prototypes are *k* random objects in the set of objects.

If the second option is selected, the method will initially create *k* no empty clusters by randomization of the set of objects.

11.2.5 Distances

The module SCLUST provides a partition of a set of symbolic data according to suitable combination of proximity measure and prototypes. SCLUST allows to select different types of dissimilarities as well as of prototypes, even according to the kind of descriptors We propose three types of distances:

- **Quantitative distance** : the choice is “type L1”, “Euclidian” or” Hausdorff” distances when the type of variables is quantitative single or interval
- **Boolean distance**: the choice is “De Carvalho”, “type L1” or “Euclidian” distances when the type of variables is categorical single or categorical multi-valued.
- **Modal distance**: the choice is “De Carvalho”, “type L1” or “Euclidian” distances when the type of variables is modal.

11.2.6 Normalization

When the set of selected variables contains different type of variables or the unit of the variables is different, DSCLUST proposes to normalize de distance between object and prototype. The normalisation is based on the part of the criteria of each variable is the same when the number of classes is equal to one.

11.2.7 Nbclust procedure

The objective of the Nbclust procedure is to find the best number of "natural" clusters if it is not provided by the user. As for clustering methods, it is not possible to propose a single procedure for the determination of the number of clusters applicable to all types of structure or to all types of data. Furthermore some methods for the determination of the number of clusters are based on tools (for example the convex hull) that are not computable with all types of data, or that may cause problems of complexity in presence of large data sets.

11.2.8 Statclust procedure

Most clustering algorithms can partition a data set into any specified number of clusters even if the data set contains no structure. So one of the most important problems when validating the results of a cluster analysis is: how many clusters are in the data?

11.3 Additional buttons

11.3.1 Preferences

The user is able to save his own default values for the parameters (see also “Save”!). He gets these parameters by pushing this button.

11.3.2 Default

If this button is pushed, all parameters will be set to their default values.

11.3.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

11.3.4 Save partition base …

With this option, the partition of n data intervals into m clusters (produced after running the SCLUST module) is stored in a SODAS file. The user is able to change the name of that file.

11.3.5 Save prototype base …

With this option, the prototypes of all m clusters produced after running the SCLUST module are stored in a SODAS file. The user is able to change the name of that file.

11.4 SCLUST Listing

```
  Sodas The Statistical Package for Symbolic Data Analysis

               Version 1.1  -  22/01/2003

                    MODULE: SCLUST

    Clustering Algorithm on Symbolic Data Table
```

```
Sodas File         :      C:\user\yves\asso\WP6\Div\enviro.sds
Log   File         :      C:\Program Files\DECISIA\SODAS version 2.0\filieres\BJBIF501.LOG
Listing   File     :      C:\Program Files\DECISIA\SODAS version 2.0\filieres\BJBIF501.LST
Sodas Out Cluster  :      C:\user\yves\asso\WP6\Sclust\enviro_part.xml
Sodas Out Prototype:      C:\user\yves\asso\WP6\Sclust\enviro_prot.xml
```

```
 ---------------------------------------------------------
 ---------------------------------------------------------

Learning Set          :    14
Number of variables   :     5
Number of iterations  :    20
Number of classes     :     3
Initialisation        :     0  random partition
Number of runs        :    10
Quantitative distance:      0  Hausdorff Distance
Boolean distance      :     0  De Carvalho Distance
Modal distance        :     0  De Carvalho Distance
Normalize             :     0  No
NBCLUST procedure     :     0  No
STABCLUST procedure   :     0  No

Initial Criterion : 18601.134331

GROUP OF SELECTED VARIABLES :
============================

( Pos )     Tj        Tj      Weight     Name             Type
          initial    used

(   5 )    13.58    13.58   1.000000     CONTROL          INTERVAL
(   6 )    19.68    19.68   1.000000     SATISFY          INTERVAL
(   7 )    24.25    24.25   1.000000     INDIVIDUAL       INTERVAL
(   8 )    20.81    20.81   1.000000     WELFARE          INTERVAL
(   9 )    21.69    21.69   1.000000     HUMAN            INTERVAL


LIST OF SYMBOLIC OBJECTS IN THE SET :
=====================================

"1241"     "1242"     "1441"     "1442"     "1641"     "1642"     "1741"
"2241"     "2242"     "2441"     "2442"     "2641"     "2642"     "2741"

 RUN NUMBER :    1
 =================

 Iteration Permutation Criterion
     1          4     14226.046698
     2          0     12211.055450

 RUN NUMBER :    5
 =================

 Iteration Permutation Criterion
     1          4     15924.259380
     2          1     14462.050129
     3          3     13410.470234
     4          1     12233.928185
     5          0     12233.928196
 RUN NUMBER :   10
```

```
  =================

  Iteration Permutation Criterion
      1          5     15817.687008
      2          2     13816.490578
      3          1     13193.366264
      4          0     12958.187763

  OPTIMAL SOLUTION
  ================

  RUN NUMBER :  1
  CRITERION  :  12211.055450


  EDITION OPTIMAL PARTITION
  =========================

 Classe :   1  Cardinal :      5
 ================================

 (  0) "1241"    [1.6]  (  1) "1242"    [0.9]  (  2) "1441"    [0.4]
(  3) "1442"    [1.3]  (  5) "1642"    [0.9]
  Classe :   2  Cardinal :      3
 ================================

 (  4) "1641"    [1.4]  (  6) "1741"    [0.9]  ( 13) "2741"    [0.7]

 Classe :   3  Cardinal :      6
 ================================

 (  7) "2241"    [1.1]  (  8) "2242"    [0.8]  (  9) "2441"    [0.7]
( 10) "2442"    [0.7]  ( 11) "2641"    [1.2]  ( 12) "2642"    [1.6]

  EDITION PROTOTYPES
  ==================


 PARTITION DESCRIPTION
 =====================

 DISPERSION : 18601.134331
 CRITERION  : 12211.055450
 Pourcentage of dispersion :    34.35




 VARIABLES DESCRIPTION
```

```
====================

Position!    Name            Bj/Tj      Wj/W      Tj/T    Quality
(   5 )      CONTROL         12.45      4.92     13.58    -63.76
(   6 )      SATISFY         40.25     23.05     19.68     17.16
(   7 )      INDIVIDUAL      45.26     31.95     24.25     31.76
(   8 )      WELFARE         44.59     27.01     20.81     29.81
(   9 )      HUMAN           20.69     13.06     21.69    -39.76


CLUSTER DESCRIPTION
===================

 Cluster  Size(Nk)  Bk/Tk       Wk/W     Tk/T     Bk/Nk.B   Wk/Nk.W
     1        5    30.13      35.92     33.75     7.185      6.750
     2        3    48.36      21.53     27.36     7.176      9.121
     3        6    28.17      42.55     38.88     7.092      6.481

EDITION PROTOTYPES BY VARIABLES
===============================

Variable ( 5 ) CONTROL

Cluster    Minimum      Maximum  Wkj/Tkj  Wkj/Wj
Set        -211.74       171.86
   1       -252.53       147.19   94.65  32.61
   2       -118.64       279.98   77.65  16.99
   3       -113.82       200.57   87.07  50.41


Variable ( 6 ) SATISFY

Cluster    Minimum      Maximum  Wkj/Tkj  Wkj/Wj
Set        -196.70       165.75
   1       -102.05       282.70   70.88  45.84
   2         -2.41       392.73   66.71  20.80
   3       -447.96      -136.84   46.65  33.36


Variable ( 7 ) INDIVIDUAL

Cluster    Minimum      Maximum  Wkj/Tkj  Wkj/Wj
Set        -222.69       119.08
   1       -637.09      -254.23   33.25  27.19
   2        222.18       555.42   52.41  27.75
   3       -160.68       128.60   93.96  45.06



Variable ( 8 ) WELFARE
```

Cluster Set	Minimum	Maximum	Wkj/Tkj	Wkj/Wj
	-157.14	210.30		
1	-180.60	233.76	88.50	23.71
2	-623.18	-179.24	39.02	32.08
3	117.65	374.35	61.84	44.20

Variable (9) HUMAN

Cluster Set	Minimum	Maximum	Wkj/Tkj	Wkj/Wj
	-194.30	157.62		
1	-204.94	168.26	98.35	46.36
2	-547.06	-126.65	49.76	13.28
3	-60.47	229.33	77.22	40.36

PROTOTYPE GRAPHICAL REPRESENTATIONS BY VARIABLES

Variable (5) CONTROL , PS files : cluster3x5.ps

Variable (6) SATISFY , PS files : cluster3x6.ps

Variable (7) INDIVIDUAL , PS files : cluster3x7.ps

Variable (8) WELFARE , PS files : cluster3x8.ps

Variable (9) HUMAN , PS files : cluster3x9.ps

CRITERION

Run	Iteration	Class	Criterion
1	2	3	12211.055450
2	2	3	12233.928196
3	2	3	12654.142811
4	4	3	12211.055450
5	5	3	12233.928196
6	5	3	13550.219282
7	4	3	12796.281860
8	2	3	15288.342253
9	3	3	12211.055450
10	4	3	12958.187763

Statistics on criterion distribution :

Minimum	12211.055450
Means	12834.819671
Maximum	15288.342253
Standard deviation	920.342609

Interpretation of the partition of 60 meteorological stations in Chine

The proposed aids to the interpretation of a partition of symbolic data has been realised on a set of data by *Long-Term Instrumental Climatic Data Base of the People's Republic of China* (*http://dss.ucar.edu/datasets/ds578,5/data*). This set of data contains the monthly temperatures observed in 60 meteorological stations of China. According a natural representation of the temperatures, they are coded in a table as the interval of the minima and maxima for each month. For our example we have considered the temperatures of the year 1988 and we have built a table of dimension 60 rows and 12 columns, corresponding to the number of stations and to the number of months of the year.

The different quality and contribution indexes have been computed on an example of partition in 5 classes obtained by a dynamical partitioning algorithm (…) on symbolic data described by interval variables.

For instance, the station "ChangSha" is described by the 12 intervals of the monthly temperatures:

[January = [2.7:7.4]] ^[February = [3.1:7.7]]
^[March = [6.5:12.6]] ^[April = [12.9:22.9]]
^[May = [19.2:26.8]] ^[June = [21.9:31]]
^[July = [25.7:34.8]] ^[August = [24.4:32]]
^[September = [20:27]] ^[October = [15.3:22.8]]
^[November = [7.6:19.6]] ^[December = [4.1:13.3]]

In the table 1 are collected the descriptions of the 60 meteorological stations:

Meteorological stations	***January***	***February***		***December***
AnQing	[1.8 : 7.1]	[5.2 : 11.2]		[4.3 : 11.8]
BaoDing	[-7.1 : 1.7]	[-5.3 : 4.8]		[-3.9 : 5.2]
BeiJing	[-7.2 : 2.1]	[-5.3 : 4.8]		[-4.4 : 4.7]
BoKeTu	[-23.4 : -15.5]	[-24 : -14]		[-21.1 : -13.1]
ChangChun	[-16.9 : -6.7]	[-17.6 : -6.8]		[-15.9 : -7.2]
ChangSha	[2.7:7.4]	[3.1:7.7]		[4.1:13.3]
ZhiJiang	[2.7 : 8.2]	[2.7 : 8.7]		[5.1 : 13.3]

Tab. 1: Minima and maxima monthly temperatures recorded by the 60 meteorological stations

Fixed the number of classes to 5, the algorithm is reiterated 50 times and the best solution is found for the minimum value of the criterion equal to: Δ=3848.97.

It is worth to noting that the obtain partition of the 60 elements follows the geographical contiguity of the stations.

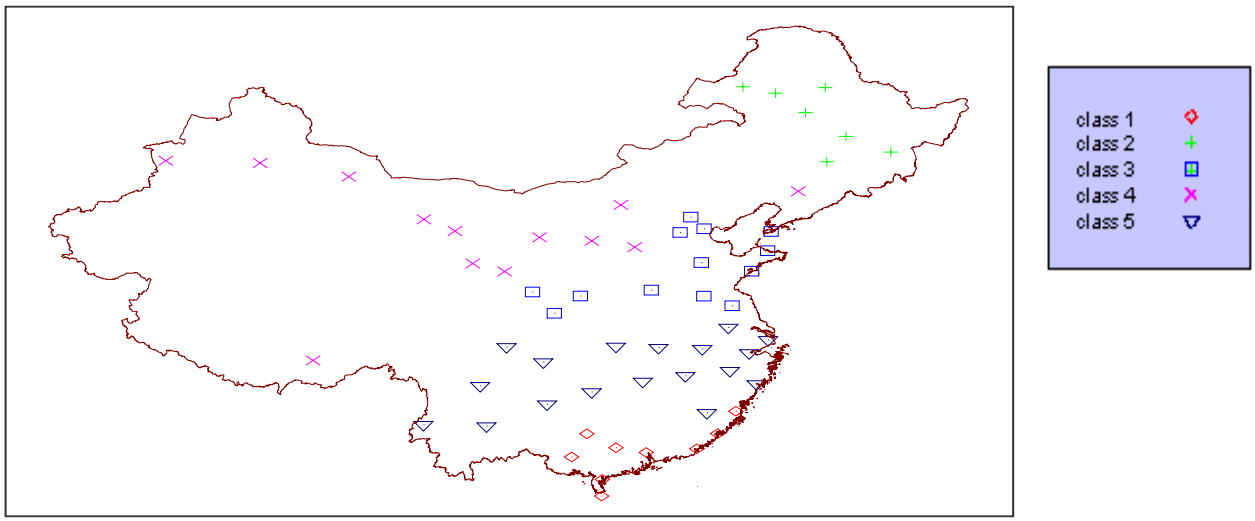


Fig. 1: Visualization on the Map of the China of the 5 Classes of the partition of the 60 meteorological stations

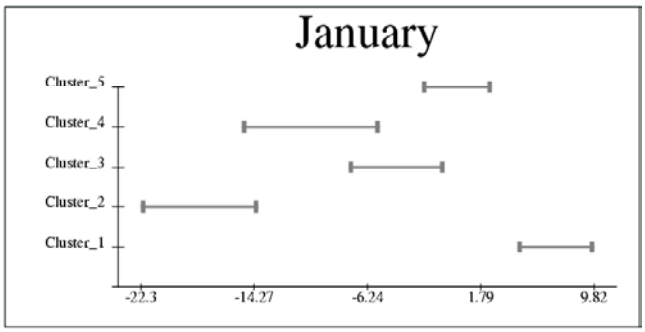
According to the kind of representation of the classes by intervals proposed in the partitioning algorithm on interval data, the prototype of each class is the interval which minimizes the Hausdroff distances from all the elements belonging to the class.

In the table 2 we have indicated the values of the different indices of quality and contribution proposed in the present paper.

Variable	Quality	Contribution with P	Contribution with E
January	69.50	13.76	12.74
February	66.18	12.63	12.28
March	64.52	9.30	9.27
April	64.36	6.74	6.73
May	61.68	6.15	6.42
June	53.36	4.56	5.50
July	46.31	4.05	5.63
August	47.19	3.73	5.08
September	61.10	6.05	6.37
October	70.41	8.97	8.19
November	70.63	10.79	9.83
December	71.33	13.26	11.96

*Tab. 2 Quality and contribution measures (*times *100) of the intervals of temperatures observed in the 12 months to the partition of the stations in 5 classes.*

We can observe that the wintering months are more discriminant of the cluster (high value of the quality index) than the summering ones. In the figures 2 the prototypes of the classes computed on the interval values of January and December are much more separated than the ones in figure 3 corresponding to the prototype of temperatures of June and September.



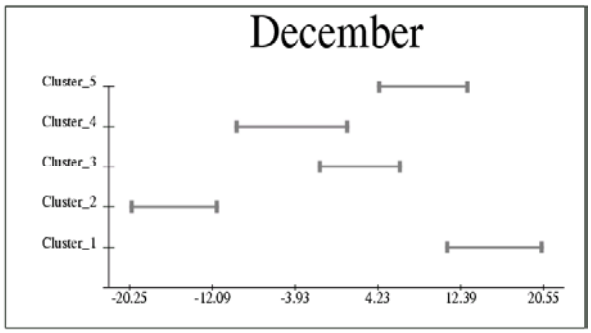
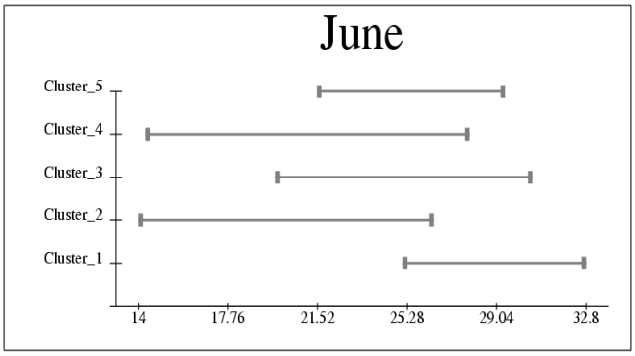


Fig. 2



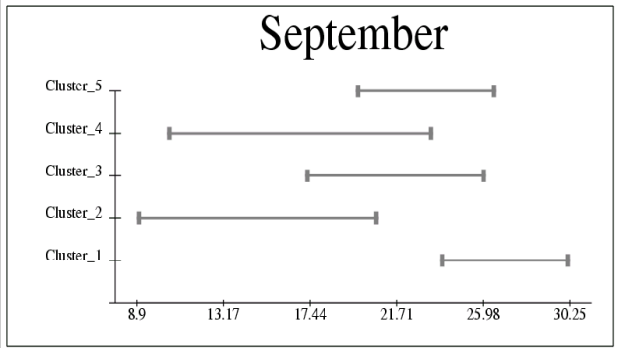


Fig. 3

Bibliography

Celeux, G., Diday, E., Govaert, G., Lechevallier, Y., Ralambondrainy, H. (1989): Classification Automatique des Données. Bordas, Paris.

De Carvalho, F.A.T, Verde, R. and Lechevallier, Y. (2001): Deux nouvelles méthodes de classification automatique d'ensembles d'objets symboliques décrits par des variables intervalles. SFC'2001, Guadeloupe.

Diday, E. (1971): Le méthode des Nuées dynamique, in Revue de Statistique Appliquée, 19, 2, 19-34.

Verde, R., De Carvalho, F.A.T., Lechevallier, Y. (2000) : A Dynamical Clustering Algorithm for Multi-Nominal Data. In : H.A.L. Kiers, J.-P. Rasson, P.J.F. Groenen and M. Schader (Eds.): Data Analysis, Classification, and Related Methods, Springer-Verlag, Heidelberg, 387-394.

Verde, R., De Carvalho, F.A.T., Lechevallier, Y. (2001): A dynamical clustering algorithm for symbolic data. Tutorial Symbolic Data Analysis, GfKl Conference, Munich.

Milligan, G.W, Cooper, M.C. (1985): An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, Vol 50, pp 159-179.

Rasson, J.P. and Kubushishi, T. (1994): The gap test: an optimal method for determining the number of natural classes in cluster analysis. In: E. Diday, Y. Lechevallier, M. Schader, P. Bertrand, B. Burtschy (Eds): New approaches in classification and data analysis. Springer Verlag, Berlin, pp 186-194.

Hardy, A. (1994): An examination of procedures for determining the number of clusters in a data set. In: E. Diday, Y. Lechevallier, M. Schader, P. Bertrand, B. Burtschy (Eds): New approaches in classification and data analysis. Springer Verlag, Berlin, pp 178-185.

Hardy, A., Andre, P. (1998): An investigation of nine procedures for detecting the structure in a data set. In: A. Rizzi, M. Vichi, H.-H. Bock (Eds): *Proceedings of the 6th Conference of the International Federation of Classification Societies*. Springer-Verlag, Berlin, pp 29-36.

Gordon, A.D. (1998). How many clusters? An investigation of five procedures for detecting nested cluster structure. In: *Proceedings of the IFCS-96 Conference*. Kobe, 109-116.

Gordon, A. D. (1994): Identifying genuine clusters in a classification. *Computational*

Statistics and Data Analysis, Vol 18, pp 561--581.

Gordon, A. D. (1996): Null models in cluster validation. In: W. Gaul and D. Pfeifer (Eds.):

From Data to Knowledge: Theoretical and Practical Aspects of Classification, *Data Analysis, and Knowledge Organization*. Springer, Berlin, pp 32--44.

Bel Mufti, G. (1998) *Validation d'une classe par estimation de sa stabilité*. Thèse,

Université Paris-Dauphine, Paris.

Bel Mufti, G., Bertrand P. (2003) Stability analysis of individual clusters.

Chavent M., De Carvalho, F.A.T, Verde, R. and Lechevallier, Y. (2003): Trois nouvelles méthodes de classification automatique de données symboliques de type intervalle. *Revue de Statistique Appliquée* , LI 4, pp 5-29.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

DCLUST Help guide

Clustering Algorithm based on Distance Tables

F. A. T. de Carvalho
UFPE
Y. Lechevallier
INRIA

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **20/05/2003**

12 DCLUST : Clustering Algorithm based on Distance Tables

12.1 Introduction

The aim of the module CDIS is to cluster interactively a large set of symbolic objects into a reduced (fixed) number of homogeneous classes, on the basis of a proximity table.

We use a clustering criterion which is based on the sum of dissimilarities between the individuals belonging to the same cluster, and try to minimize this clustering criterion by a suitable choice of the classes.

These proximity functions (similarity or dissimilarity) may take into account dependencies (hierarchical or logical) between variables.

Algorithm :

The clustering method uses an optimality criterion which is based on the sum of distances between the individuals belonging to the same cluster.

The dissimilarity tables is given:

1) by procedures from SODAS software (e.g., DI and DIM) or from ASSO by DISS or MATCH;
2) or directly by proximity functions which can take into consideration dependencies between variables.

12.2 The input to the DCLUST module

The input for the module DCLUST is a ASSO file which contains a distance table.

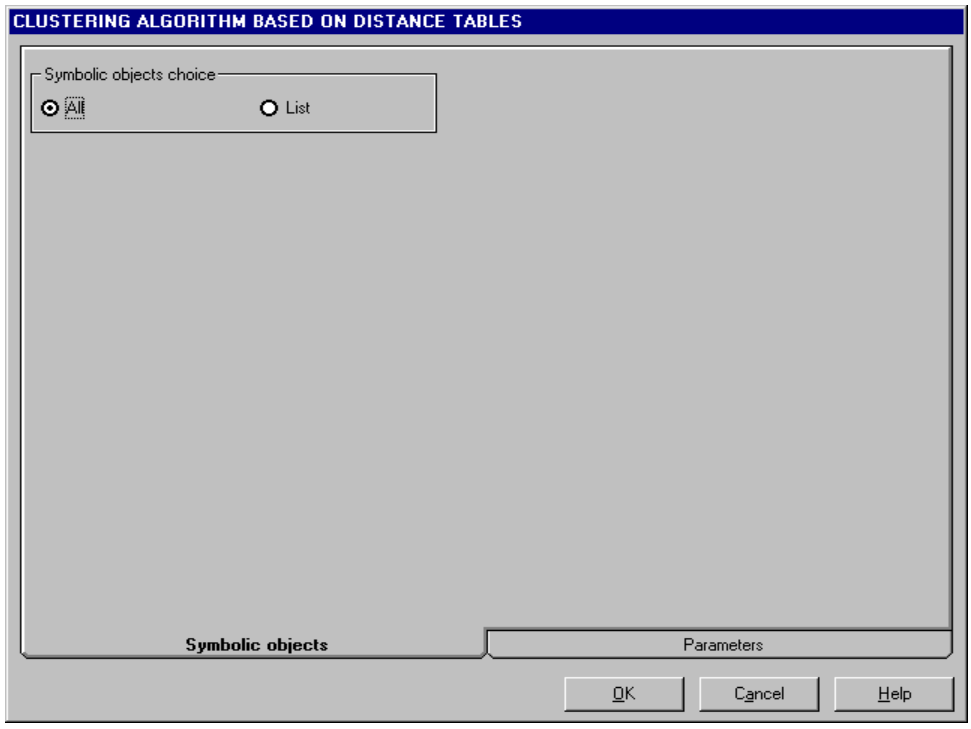
12.3 Parameters of DCLUST

DCLUST provides several different options for treating data, which differ, e.g., by the definition of dissimilarities, cluster prototypes, update formulas etc. Therefore, the user must specify the values of a variety of parameters which characterize special options. Insofar the user is able to tune the parameters such that the resulting clustering method fits optimally his special application problem.

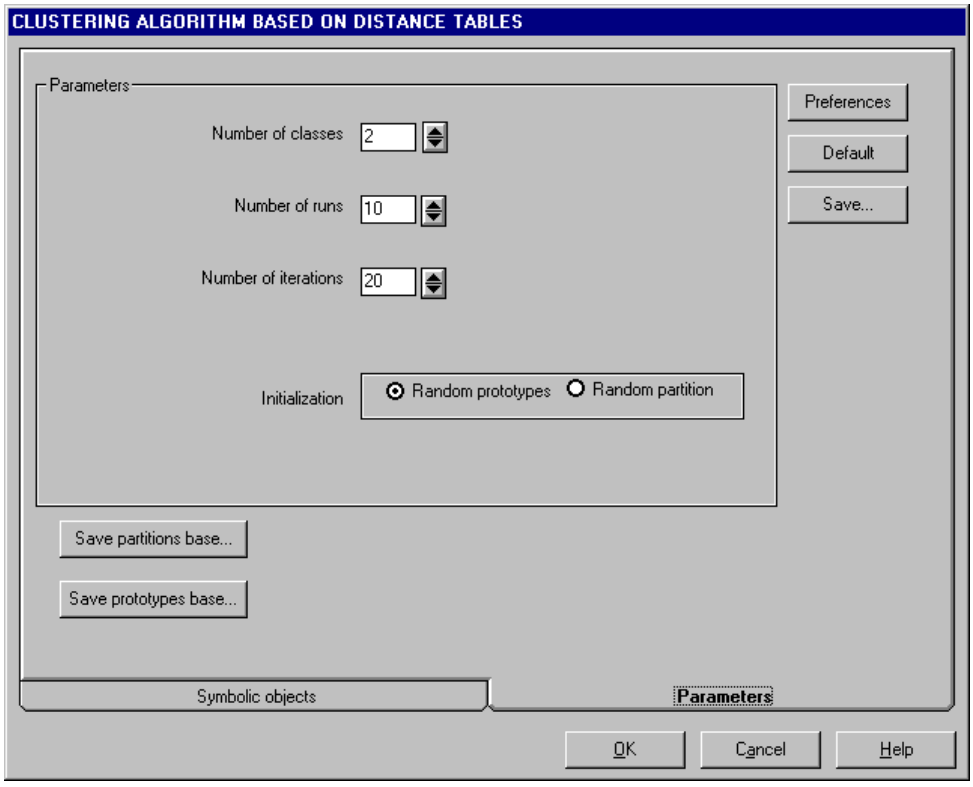
On the other hand, it is also possible to run DCLUST in a automatic way, i.e., without caring too much about the correct choice of all parameters, or just by tuning only a part of the parameters: In fact, DCLUST provides default (standard) values for all parameters, and replaces non-specified parameter values by standard values.

Besides, the menu of DCLUST ensures that the selected parameter combinations are consistent. That means, that it is not possible to combine some parameters in a way that makes no sense or is contradictory. That is why in some cases a couple of parameters cannot be chosen and appears *grey* in the parameter window

The **Symbolic object window** of DCLUST looks as follows:



The **parameter** window of DCLUST looks as follows:



The several options and parameters are explained below in the same order as they appear in the window.

12.3.1 Number of classes

The number *k* of classes must be fixed, but a different number of classes can be request in order to look for the best partition in *k* classes. That can be performed by moving *k* between two integers corresponding to the minimum and maximum number of classes.

12.3.2 Number of runs

DCLUST treats the n data table in a sequential way. That means, it takes the first object, assigns it to a cluster, takes the second object, assigns it to a cluster, etc. When all n objects are assigned to their clusters DCLUST checks if a local optimal partition is attained. If this is not yet the case, the module DCLUST starts a new cycle in which all *n* objects are processed again in the previous order.

12.3.3 Number of iterations

To prevent an endless loop DCLUST stops after at most a maximal number of iterations or cycles which is determined by this parameter. Increasing this parameter can result in a linear increase in running time.

12.3.4 Initialization

As every clustering method DCLUST assigns each object to that class, which has the “most fitting” prototype, initial class prototypes are needed to start the clustering method. The module offers two different options (“Random prototypes”, “Random partition”).

If the first option is selected, the method will initially create *k* prototypes, whose prototypes are *k* random objects in the set of objects.

If the second option is selected, the method will initially create *k* no empty clusters by randomization of the set of objects.

12.4 Additional buttons

12.4.1 Preferences

The user is able to save his own default values for the parameters (see also “Save”!). He gets these parameters by pushing this button.

12.4.2 Default

If this button is pushed, all parameters will be set to their default values.

12.4.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

12.4.4 Save partition base …

With this option, the partition of n data intervals into m clusters (produced after running the DCLUST module) is stored in a SODAS file. The user is able to change the name of that file.

12.4.5 Save prototype base …

With this option, the prototypes of all m clusters produced after running the DCLUST module are stored in a SODAS file. The user is able to change the name of that file.

12.4.6 Output Data and Results

If the input is just a given dissimilarity table it is possible only to use the alternative algorithm DCLUST. There are no symbolic description of the classes. In that case, the DCLUST algorithm produces as output a file .sds or .xml containing the original symbolic data table where it's added an indicator variable with the index of the classes of the final partition.

If the input is the symbolic data table plus the proximity table, the basic DCLUST algorithm produces as output a file .sds containing a new set of symbolic objects: the clusters of the final partition, which are described by a description vector of the corresponding prototypes. In both basic and alternative DCLUST algorithm, it is added to the original symbolic data table an indicator variable with the index of the classes of the final partition.

Supplementary results are furnished in a text file as:

- List of classes (membership list, binary membership vector, class summary),
- Description vector associated to the prototypes
- The relation for each variable
- The extension mapping.

Bibliography

De Carvalho, F.A.T. (1994): Proximity coefficients between Boolean symbolic objects, in New Approaches in Classification and Data Analysis, Diday et al. (Eds.), Springer Verlag, Heidelberg, 387-394.

De Carvalho, F.A.T., Souza, R. M. C. (1998): Statistical proximity functions of Boolean symbolic objects based on histograms. In: Rizzi, A., Vichi, M., Bock, H.-H. (Eds.): Advances in Data Science and Classification, Springer-Verlag, Heidelberg, 391-396

De Carvalho, F.A.T, Verde, R. and Lechevallier, Y. (1999): A dynamical clustering of symbolic objects based on a context dependent proximity measure. In: Bacelar-Nicolau, H., Nicolau, F.C. and Janssen, J. (Eds.): Proc. IX International Symposium - ASMDA'99. LEAD, Univ. de Lisboa, 237--242.

Diday, E., Govaert, G., Lechevallier, Y. and Sidi, J. (1980): Clustering in pattern recognition, NATO Advanced study Institute on Digital Image Processing and Analysis, Bonas. Available at INRIA-Rocquencourt

Ichino, M., Yaguchi, H. (1994): Generalized Minkowsky Metrics for Mixed Feature Type Data Analysis. IEEE Transactions System, Man and Cybernetics 24, 698-708.

Verde, R., De Carvalho, F.A.T., Lechevallier, Y. (2001): A dynamical clustering algorithm for symbolic data. Tutorial Symbolic Data Analysis, GfKl Conference, Munich.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SYKSOM Help Guide

Kohonen Self-Organising

Edited by RWTH

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **6/06/2003**

13 SYKSOM : Kohonen Self-Organising

13.1 Introduction

The module SYKSOM can be used to partition a set of n p-dimensional interval data into a number m of homogeneous clusters in a way analogous to the construction of Kohonen maps. The main emphasis of such methods is oriented towards a graphical visualization of the clusters, their (dis-)similarities and their properties by representing the clusters in a two-dimensional rectangular lattice (map) of size a×b (such that m=ab) such that clusters with a high similarity are grouped together in about the same location of the lattice. The properties of the clusters can then be visualized by the ASSO module VMAP, e.g., by clicking on the corresponding vertices in the map.

13.2 The input to the SYKSOM module

The input for the module SYKSOM is a sequence of n p-dimensional *interval* data (p-dimensional rectangles) of the type [u,v] where u and v are p-dimensional real-valued vectors. If u_1,...., u_p and v_1,...,v_p are the p coordinates of u and v, respectively, the closed interval [u_i, v_i] characterizes the observation (data) for the i-th underlying variable.

Furthermore SYKSOM is able to process p-dimensional *single-valued* data, converting a single data point x to a p-dimensional interval [x,x].

Therefore, SYKSOM accepts, in the window "variables", interval-type and single-valued **variables.** The appearance and the operation of the window "variables" is the same as in all other ASSO modules.

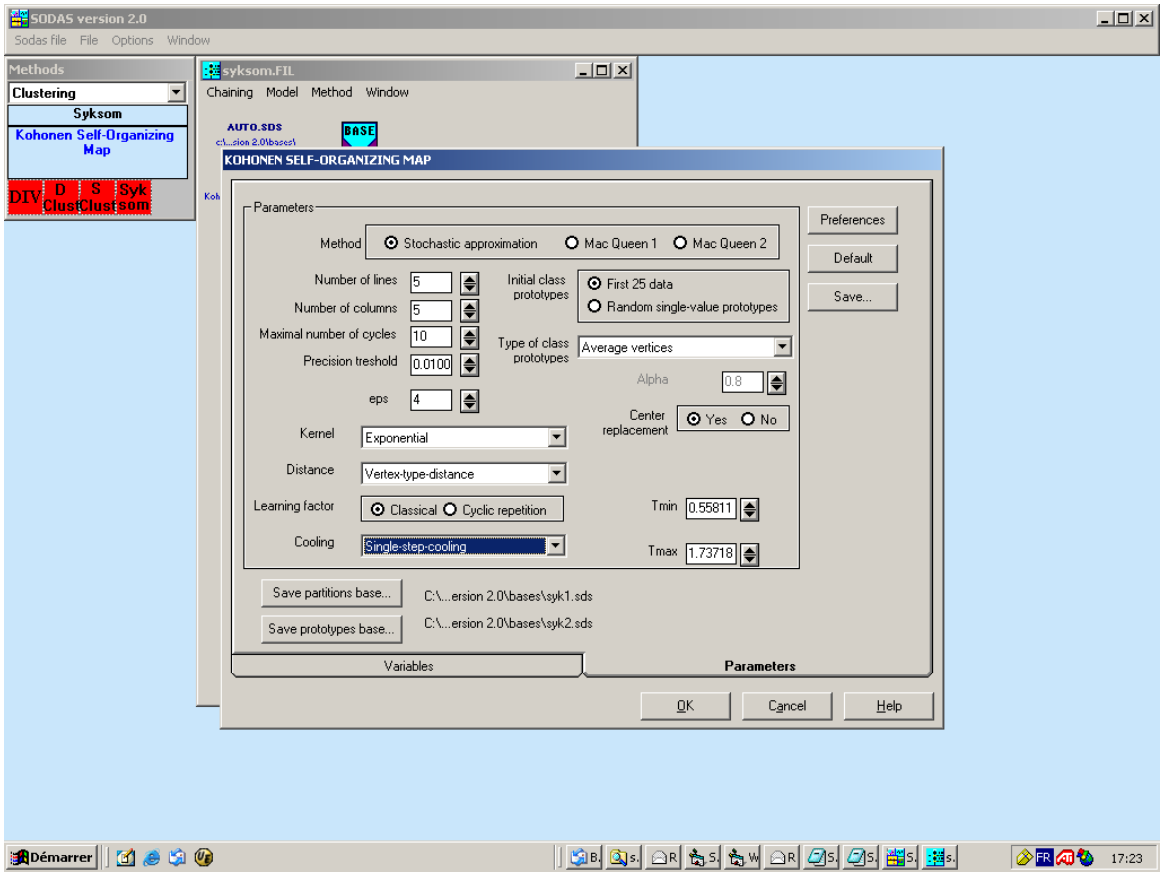
13.3 The parameters of SYKSOM

SYKSOM provides several different options for treating data, which differ, e.g., by the definition of dissimilarities, cluster prototypes, update formulas etc. Therefore, the user must specify the values of a variety of parameters which characterize special options. Insofar the user is able to tune the parameters such that the resulting clustering method fits optimally his special application problem.

On the other hand, it is also possible to run SYKSOM in a automatic way, i.e., without caring too much about the correct choice of all parameters, or just by tuning only a part of the parameters: In fact, SYKSOM provides default (standard) values for all parameters, and replaces non-specified parameter values by standard values.

Besides, the menu of SYKSOM ensures that the selected parameter combinations are consistent. That means, that it is not possible to combine some parameters in a way that makes no sense or is contradictory. That is why in some cases a couple of parameters cannot be chosen and appears *grey* in the parameter window.

The parameter window of SYKSOM looks as follows:



The several options and parameters are explained below in the same order as they appear in the window. For a precise explanation of the methods and parameters please look at the specification of this module ("SYKSOM, An outline with options and formulas").

13.3.1 Method

The module SYKSOM provides three different clustering methods, which are called "Stochastic approximation", "Mac Queen 1" and "Mac Queen 2". The selection of method has influence on other parameters, in particular, on the list of selectable parameters.

13.3.2 Number of lines

It is the purpose of SYKSOM to visualize the overall structure of the p-dimensional data in the two-dimensional plane. For this a fixed rectangular lattice is used which has a prespecified number of rows (horizontal lines) and columns (vertical lines) whereas the vertices correspond to the clusters. The parameter 'number of lines' determines how many horizontal lines this lattice consists of. The whole number of cluster is the product of "Number of lines" and "Number of columns".

13.3.3 Number of columns (see also "Number of lines"!)

This parameter determines how many vertical columns this lattice consists of.

13.3.4 Maximal number of cycles

SYKSOM treats the n p-dimensional data in a sequential way. That means, it takes the first data interval, assigns it to a cluster, takes the second data interval, assigns it to a cluster, etc. When all n data intervals are assigned to their clusters SYKSOM checks, if a prespecified precision of computing ("Precision threshold") is attained. If this is not yet the case, the module SYKSOM starts a new cycle in which all n data intervals are processed again in the previous order: it treats the first data interval like the (n+1)-th data, the second data like the (n+2)-th data, etc. To prevent an endless loop SYKSOM stops after at most a maximal number of cycles which is determined by this parameter. Increasing this parameter can result in a linear increase in running time.

13.3.5 Precision threshold

In each step of the clustering method the cluster prototypes (centers of the clusters) are updated and changed. The module SYKSOM measures the relative overall change of prototypes after integrating each data interval and stops at the end of the cycle if this overall change is less than a precision threshold. This parameter specifies the precision threshold to be used. Decreasing this parameter can effect an increase in running time. Notice: When the method stops, it is not sure that the desired precision is reached since the selected maximum number of cycles has been attained before (see also "Maximal number of cycles"!).

13.3.6 Eps

After assigning a data interval to a cluster the center of that cluster will be updated and also the centers of all neighbouring (neighbouring in the lattice!) clusters. The parameter eps determines the size of this lattice neighbourhood. That means: if eps is 1, only the centers of the direct neighbours in the lattice will be changed; if eps is 2, the clusters that are neighbours of the direct neighbours (in the lattice) will be involved as well, etc.

13.3.7 Kernel

Assigning a data interval to its cluster has some influence on the centers of neighbouring clusters. How big this influence is, is determined by a kernel function. SYKSOM offers four different kernel functions ("Threshold with standard distance", "Threshold with Euclidean distance", "Gaussian", "Exponential").

13.3.8 Distance

In case of choosing the "stochastic approximation" method three different distance measures between p-dimensional intervals (hypercubes) can be selected ("Vertex-type distance", "Hausdorff-L1 distance", "Hausdorff-L2 distance").

In case of choosing the methods "Mac Queen 1" or "Mac Queen 2" the "Vertex-type distance" is stipulated.

13.3.9 Learning factor

If choosing the "stochastic approximation" method SYKSOM offers two different learning factors ("Classical", "Cyclic repetition").

In case of choosing the methods “Mac Queen 1” or “Mac Queen 2” no learning factor is to be selected (since it is automatically determined by the method).

13.3.10 Cooling

SYKSOM offers three different options of cooling (“No cooling”, “Single-step-cooling”, “Batch-wise-cooling”). The second and the third option use the parameters “**Tmin**” and “**Tmax**”. If one of both cooling methods is selected, the relevant neigbourhoods of the clusters (vertices) are larger in the beginning of the procedure than afterwards, they will shrink in the course of the algorithm. This shrinkage is controlled by a parameter ‘temperature’ T which decreases from its initial value Tmax to Tmin afterwards.

The difference between “single-step” and “batch-wise” resides in the fact that “batch-wise” considers, in which cycle (see also “Number of cycles”!) of the clustering method the data is treated.

13.3.11 Initial class prototypes

As every clustering method of SYKSOM assigns each data interval to that class, which has the “most fitting” prototype, initial class prototypes are needed to start the clustering method. The module offers two different options (“First m data”, “Random single-value prototypes”).

If the first option is selected, the method will initially create m clusters, each consisting of one of the first m data. Then the clustering process will start the assignment of the (m+1)-st data and then proceed with the other ones.

If the second option is selected, the method will initially create m empty clusters, whose prototypes are m random points in p-dimensional space (i.e., intervals of the form [x,x]).

13.3.12 Type of class prototypes

In case of choosing the “stochastic approximation” method SYKSOM offers three different ways for characterizing a class by a class prototype (“Average vertices”, “Envelope-type rectangle”, “Truncated envelope-type rectangle”). If the third option is selected, the covering percentage “**Alpha**” (percentage of cluster members fully contained in the prototype interval) can be chosen.

In case of choosing the methods “Mac Queen 1” or “Mac Queen 2” the class prototype “Average vertices” is selected automatically.

13.3.13 Alpha

See “Type of class prototypes”!

13.3.14 Center replacement

If this option is selected, the clustering method will replace, after each cycle, each class center (obtained by the update method) by the prototype of this class.

13.3.15 Tmin

See “Cooling”!

13.3.16 Tmax

See “Cooling”!

13.4 Additional buttons

13.4.1 Preferences

The user is able to save his own default values for the parameters (see also “Save”!). He gets these parameters by pushing this button.

13.4.2 Default

If this button is pushed, all parameters will be set to their default values.

13.4.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

13.4.4 Save partition base …

With this option, the partition of n data intervals into m clusters (produced after running the SYKSOM module) is stored in a SODAS file. The user is able to change the name of that file.

13.4.5 Save prototype base …

With this option, the prototypes of all m clusters produced after running the SYKSOM module are stored in a SODAS file. The user is able to change the name of that file.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

CLINT Help Guide

Interpretation of Clusters

Edited by FEP

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **6/06/2003**

14 CLINT : Interpretation of Clusters

14.1 Introduction

The module CLINT is used to obtain an interpretation of classes (subsets of the dataset). The classes are identified by categorical single or categorical multi-valued variables: a class is formed by the set of individuals (symbolic objects) sharing a given category of the defining variable. Classes are interpreted on the basis of a given set of variables, by descriptions in the form of symbolic objects and by numerical indicators related to variable contributions.

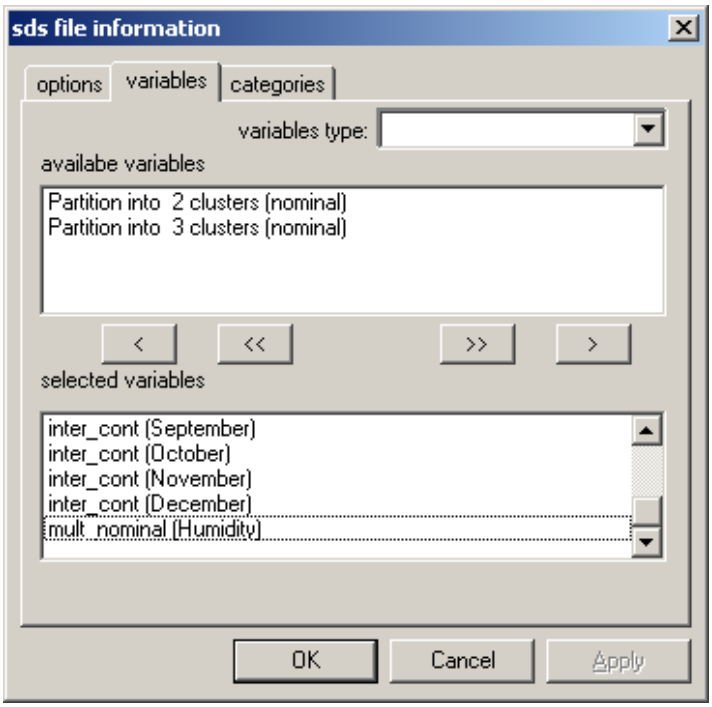
14.2 The input to the CLINT module

The input for the module CLINT is a symbolic data matrix, where individuals, associated to symbolic objects, are described by quantitative single, interval, categorical single, categorical multi-valued and /or modal variables (mixed types are allowed). There must be at least one categorical single or categorical multi-valued variable, which will be used to define the classes to interpret.

Taxonomies and hierarchical dependencies defined on variables may be taken into account.

14.3 The “Variables” window

In this window, the user chooses the variables that will be used to interpret the classes. CLINT accepts all types of variables. The appearance and the operation of the window “variables” is the same as in all other ASSO modules.

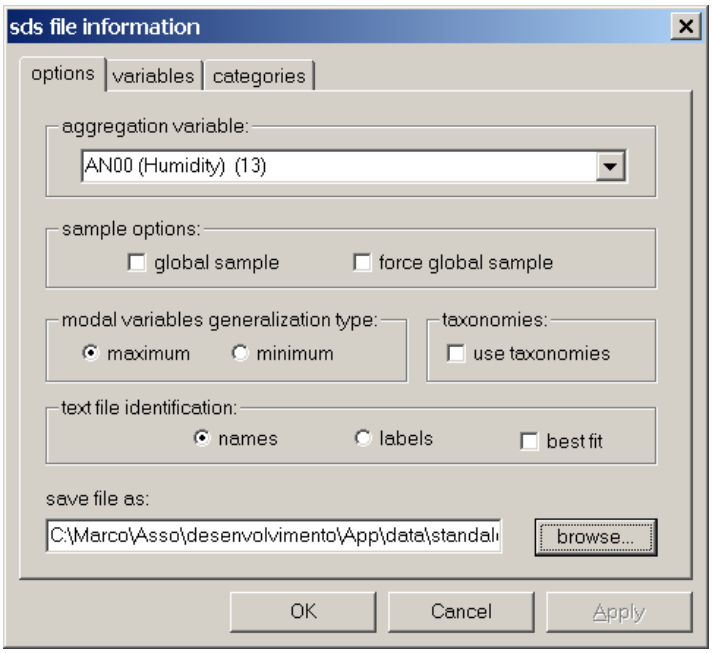


14.4 The Options of CLINT

CLINT provides several different options for treating data, which mainly differ in the way generalization is performed. Therefore, the user must specify the values of a variety of parameters which characterize special options.

On the other hand, it is also possible to run CLINT in a semi-automatic way, i.e., without caring too much about the correct choice of all parameters. In fact, CLINT provides default (standard) values for all parameters.

The options window of CLINT looks as follows:



The several options are explained below in the same order as they appear in the window. For a precise explanation of the methods please look at the specification of this module.

14.5 Aggregation Variable

Here the user must select the variable with defines the classes to be interpreted. A class is formed by the set of individuals (symbolic objects) sharing a given category of this variable. Only categorical single or categorical multi-valued variables are displayed, in brackets the number of categories of each variable is shown. To interpret clusters of a previously obtained partition, hierarchy or pyramid, the corresponding variable must be chosen. Partitions are represented by categorical single variables, hierarchies and pyramids ar represented by categorical multi-valued variables. By default, the first categorical single or categorical multi-valued variable in the data matrix is chosen.

14.6 Sample Options

14.6.1 "Global Sample"

Here the user may enter the information whether the dataset covers the total variable range, for all variables; if this is the case, this option should be checked. By default, this option is not checked.

14.6.2 "Force Global Sample"

This option should be checked if the user wishes that the generality be computed by reference to the observed range of the variables (as opposed to compute it by reference to the declared range). If the "Global Sample" option is checked, then this option is disabled. By default, this option is not checked.

14.6.3 Modal Variables Generalization Type

If Modal Variables are present, the user may choose whether to perform generalization by the **Maximum** or by the **Minimum**. Only one of these options may be chosen; by default, generalization is performed by the Maximum.

14.6.4 Taxonomies

If taxonomies are defined for some variables, CLINT allows the user to use them in the generalization process. If this option is checked taxonomies are used, otherwise they are not. By default, taxonomies are not used.

14.6.5 Text File Identification

Here the user must choose whether to use names or labels in the text output file. A "best fit" option is also available. By default, names are used.

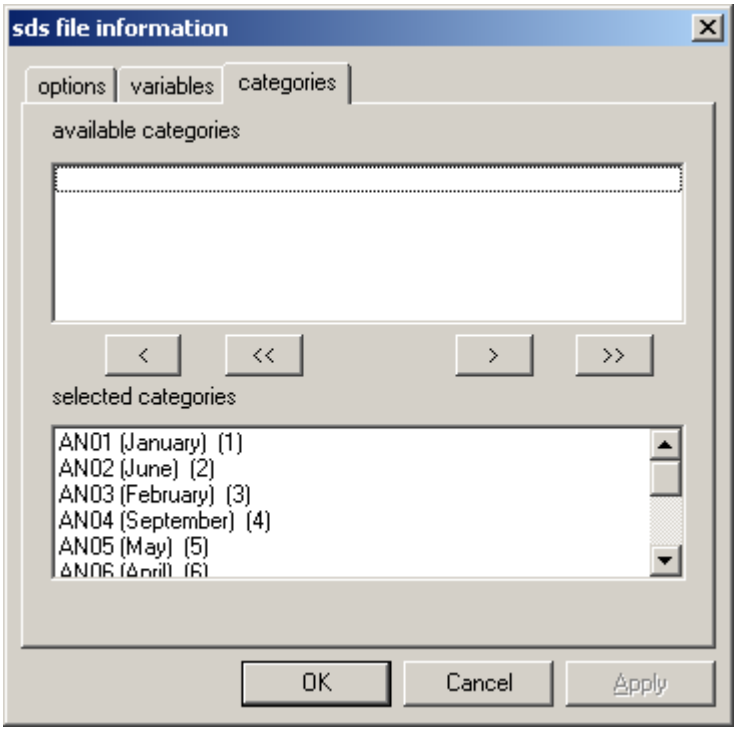
14.6.6 Save file as…

With this option, the description of the classes by symbolic objects and the contribution values are stored, together with the initial data, in a SODAS file. Extra lines and columns have been added to the initial data file. One line is added for each class interpreted by CLINT, containing its generalized description on the basis of the variables. Three extra columns are added containing contribution values: variable to class contributions (modal variable), class to variable contribution (modal variable) and class contribution to global generality (quantitative single variable). By default, the file is called {<chaining directory>clint.xml}. The user is able to change the name and location of that file.

14.7 The "Categories" window

In this window, the user must select the categories corresponding to the classes he wishes to interpret. All categories of the "Aggregation Variable" chosen in the previous window are displayed. This means that the "Aggregation Variable" must be chosen prior to the categories

selection. By default, all categories are select, which means that all classes defined by this variable will be interpreted.



INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SCLASS Help Guide

Unsupervised Classification Tree

Edited by FUNDPMa

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **02/10/2003**

15 SCLASS : Unsupervised Classification Tree

15.1 Introduction

The module SCLASS is a divisive clustering method.

By definition of a divisive clustering method, the algorithm starts with all objects in one large cluster, and splits successively each cluster into (two) smaller ones until a suitable stopping rule prevents further divisions.

This algorithm proceeds in a monothetic way. In other words, it assumes as input *n* data vectors and proceeds such as each split is carried out by using only one single variable (which is selected optimally).

The proposed method is a clustering tree method where nodes are split recursively by choosing the best variable. The original contribution of this method lies in the way of splitting a node. Indeed, the cut will be based on the only assumption that the distribution of points can be modeled by non-homogeneous Poisson process where the intensity will be estimated by the kernel method. The cut will be made in order to maximize the likelihood function.

The resulting classification is a *m*-partition, where the created clusters are disjoint and the union of all classes is the whole set of objects.

15.2 The input to the SCLASS module

This algorithm studies the case where a set of Symbolic Objects are described by *p* **interval variables**.

We take as an example the Ichino'oils dataset. The symbolic data matrix is here composed of 8 symbolic objects described by 4 symbolic interval variables. The "BASE" (see FIG1.) is the file ichino.sds.

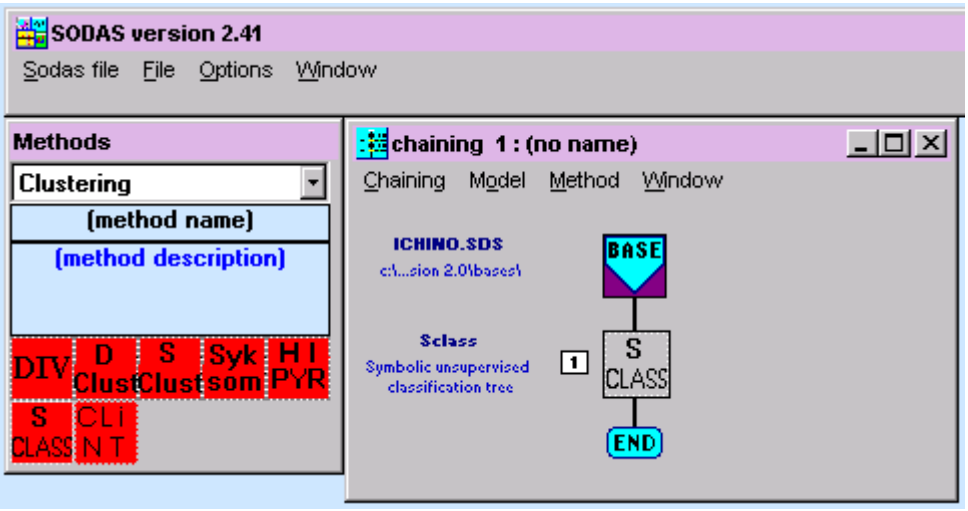


FIG1. Chaining example

15.3 The parameters of SCLASS

The SCLASS method takes also as input a list of variables and some parameters that we will define now.

15.3.1 List of the selected variables

The user must select the symbolic interval variables which will be used inside the SCLASS module.

In the SODAS SCLASS's parameters definition window (see FIG2.), the user has to choose only symbolic interval variables. In the Ichino example, 3 interval variables "specification, freezing, iodine" are selected.

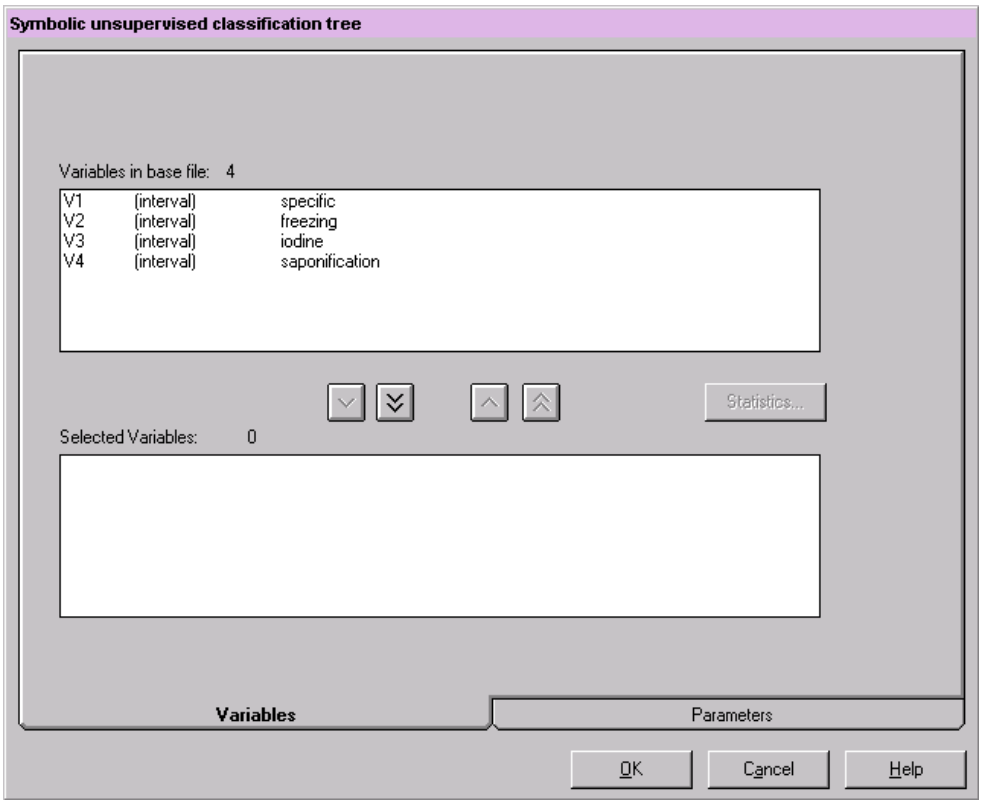


FIG2. The SODAS window for the selection of the variables

15.3.2 The parameter definition

The user must specify the values of parameters which characterize special options. Insofar the user is able to tune the parameters such that the resulting clustering method fits optimally his special application problem.

Two parameters have to be defined :

- **Pruning parameter** : Alpha value for the gap test (value between 0 and 1). Default value is set to 0.5.

- **Minimum size to split the node**: the minimum number of individuals in a node to split this node. Default value is set to 3.

In the Ichino example (see FIG3), we have chosen 0.5 as pruning parameter and we split a node if the number of individuals in a node is greater than 5.

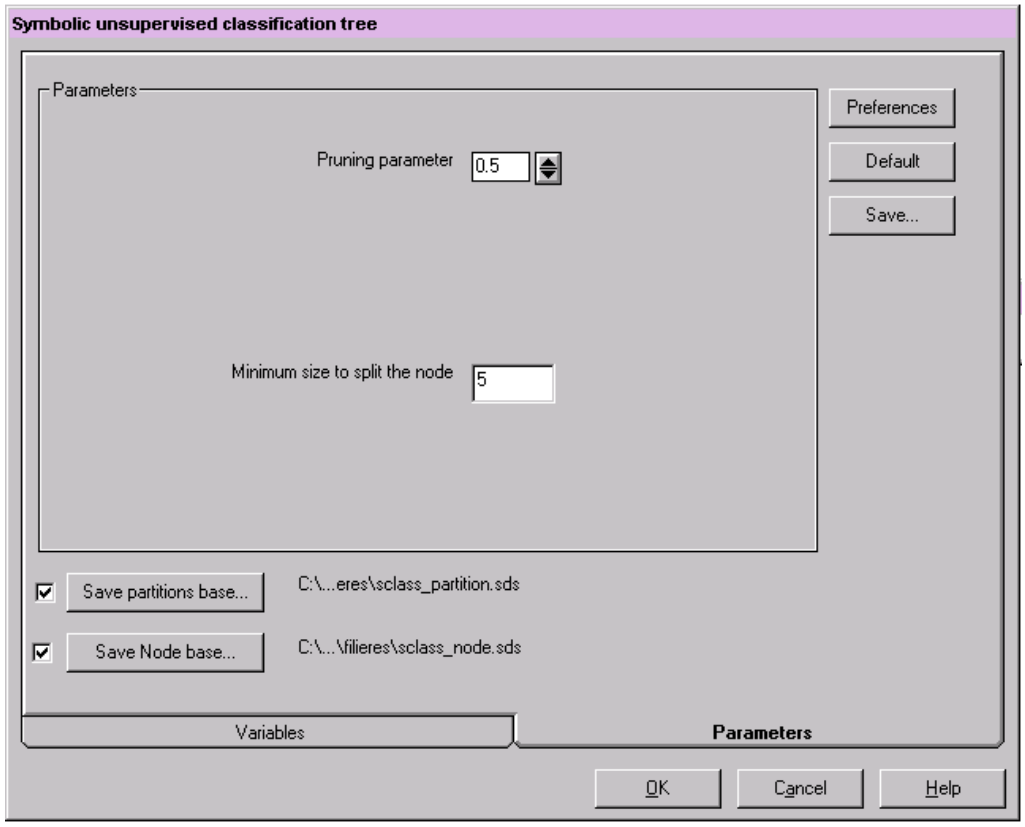


FIG3. The SODAS window for the definition of the parameters

15.4 Additional buttons

15.4.1 Preferences

The user is able to save his own default values for the parameters (see also "Save"!). He gets these parameters by pushing this button.

15.4.2 Default

If this button is pushed, all parameters will be set to their default values.

15.4.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

15.4.4 Save Partition base …

With this option, the partition of n data intervals into m clusters (produced after running the SCLASS module) is stored in a SDS file. Moreover, a new categorical multiple variable is created. This variable enumerates the nodes to which the object belongs. The user is able to change the name of that file.

15.4.5 Save Node base …

In this SODAS file, each node is represented by a symbolic object (produced after running the SCLASS module). The user is able to change the name of that file.

15.5 The output

After having defined the parameters and run the SCLASS method, a listing and a graphical representation of the tree are given as output (see FIG4.).

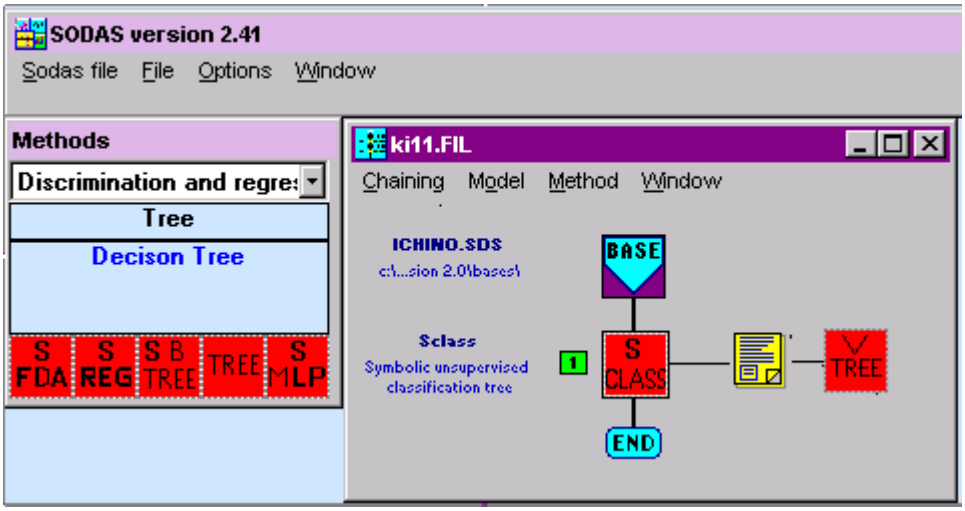


FIG3. The chaining after running the SCLASS method

The listing contains :

- The tree growing strategy :
  For each split of a node, a list of the objects contained in each cluster and the criteria of cut, …

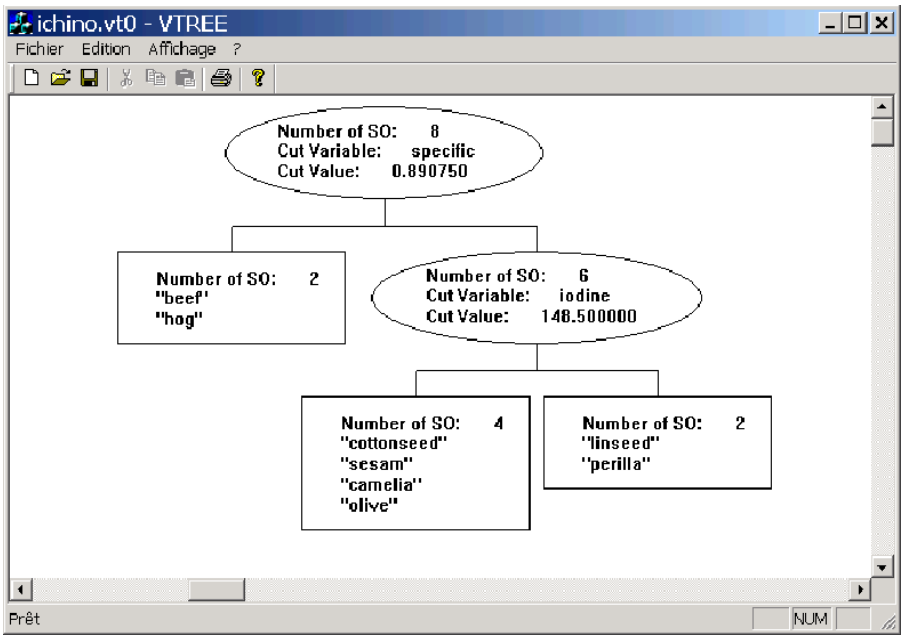
For instance, the listing obtained with ichino example is the following :

```
=========================
Split of the node :    1
=========================

Number of Symbolic objects in the node:      8pt
-------------------------------------
```

```
  Criteria of cut :
  -----------------

 Cut variable :(  1) specific
 Cut value :   0.89
 Smoothing parameter CENTER :   0.02
 Smoothing parameter LENGTH :   0.00
 Rule : if value of i<0.89 -> the SO i is in the left node (next even node)
        if value of i>0.89 -> the SO i is in the right node (next odd node)

Node :   2    Cardinal :     2pt
====================================
 (6)  beef
 (7)  hog

Node :   3    Cardinal :     6pt
====================================
 (0)  linseed
 (1)  perilla
 (2)  cottonseed
 (3)  sesam
 (4)  camelia
 (5)  olive


  =========================
  Split of the node :    2
  =========================

  Number of Symbolic objects in the node:      2pt
  ---------------------------------------

     TERMINAL NODE

 the stop-splitting is true : the size of the node is too small
 value of the stop-splitting rule:     5


  =========================
  Split of the node :    3
  =========================

  Number of Symbolic objects in the node:      6pt
  ---------------------------------------


   Criteria of cut :
  -----------------
```

```
 Cut variable :(  3) iodine
 Cut value : 148.50
 Smoothing parameter CENTER :  36.10
 Smoothing parameter LENGTH :   3.40
 Rule : if value of i<148.5 -> the SO i is in the left node(next even node)
        if value of i>148.5 -> the SO i is in the right node(next odd node)

Node :   4    Cardinal :     4pt
=====================================
 (2)  cottonseed
 (3)  sesam
 (4)  camelia
 (5)  olive



Node :   5    Cardinal :     2pt
=====================================
 (0)  linseed
 (1)  perilla


  =========================
  Split of the node :    4
  =========================

  Number of Symbolic objects in the node:      4pt
  --------------------------------------

     TERMINAL NODE

 the stop-splitting is true : the size of the node is too small
 value of the stop-splitting rule:     5


  =========================
  Split of the node :    5
  =========================

  Number of Symbolic objects in the node:      2pt
  --------------------------------------

     TERMINAL NODE

 the stop-splitting is true : the size of the node is too small
 value of the stop-splitting rule:     5
```

This divisive algorithm yields the 3-cluster partition represented in the following tree (by VTREE) :



You can read the previous tree as follows:

- The first division has been performed according to the variable “specific gravity ” and to the cut value 0.89. The first binary question is [specific ≤ 0.89]? 2 objects (beef and hog) have answered “yes” to this question. They correspond to the 2 objects of the cluster 1. The 6 other objects (which will be divided after) have answered “no” to this question.
- The second division has been performed according to the binary question [Iodine Value ≤ 148.5]?and this question has induced the two classes (cluster2 and cluster3).

So, each cluster corresponds to a symbolic object, e.g. a query assertion:

Cluster1 = [Specific Gravity (x) ≤ 0.89075]

Cluster2 = [Specific Gravity (x) > 0.89075] and [Iodine Value (x) ≤ 148.5]

Cluster3 = [Iodine Value (x) > 148.5].

Then, the resulting 3-cluster partition is:

Cluster1 = {beef, hog},

Cluster2 = {cottonseed, sesam, camelia, olive},

Cluster3 = {linseed, perilla}.

For instance, the object "linseed" will be in the Cluster3 because

- Specific Gravity (linseed) = [0.930;0.935] and (0.930+0.935)/2>0.89075,
- Iodine Value (linseed) = [170;204] and (170+204)/2>148.5.

Factorial

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SPCA HELP GUIDE

Principal Component Analysis

N. C. Lauro, R. Verde, A. Irpino, M. Guerra
DMS

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **24/06/2003**

16 SPCA : Principal Component Analysis

16.1 Introduction

The SPCA is an extensions of the PCA to the case of symbolic data. SPCA aims to represent, in a space of reduced dimensions, statistical units described by interval data, that are represented by hyper cubes, pointing out differences and similarities according to their structural features. This procedure allows to use three different strategy to perform SPCA:

Vertices Principal Component Analysis (V_PCA),

Range Transformation SPCA

Mixed Strategy.

16.1.1 Vertices PCA

The Vertices Principal Component Analysis(V_PCA) strategy is an improvement of the V_PCA proposed by Cazes et. Al (1997). The new approach aims to evaluate symbolic objects with respect to their positioning in the space of the recoded descriptors, introducing a suitable cohesion constraint among all the vertices of each symbolic objects in the analysis, so that the criterion optimized in the analysis is to maximize the variance among the symbolic objects.

16.1.2 Range Transformation PCA

In order to take into account the symbolic objects structural elements, it is proposed a **Range Transformation** $\overline{V}_{ij} = [\overline{y_{i,j}} - \underline{y_{i,j}}]$ of the objects that reveals useful information in studying their size and shape (**RT_PCA**). This approach consists in translating the minimum vertices of the hyper cubes associated to the symbolic objects in the origin of the original space of representation. Then, SPCA is performed on the maximum vertices of the hyper cubes.

16.1.3 Mixed Strategy

The **Mixed Strategy** combining the SPCA and the RT_PCA, in order to improve symbolic objects representation taking into account their differences in terms of scale and structural (size and shape) characteristics. This approach is performed by three steps:

perform RT_PCA in order to extract the principal axis that better represent the size and the shape of symbolic objects;

project the original vertices on the space of the objects, that permits to take into account the symbolic objects vertices cohesion;

perform a PCA on the projections of the vertices previously transformed on the principal axis deriving from the RT_PCA, in order to stress the size and shape information extracted in the previous analysis.

On the factorial plane, the graphical representation for all approaches is performed by rectangles whose vertices represents the minimum and the maximum coordinates of objects.

16.2 The input data to the SPCA.

SO-PCA is performed on a set of n SO's, described by p descriptors, all continuous at interval.

Moreover, logical relations among the descriptors, as well as missing values (NULL) are admitted. In the window “Variables”, SPCA accept only interval-type and single-valued variables**.**

16.3 The output data of the SPCA

As graphical results, SPCA returns:

1) coordinates of the vertices of the hypercubes and the coordinates of the extremes of the intervals of the rectangles which reproduce the hypercube projections on each factorial axis. The latter table is available in a file “.sds” to use as input for the VPLOT module for the graphical representation.

2) the convex hull representation of objects on the base of the two axes chosen in parameter windows

3) the circle of correlation for the representation of the single value correlations of the variables with respect to the factorial axes

4) the circle of interval correlation where the minimum and the maximum correlation of each descriptor with respect each factorial axis is represented

The other main results of SPCA are: the absolute value and the percentage of inertia explained by each factorial axis; some indices to evaluate the quality of the representation of the symbolic objects on the plane; the contribution of each object and of each descriptor to the achievement of the several factorial axes. Furthermore, it is possible to save in output the symbolic variables “point correlations” and the symbolic variables interval correlations in a SOM file to use both as input for the VPLOT module.

16.4 Parameters of SPCA

The SPCA module provide several options to perform the analysis. The user can specify the values of several different parameters. However, if they are not chosen by the user, SPCA provide default values for these parameters, except for the variables selection that is mandatory.

The SPCA module set out the following three windows, which allows to select options and parameters for the analysis: Symbolic Object”, “Variables” and “Parameters”.

16.4.1 “Symbolic Objects” options

The window “Symbolic object” set a drop list that allows to select symbolic objects as Active or Illustrative in the analysis. As default all objects are selected as Active.

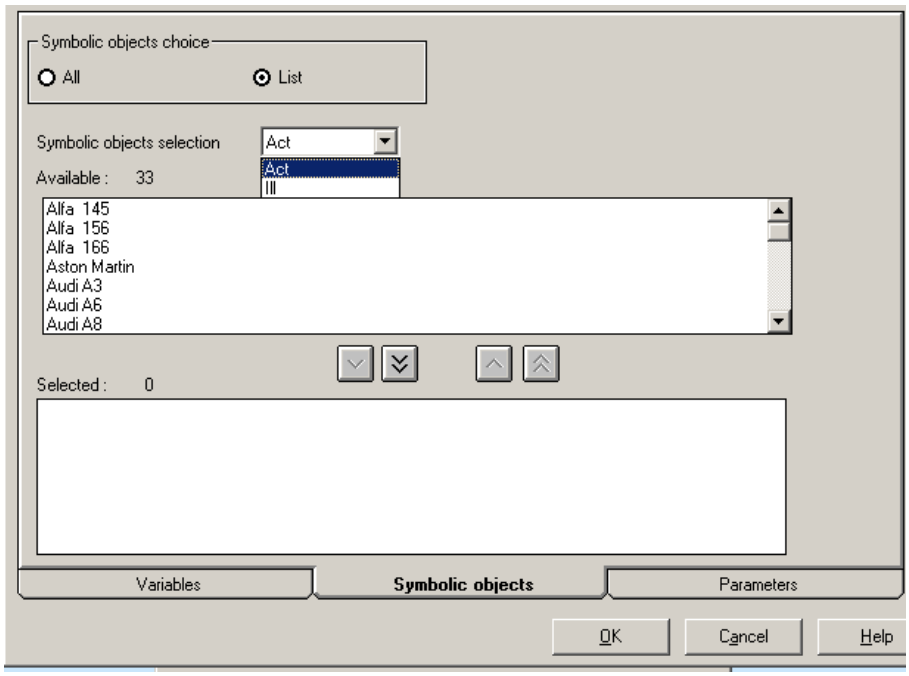


Figure 33 : Symbolic Objects selection window

16.4.2 "Variables" options

The window "Variables" allows to specify the Active and Illustrative variables to use in the determination of the factorial axes. The selection is mandatory, in this case it is not possible to opt for the default value.

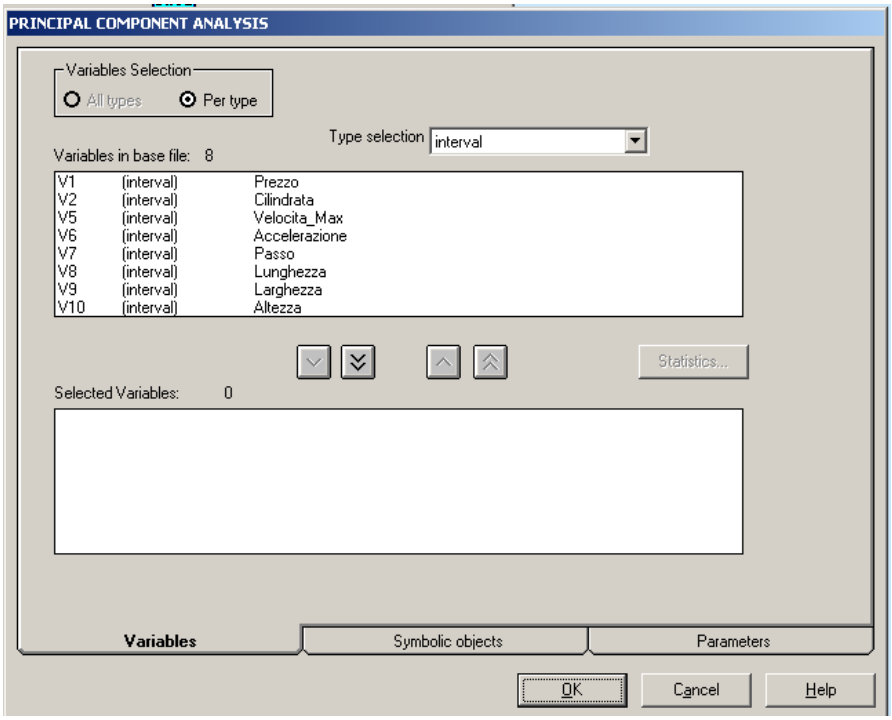


Figure 34: Variables selection window

16.4.3 "Parameters" options

In this display it's possible to choose all the parameters needed for the method. The several options and parameters are explained below.

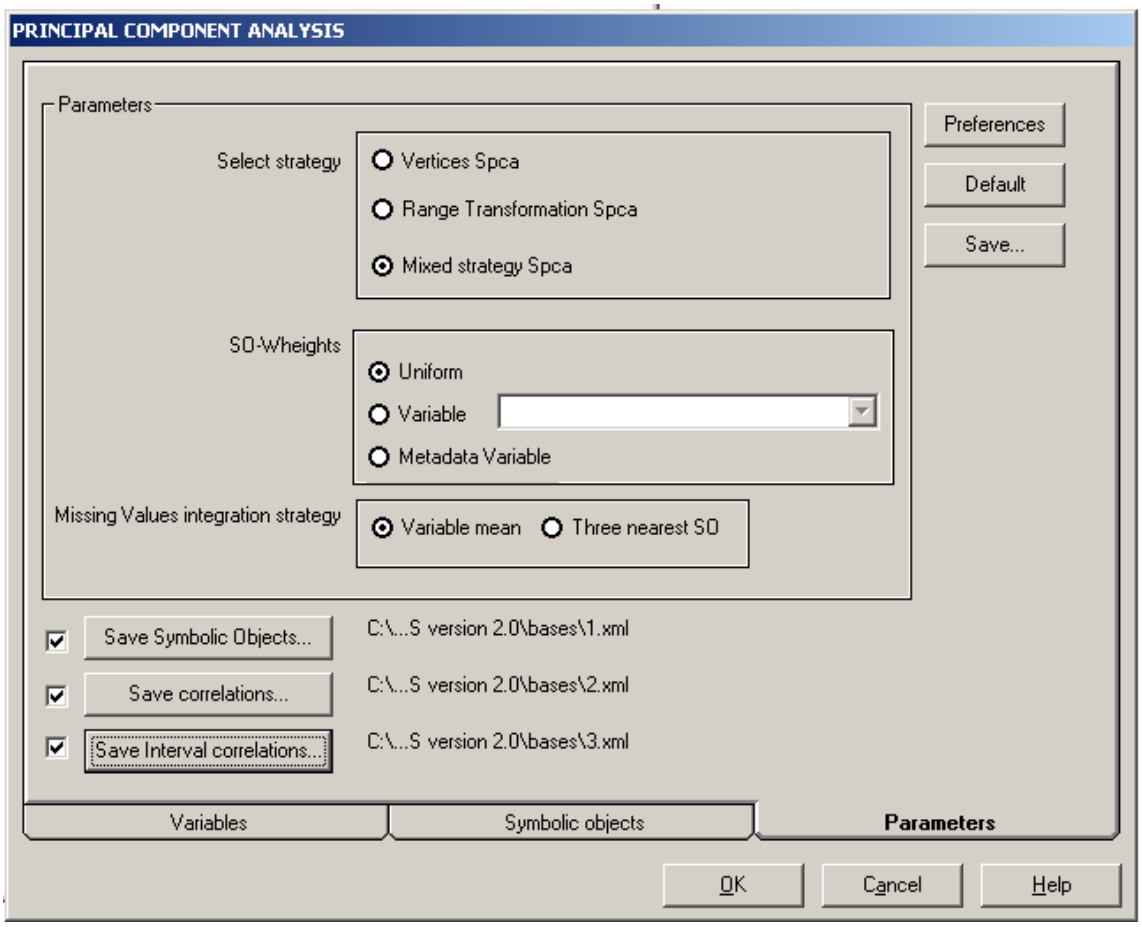


Figure 35: Parameters selection window

16.4.4 Strategy

In the box **"Select Strategy"** it is possible to choose one of the three different approaches to perform a Principal Component Analysis:

- ✓ **"Vertices SPCA" :** The Vertices Principal Component Analysis (V_PCA) strategy is an improvement of the V_PCA proposed by Cazes et. Al (1997). The new approach aims to evaluate symbolic objects with respect to their positioning in the space of the recoded descriptors, introducing a suitable cohesion constraint among all the vertices of each symbolic object in the analysis, so that the criterion optimized in the analysis is to maximize the variance among the symbolic objects.
- ✓ **"Range Transformation PCA"** : In order to take into account the symbolic objects structural elements, it is proposed a Range Transformation of the objects that reveals useful information in studying their size and shape (RT_PCA). This approach consists in translating the minimum vertices of the hyper cubes associated to the symbolic objects in the origin of the original space of representation. Then, SPCA is performed on the maximum vertices of the hyper cubes.

- ✓ **"Mixed Strategy PCA"**: The Mixed Strategy combining the SPCA and the RT_PCA, in order to improve symbolic objects representation taking into account their differences in terms of scale and structural (size and shape) characteristics. This approach is performed by three steps:

  • perform RT_PCA in order to extract the principal axis that better represent the size and the shape of symbolic objects;

  • project the original vertices on the space of the objects, that permits to take into account the symbolic objects vertices cohesion;

  • perform a PCA on the projections of the vertices previously transformed on the principal axis deriving from the RT_PCA, in order to stress the size and shape information extracted in the previous analysis.

  On the factorial plane, the graphical representation for all approaches is performed by rectangles whose vertices represent the minimum and the maximum coordinates of objects.

16.4.5 SO - Weights

In the **"SO-Weights"** box it is possible to assign a weights system to symbolic objects in order to take into account the different importance of the objects in the analysis.
The alternatives for this option are:

- **Uniform weights**; all objects has the same weight
- **Variable**; the variable that represent weights needs to be a "Quantitative single value" variable. The drop list contains only the "Real" single value variables present in the file.
- **Metadata**; if a metadata it is present in the input file, it can be used as weights.

The default value for this option is the Uniform weight.

16.4.6 Missing Value Integration Strategy

The "Missing Value Integration Strategy" box allows to choose a strategy to integrate missing value in symbolic data matrix.

The "Variable Mean" strategy replace missing value with an interval which the minimum is the mean of the minima of the variable for all objects, and the maximum is the mean of the maxima of the variable for all objects.

The "Three nearest SO" strategy replace missing value with the mean of the three more similar objects.

16.4.7 Axes for Convex hulls representation

Two box are present in the parameter window in order to choice the two axes for the representation of the 2D convex hulls of symbolic objects.

16.5 Additional buttons

16.5.1 Preferences

The user is able to save default values for the parameters. He gets these parameters by pushing this button.

16.5.2 Default

If this button is pushed, all parameters will be set to their default values.

16.5.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

16.5.4 Save Symbolic Objects …

This option allows users to choose an alternate name and a path to save the symbolic objects factorial coordinates. These coordinate are stored in a SODAS file.

16.5.5 Save Correlation …

This option allows users to choose an alternate name and a path to save the symbolic variables point correlations. These point correlations are stored in a SODAS file.

16.5.6 Save Interval Correlation …

This option allows users to choose an alternate name and a path to save the symbolic variables interval correlations. These interval correlations are stored in a SODAS file.

References

Bock, H. H., Diday, E. (eds) (2000), Analysis of Symbolic Data, Springer-Verlag. Heidelberg.

Cazes, P., Chouakria, A., Diday, E., Schektman, Y.: 1997, Extension de l'analyse en composantes principales à des données de type intervalle, *Revue de Statistique Appliquée* XIV(3).

D'Ambra, L., Lauro, C. : 1982, Analisi in componenti principali in rapporto a un sottospazio di riferimento*, Rivista di Statistica Applicata* 15(1), 51-67.

Lauro, C., Palumbo, F.: 2000, Factorial Methods with cohesion constraints on Symbolic Objects. In Proceeding of IFCS2000, Springer-Verlag, .

Lauro, C., Palumbo, F.: 2001, Principal Component Analysis of Interval Data: a Symbolic Data Analysis Approach, *Computational Statistics*.

Lebart, L., Morineau, A. and Piron, M.: 1995, Statistique exploratoire multidimensionnelle, Dunod, Paris.

Verde, R.: 1997, Symbolic object decomposition by factorial techniques. Indo-French Meeting, LISE-CEREMADE, Université Paris IX Dauphine.

Verde, R. and De Angelis, P.: 1997, Symbolic objects recognition on a factorial plan, NGUS'97, Bilbao Spain.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SGCA Help guide

Generalised Canonical Analysis

N. C. Lauro, R. Verde, A. Irpino, M. Guerra
DMS

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: 24/06/2003

17 SGCA : Generalised Canonical Analysis

17.1 Introduction

The SGCA is an extension of the Generalised Canonical Analysis to the case of symbolic data. SGCA aims to analyse symbolic objects described by different kind of descriptors. It looks for the more suitable factorial axes in order to study and visualise the relations among symbolic objects and descriptors in a subspace of reduced dimensions. In the analysis it is considered a cohesion constraint in order to preserve the unitary of the symbolic objects in the analysis.

The interpretation of factorial axes in symbolic terms is accomplished with reference to the variables and categories which have maximal contributions. The measures of descriptors contribution to the axes are expressed as the squared correlation variable/factor.

17.2 The input to the SGCA

The input of SGCA is a set of *n* symbolic data characterised by *p* symbolic variables of different type: “quantitative single”, “interval”, “categorical single”, “categorical multi-valued” and “modal”. Missing data and logical rules, as well as hierarchical rules, are admitted. Furthermore, logical dependencies among the descriptors, taxonomies, as well as missing data (NULL), are suitably admitted in the technique.

All numerical variables are fuzzy coded into a categorical variable with three statuses: High, Medium and Low. This means that if we have an interval variable describing, for instance the “Height” of an object, it will be fuzzy coded into a categorical modal variable that assumes three states such as “Height High”, “Height Medium” and “Height Low”, according to a semi linear B-Spline function. No transformations are performed on the nominal, multinominal and modal variables.

Logical dependencies reduce the space of symbolic objects description, so that, we SGCA decomposes the symbolic object's into sub-objects, whose descriptions are consistent with the condition expressed by the logical rules.

17.3 The output data of the SGCA

As results, the SGCA provide the coordinates of the vertices of the hyper cubes and the coordinates of the extremes of the intervals of the rectangles which reproduce the hypercube projections on each factorial axis of the objects and the categories. The tables for the objects and the categories are available in two file (for the symbolic objects coordinates and for the categories coordinates) to use as input for the VPLOT module to achieve the classical graphical visualisation by hyper cubes (the objects) and by points (the categories).

The other main results of SGCA are a table summarizing the absolute value, the percentage and the cumulative inertia explained by each axis, the descriptor and objects absolute contributions tables to evaluate the quality of the representation of the symbolic objects on the plane, and the relative contribution tables of each object and of each descriptor to the achievement of the several factorial axes.

Therefore, it is provided a table containing the descriptions of the factorial axes in terms of contributions of each category to positive (+) or negative (-) versus of factorial axes.

17.4 The options for SGCA

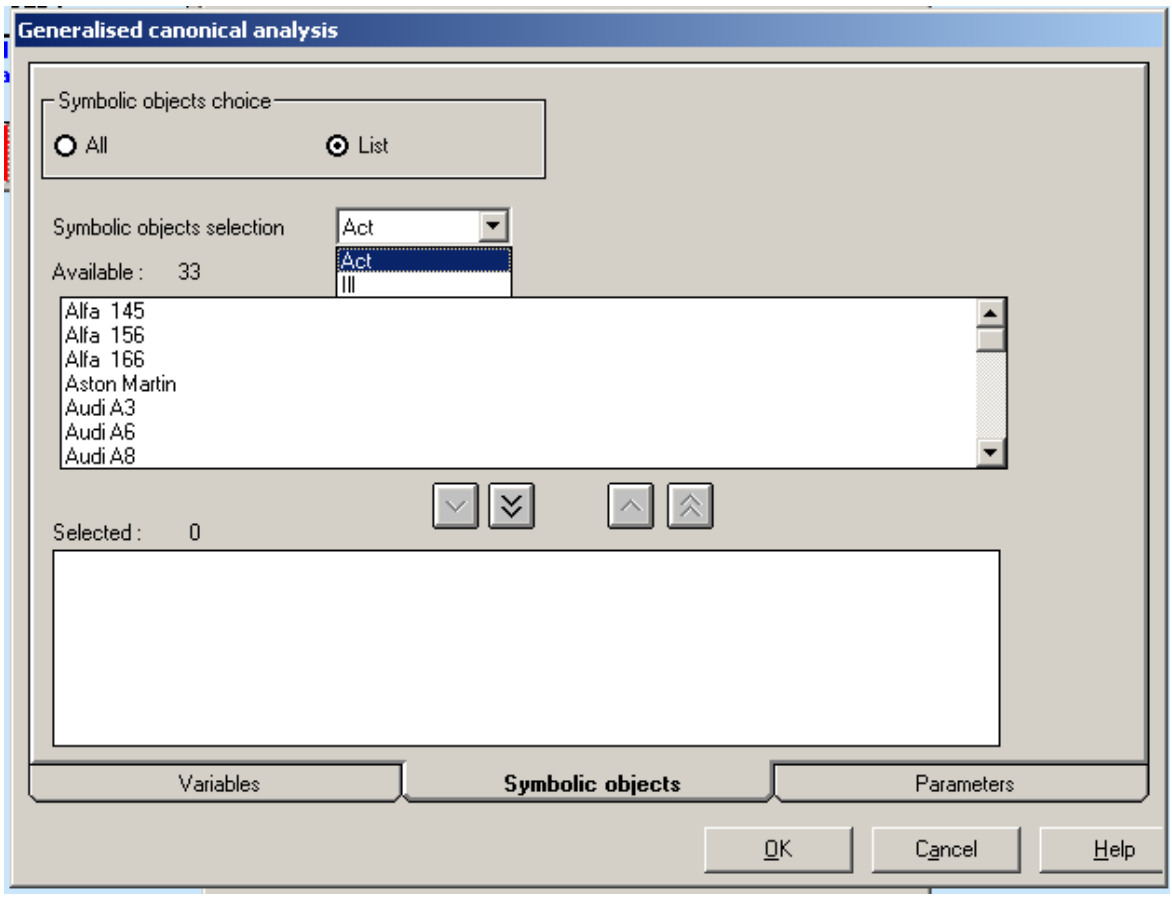
The SGCA module provides several options to perform the analysis. The user can specify the values of several different parameters. However, if they are not chosen by the user, SGCA provide default values for these parameters, except for the variables selection that is mandatory.

The SGCA module sets out the following four windows, which allows to select options and parameters for the analysis: Symbolic Object", "Variables", "Rules" and "Parameters".

17.4.1 "Symbolic Objects" options

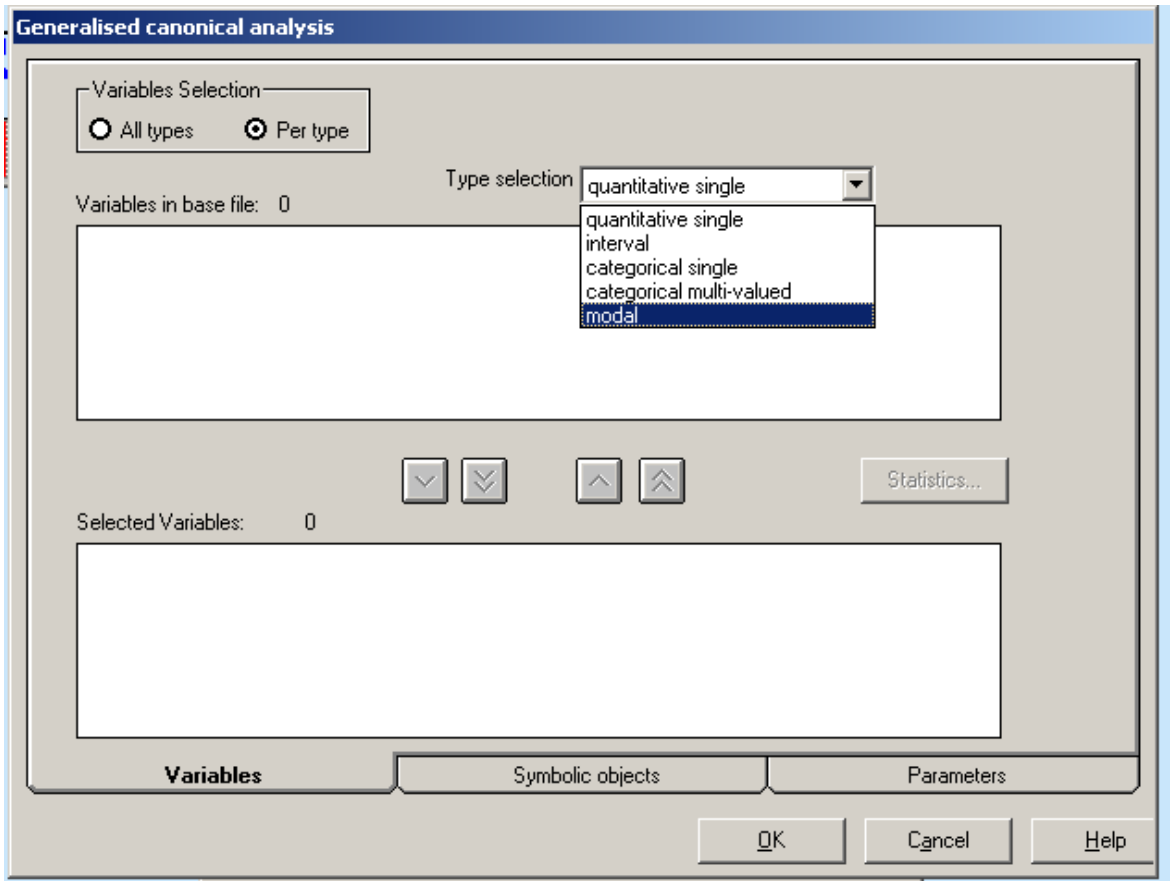
The window "Symbolic objects" sets a drop list that allows to select symbolic objects as Active or Illustrative in the analysis.

As default all objects are selected as Active.



17.4.2 "Variables" options

The window "Variables" allows to specify the Active and Illustrative variables to use in the determination of the factorial axes. The selection is mandatory; in this case it is not possible to opt for the default value.



17.4.3 "Rules" options

The window "Rules" allows user to select rules. The rules represent logical relationships between symbolic descriptors of the objects, and they can be Hierarchical Dependence rules and Logical Dependence rules.

The SGCA module allows user to select a maximum of 5 rules as active, whereas by default no rules are selected.

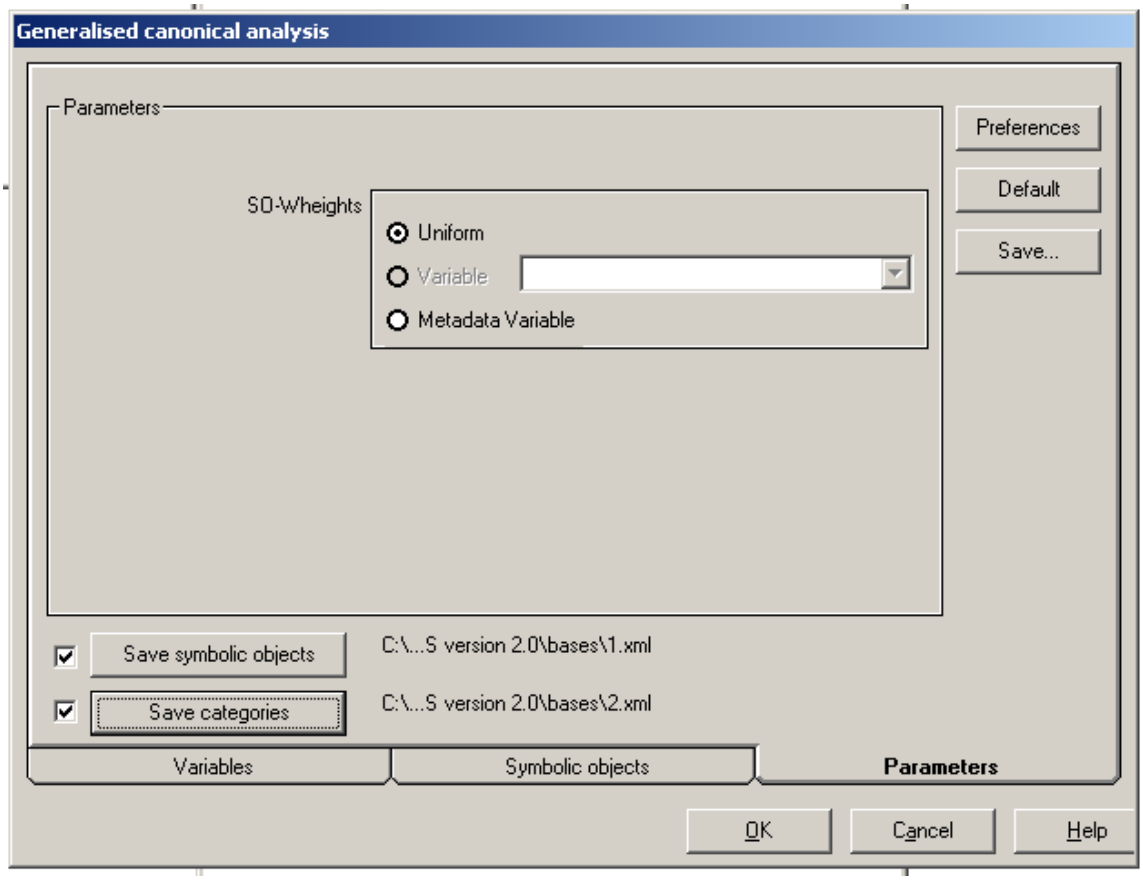
17.4.4 "Parameters" options

The "Parameters" window allows user to specify the weights system to assign to the symbolic objects in order to take into account the different importance of the objects in the analysis

In the **"SO-Weights"** box it is possible to choose one of the three following alternatives weight systems provided by the SGCA:

- **Uniform weights**; for which all objects has the same weight.
- **Variable**; for which is possible to choose as weight a variable in the input file. The variable that represent weights need to be a "quantitative single" variable. The drop list contains only the "quantitative single" variables present in the file.
- **Metadata**; if a metadata it is present in the input file, it can be used as weights.

The default value for this option is the Uniform weight.



17.5 Additional buttons

17.5.1 Preferences

The user is able to save default values for the parameters. He gets these parameters by pushing this button.

17.5.2 Default

If this button is pushed, all parameters will be set to their default values.

17.5.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

17.5.4 Save Symbolic Objects

This option allows users to choose an alternate name and path to save the symbolic objects coordinate on factorial axes. This factorial coordinates are stored in a SODAS file (with ".sds or .xml" extension).

17.5.5 Save Categories

This option allows users to choose an alternate name and a path for the symbolic categories coordinates on factorial plane. The coordinates are stored in a SODAS file (with ".sds or .xml" extension).

Bibliography

Bock, H. H. and Diday, E. (eds): 2000, *Analysis of Symbolic Data*, Springer.

Lauro, C., Palumbo, F.: 2000, Factorial Methods with cohesion constraints on Symbolic Objects. In Proceeding of IFCS2000, Springer-Verlag, .

Lebart, L., Morineau, A. and Piron, M.: 1995, Statistique exploratoire multidimensionelle, Dunod, Paris.

Verde, R.: 1997, Symbolic object decomposition by factorial techniques. Indo-French Meeting, LISE-CEREMADE, Université Paris IX Dauphine.

Verde, R. and De Angelis, P.: 1997, Symbolic objects recognition on a factorial plan, NGUS'97, Bilbao Spain.

Verde, R. : 1999 Generalised Canonical Analysis on Symbolic Objects. In *Classification and Data Analysis, Theory and Application*. Vichi M. - Opitz O. (Eds), Springer-Verlag, Heidelberg, 195-202.

Verde, R. and Lauro, C.: 1993, Non symmetrical data analysis of multiway fuzzy coded matrices, ISI, Florence.

Discrimination and Regression

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

TREE Help guide

Decision Tree

Y. Lechevallier and E. Périnel
INRIA

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **20/05/2003**

18 TREE : Decision Tree

18.1 Introduction

The module Decision Tree proposes an algorithm of tree growing applied to explicitly imprecise data. These are formally described by probabilistic assertions in the framework of symbolic data analysis. In this context, the recursive partition procedure can be viewed as an iterative search for an organized set of symbolic objects which best fits the initial data. At each step, the best split is obtained through the use of a general information measure.

In output, we obtain a new list of symbolic objects that permit to assign future new objects to one class of the known prior partition.

18.2 The input to the TREE module

The input of the TREE method is a classical data matrix or a symbolic data matrix.

We take as an example the symbolic transformation of the waveform recognition data proposed L. Breiman, J.H. Friedman, R.A. Oslhen and C. J. Stone in "Classification and Regression Trees"; Belmont Eds, 1984.

The symbolic data matrix is here composed of 30 symbolic objects described by 21 symbolic interval variables and a classification variable which is a qualitative variable. The “BASE” (see FIG1.) is the file wave30.sds.

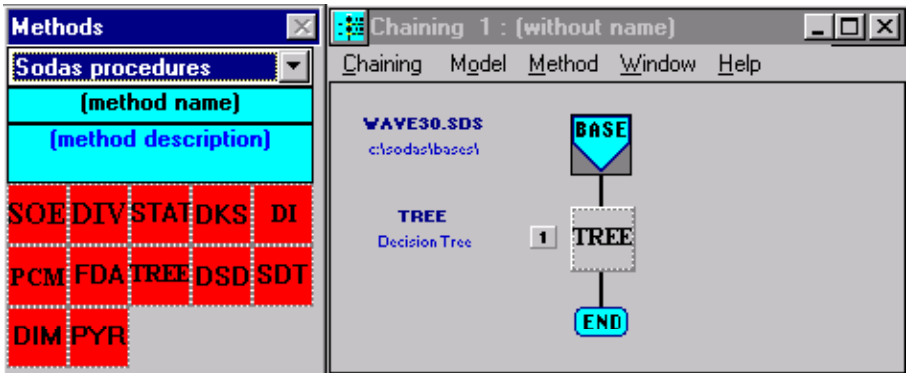


FIG1. Chaining example

18.3 Parameters of TREE

The TREE method takes also as input a list of variables and some parameters we will define now.

18.3.1 List of the selected variables

The user must choose the set of the predictor variables which is:

- a set of quantitative or interval variables,
- or a set of qualitative or categorical multi-valued or modal variables.

The **mixed data table** is not treated in this version..

18.3.2 Parameter details

Seven parameters have to be defined:

- Number of terminal nodes

possible values : 2 to number of OS.

default value : number of OS/2

- Soft Assignment

possible values : Pure or Fuzzy

default value : Fuzzy

If the value is Pure then the a posteriori probabilities of OS are 1 or 0. If the value is Fuzzy then a posteriori probabilities of OS are included in [0,1].

- Splitting criterion

possible values : Gini or Information or Likelihood

default value : Gini for Pure or Likehood for Fuzzy

Remark:

If Soft Assignment is Fuzzy then Splitting criterion is Likelihood

If Soft Assignment is Pure then Splitting criterion is Gini or Information.

- Minimum size to split the node

possible values : from 5 to number of OS

default value : 5

The construction of the decision tree is equivalent to a recursive partitionning of the representation space into subspace. During tree construction, nodes are successively split until the following stopping rule is true for all leafs of the tree. TREE use two stopping rules defined by the parameters ***Minimum size of no-majority*** class and ***Minimum size of right or left descendant nodes***.

If the size of the leaf is smaller than ***Minimum size to split the node*** this leaf is not split.

- Minimum size of no-majority class

possible values : from 2 to (number of OS)/2

default value : 2

If the size of the OS don't belong to the no-majority class is smaller than ***Minimum size of no-majority*** this leaf is not split.

- Minimum size of right or left descendant nodes

possible values : 1 to Minimum size to split the node/2

default value : 1

If the size of right or left descendant nodes is smaller than ***Minimum size of right or left descendant nodes***. then the binary split is not selected.

- **Frequency of objects selected in test set**

possible values :0. to 90.

default value : 0.

If the value is 0. test set is empty.

18.4 Additional buttons

18.4.1 Preferences

The user is able to save his own default values for the parameters (see also “Save”!). He gets these parameters by pushing this button.

18.4.2 Default

If this button is pushed, all parameters will be set to their default values.

18.4.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

18.4.4 Save partition base …

With this option, the partition of n data intervals into m clusters (produced after running the TREE module) is stored in a SODAS file. The user is able to change the name of that file.

18.4.5 Save Node base …

With this option, the symbolic representation of all m nodes produced after running the TREE module are stored in a SODAS file. The user is able to change the name of that file.

18.5 The output

After having defined the parameters and run the TREE method, a listing is given as output (see FIG5.)

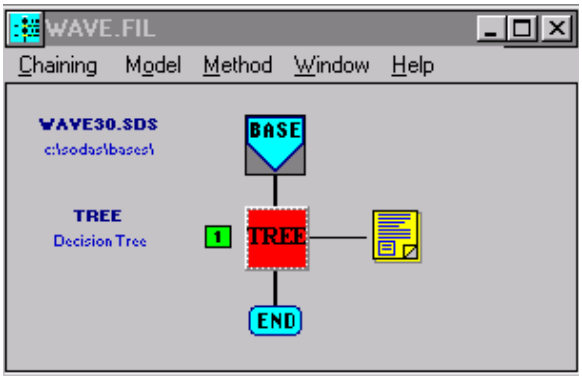


FIG2. The chaining after running the TREE method

- the list of variables used
- the list of SO belonging to a training set.
- the list of SO belong to a test set
- the list of the nodes. Each node is described by the rule.
- the list of terminal nodes. Each terminal node is described by the standard error estimates for training set and test set, the list of SO belong to learning and test sets and the rule for assigning a new SO.

18.6 Waveform example

18.6.1 TREE Windows

In this, in the SODAS TREE's parameters definition window (see FIG 2.), the user has to choose between qualitative and continuous variables and **should not mix them**. In the waveform example, the 21 continuous variables position1 to position21 are selected.

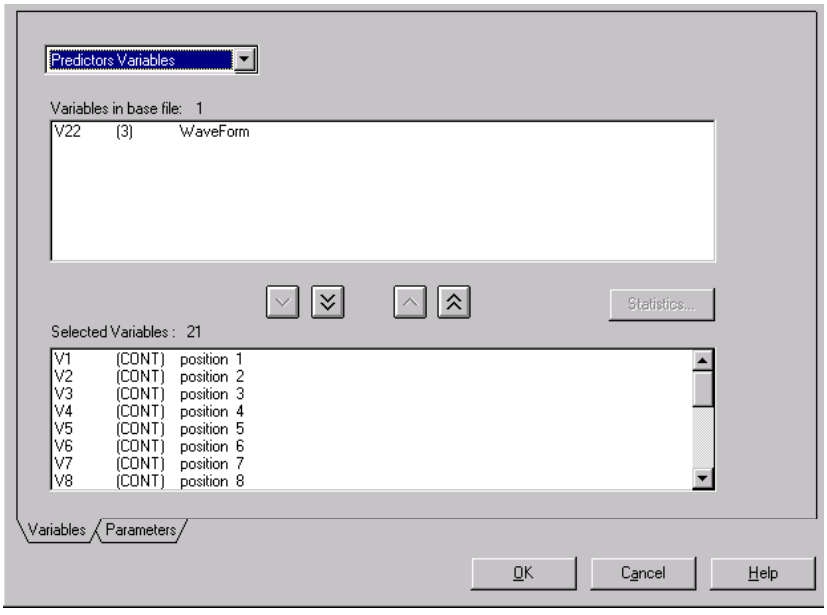


FIG3. The SODAS window for the selection of the predictor variables

After you must selected only one qualitative variable which represents classification variable.

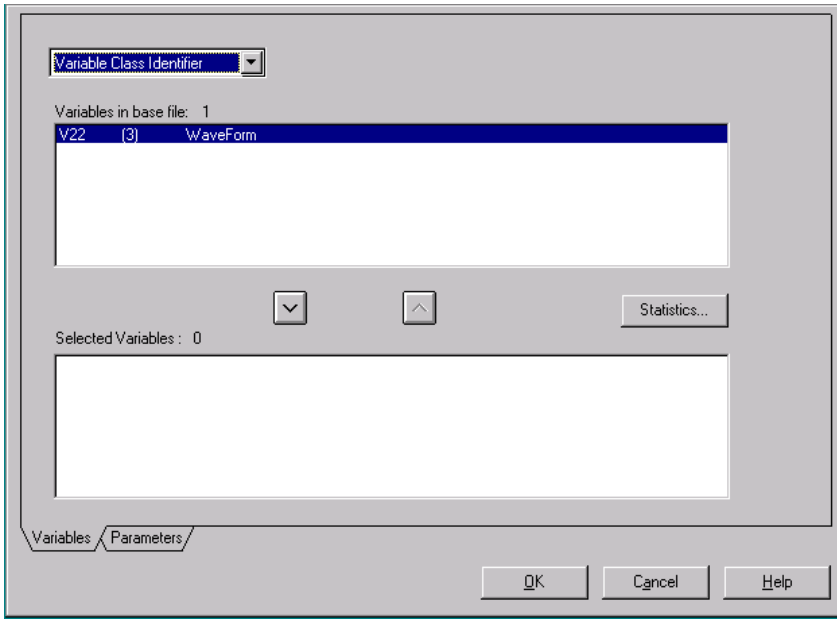


FIG4. The SODAS window for the selection of the classification variable

In the waveform example (see FIG3.) we have chosen a dissimilarity with no normalization, the number of classes is 3 which mean that the DIV method will perform a partition in two classes and a partition in 3 classes, and we have chosen not to create a partition file.

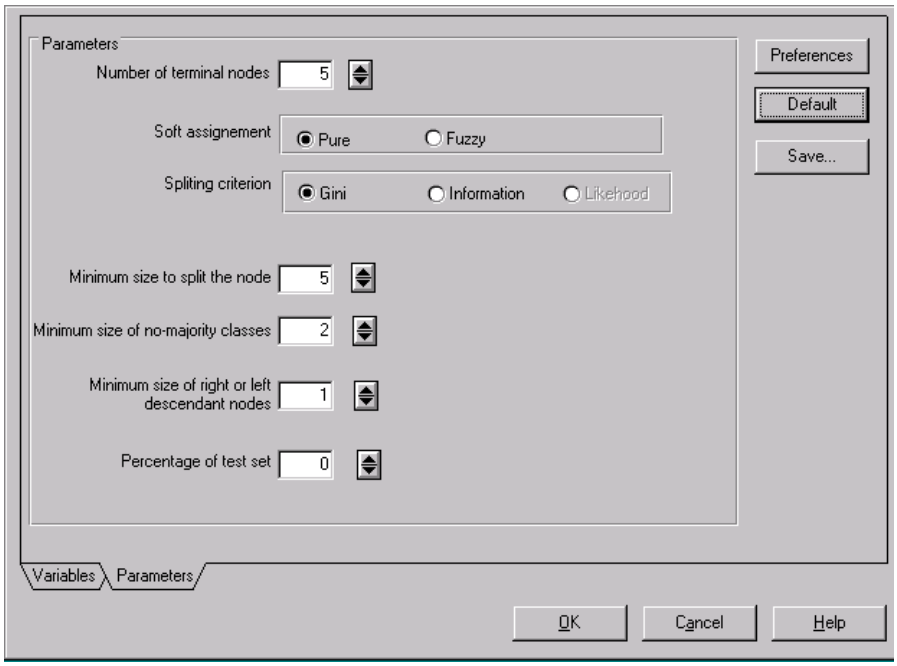


FIG5. The SODAS window for the definition of the parameters

18.6.2 General information

For instance, the end of the listing obtained with the waveform example is the following :

```
NUMBER OF A PRIORI CLASSES :   3
      ID_CLASS     NAME_CLASS
         1           wave_1
         2           wave_2
         3           wave_3

   CLASS  SIZE   LEARNING     TEST

     1     10       10          0
     2     10       10          0
     3     10       10          0

 TOTAL     30       30          0
```

In this case the test set is empty and each a priori class contains 10 symbolic objects. The first table gives the correspondence between the name NAME_CLASS and the number ID_CLASS for each a priori classes.

The aim is to build the set of binary questions that will be used in order to grow the tree. TREE will employ *standard* binary questions.

18.6.3 Splitting step

TREE aims at choosing the best binary split at a given node, among all admissible questions. It relies on the choice of an information measure that permits to calculate the degree of homogeneity of two new nodes induced by a given binary split.

For each node we have :

```
  ================================
  | SPLIT OF A NODE       :     1 |
  ================================

  LEARNING SET

     ========================================================
     |          |  N(k/t)  |   N(k)   |  P(k/t)  |  P(t/k)  |
     ========================================================
     | wave_1   |    10.00 |    10.00 |    33.33 |   100.00 |
     | wave_2   |    10.00 |    10.00 |    33.33 |   100.00 |
     | wave_3   |    10.00 |    10.00 |    33.33 |   100.00 |
     ========================================================
```

N(k/t) represents the number of SO contain in the node t (In this case t=1) the a priori class k.
N(k/t) represents the number of SO contain in the a priori class k .

P(k/t) is the proportion of the SO of node t belonging to a priori class k.
P(t/k) is the proportion of the SO of class k belonging to the node t.

```
TREE CRITERION  0.666667

=========================================================
     |  Ord |   variable          |    value     |    criterion
|

=========================================================
     |    1 |(  13) position 13   |       3.8100 |       0.3333
|
     |    2 |(   5) position  5   |       0.4400 |       0.4167
|
     |    3 |(   9) position  9   |       3.5200 |       0.4167
|
     |    4 |(   7) position  7   |       2.1100 |       0.4284
|
     |    5 |(   8) position  8   |       2.8600 |       0.4444
|
=========================================================
The variables "position 5", "position 9","position 7" and
"position 8" are alternative variables of the variable
"position 13".

 SPLITTING NODE:    1

 VARIABLE      : (  13) position 13
 SPLIT         : 3.810000
 CRITERION     : 0.333333
```

The first division has been performed according to the variable "position 13" and to the cut value 3.81. The first binary question is [position 13 ≤ 3.81] ? The objects in node 2 have answered “yes” to this question. They correspond to the 20 objects (10 belong the a priori class wave_1 and 10 belong wave_2. The 10 other objects have answered “no” to this question and correspond to the objects of the node 3 .

```
       LEARNING SET

      ================================================
      |          | left node  | right node | Row totals |
      |  node    |         2  |       3    |       1    |
      ================================================
      | wave_1   |     10.00  |      0.00  |     10.00  |
      | wave_2   |     10.00  |      0.00  |     10.00  |
      | wave_3   |      0.00  |     10.00  |     10.00  |
      ================================================
      |  Total   |     20.00  |     10.00  |     30.00  |
      ================================================
```

Each node has to be tested in order to decide if it should be or not a terminal node (= a leaf). Classical tests will be used for this phase: a node will be considered terminal if its size is less than a given parameter (fixed by the user), or if it contains objects belonging all to the same class of the prior partition (the node is called *pure*). The node 3 is a terminal node.

```
  ================================
  | SPLIT OF A NODE       :     3 |
  ================================

  LEARNING SET

     ========================================================
     |          |  N(k/t)  |   N(k)   |  P(k/t)  |  P(t/k)  |
     ========================================================
     | wave_1   |     0.00 |    10.00 |     0.00 |     0.00 |
     | wave_2   |     0.00 |    10.00 |     0.00 |     0.00 |
     | wave_3   |    10.00 |    10.00 |   100.00 |   100.00 |
     ========================================================

 THIS STOP-SPLITTING RULE IS TRUE : The size of the no-
majority classes is too small
      SIZE OF THE NO-MAJORITY CLASSES 0.000000  VALUE OF STOP-
SPLITTING RULE 2.000000
  THIS NODE IS A TERMINAL NODE
```

We have the list of objects assigned to the terminal node 3.

```
 List of objects :

     (3)"wave_3_0"  (3)"wave_3_6"  (3)"wave_3_1"  (3)"wave_3_7"
     (3)"wave_3_3"  (3)"wave_3_4"  (3)"wave_3_5"  (3)"wave_3_2"
     (3)"wave_3_8"  (3)"wave_3_9"
```

18.6.4 More details on each leaf or terminal node

Each leaf or terminal node are described by the following tables:

```
 ==============
 LEAF    :    4
 ==============

     ========================================================
     |          |  N(k/t)  |   N(k)   |  P(k/t)  |  P(t/k)  |
     ========================================================
     | wave_1   |    10.00 |    10.00 |    90.91 |   100.00 |
     | wave_2   |     1.00 |    10.00 |     9.09 |    10.00 |
     | wave_3   |     0.00 |    10.00 |     0.00 |     0.00 |
     ========================================================
```

where

N(k/t) is the number of SO contain in the node t (In this case t=4) the a priori class k .

N(k/t) is the number of SO contain in the a priori class k .

P(k/t) is the probability that the SO of the node t belongs to a priori class k.

P(t/k) is the probability that the SO of a priori class k is assigned to the node t.

```
 RULE :
    IF    [ position  9 <= 3.800000] IS TRUE
   AND    [ position 13 <= 3.810000] IS TRUE

      THEN ASSIGN_CLASS IS wave_1
```

If the object verifies this rule then the object is assigned to the class wave_1.

```
 r(t)= 0.090909 p(t)= 0.366667
 R(t)= 0.033333
```

r(t) is the resubstitution estimate of the expected misclassification cost of the node t.

p(t) is the proportion of the objects which are assigned to the node t.

R(t)is equal to p(t)*r(t).

```
 List of objects :

 (1)"wave_1_3"          (1)"wave_1_1"          (1)"wave_1_6"
(1)"wave_1_9"           (1)"wave_1_5"          (1)"wave_1_7"
(1)"wave_1_2"           (1)"wave_1_8"          (1)"wave_1_0"
(1)"wave_1_4"           (2)"wave_2_5"
```

18.6.5 Final result

It contains the main following information's :

- The confusion matrix
- An estimation of the global misclassification probability (and the one associated to each prior class)
- The list of objects inside each node of the tree
- The estimated probabilities of the prior classes inside any terminal node

```
RESULTS BY SYMBOLIC OBJECT

     ============================================
     | No |    Nom       |Leaf    |  Class       |
     |    |              |  No    | true  | assig.|
     ============================================
     |  1 | "wave_1_3"   |      4 |     1 |  1    |
     |  2 | "wave_1_1"   |      4 |     1 |  1    |
```

```
|  .  | "........"   |      .  |     .  |  .     |
| 12  | "wave_2_2"   |      5  |     2  |  2     |
| 13  | "wave_2_7"   |      5  |     2  |  2     |
| 14  | "wave_2_5"   |      4  |     2  |  1 (*)|
| ..  | "........"   |      .  |     .  |  .     |
| 29  | "wave_3_8"   |      3  |     3  |  3     |
| 30  | "wave_3_9"   |      3  |     3  |  3     |
=============================================
```

For each object we have the code of the terminal node or leaf which is assigned, the a priori class or true class and the assign class. If the mark (*) exists then the object is misclassify.

R(T)= 0.0333

R(T) is equal to the sum of R(t) for all leaf of the tree T. R(T) represents the resubstitution estimate of the misclassification cost of the tree T.

On the test set the value R(T) gives an estimate of the misclassification cost.

CONFUSION MATRIX FOR TRAINNING SET

```
============================================================
     |           | wave_1    | wave_2    | wave_3    |  Total
|

============================================================
     | wave_1  |       10  |        0  |        0  |       10
|
     | wave_2  |        1  |        9  |        0  |       10
|
     | wave_3  |        0  |        0  |       10  |       10
|

============================================================
     | Total   |       11  |        9  |       10  |       30
|

============================================================
```

The value 1 represents the number of objects of a priori class wave_2 which are assigned to the assign class wave_2. The assign class wave_2 is build by the rules of the decision tree.

MISCLASSIFICATION RATE BY CLASS

```
   TRUE CLASS            (  ERROR /SIZE    )   FREQUENCY
   wave_1                (      0 /     10 )       0.00
   wave_2                (      1 /     10 )      10.00
```

```
     wave_3                (      0 /     10 )       0.00

     TOTAL                 (      1 /     30 )       3.33
```

The resubstitution global error rate is 3.33 %.

```
   ==================================
   | EDITION OF DECISION TREE       |
   ==================================


          +----  [ 4 ]wave_1  (    10.00      1.00      0.00)
          !
      !----2[ position  9 <= 3.800000]
      !    !
      !    +----  [ 5 ]wave_2  (     0.00      9.00      0.00)
      !
     !----1[ position 13 <= 3.810000]
      !
      +----  [ 3 ]wave_3  (     0.00      0.00     10.00 )
```

If the answers of the binary question [position 13 <= 3.810000]? is false then the object is assigned to a priori class wave_3 else the result depends of the response of the answers position 9 <= 3.800000]?. If the response is false then the object is assigned to a priori class wave_2 else the object is assigned to a priori class wave_1 and the vector (10.00, 1.00, 0.00) gives the number of objects of each a priori class assigned to leaf 4 (10 is the number of objects of a priori class 1 "wave_1").

Bibliography

[1] Ciampi A. *et al.* (1996): *Recursive Partition with probabilistically imprecise data*. Ordinal and Symbolic Data Analysis. Springer Verlag, E. Diday *et al.* (eds.), pp. 201-212.

[2] E. Périnel (1996) *Segmentation et Analyse des Données Symboliques. Application à des données probabilistes imprécises*. Thèse de doctorat de l'Université Paris IX Dauphine.

[3] E. Périnel, A. Ciampi. (1997). *Growing a probabilistic decision tree*, AMSDA 1997 Naples, Italy.

[4] E. Périnel (1998) Construire un arbre de discrimination binaire à partir de données imprécises.RSA.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SDT Help Guide

Strata Decision Tree

M.C. Bravo Llata
UCM (Madrid)

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

19 SDT : Strata Decision Tree

19.1 Scope of SDT

19.1.1 SDT overview

Decision trees provide an efficient means of construction discrimination functions dependent on many variables.

SDT implements a tree-growing algorithm for stratified data. Individuals, described by a set of predictors and by a classifying variable, are also divided into strata. A generalized tree-building algorithm is proposed to identify sets of strata for which the same rule for prediction is obtained. Each decisional tree node, not necessarily a tree leave, is composed of a set of strata and gives a rule for individuals in these strata to explain simultaneously the dependent variable. Algorithm is based in a recursive partition method that combines in each step maximisation of an information content measure and selection of decisional nodes.

SDT gives prediction for input data of unknown classes (identified by MISSING values).

VSDT let visualisation of the built Strata Tree as a capability for Strata Tree explanation, together with a symbolic description of the Strata Tree, decisional nodes and strata.

19.1.2 Identification of the design and prediction samples

Objects in the sample to be predicted are identified by SDT with a missing value (NU) in the predictive variable in the input SODAS file.

Treatment of missing values and hierarchical dependences

Missing values or non-observable (internal Nu or null values) are permitted and hierarchical dependences (internal NA or non-applicable values) are also permitted. A hierarchical dependence is defined by an expression like “If variable1=value then variable2=Non-Applicable (NA)”. Hierarchical dependences can be specified in DB2SO.

SDT is able to process NA values without an associated hierarchical dependence.

See Annex for treatment of missing values and hierarchical dependences.

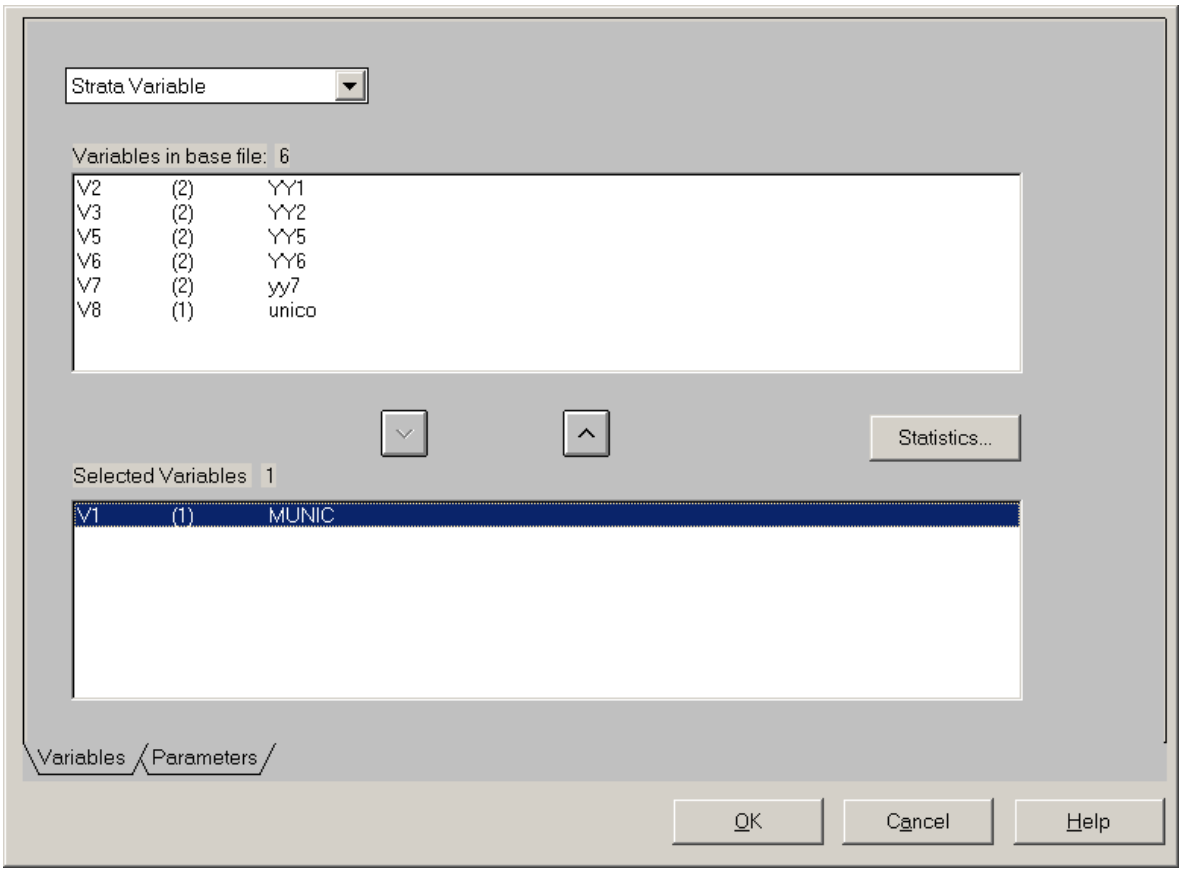
Notes about SDT version 2.23

Current version admits individuals described by categorical single (classical) and modal (probabilistic) binary predictors. Strata and class variables must be categorical single. Class variable must be binary.

Non binary predictors could be analysed, building outside SODAS, binary variables with the binary partitions of the categories of the predictor and specifying, in DB2SO, the corresponding hierarchical dependences.

19.2 Parameters of SDT

SODAS user interface to select strata, predictive and predictor variables:



STRATA. Strata variable. This variable must be of CATEGORICAL SINGLE type.

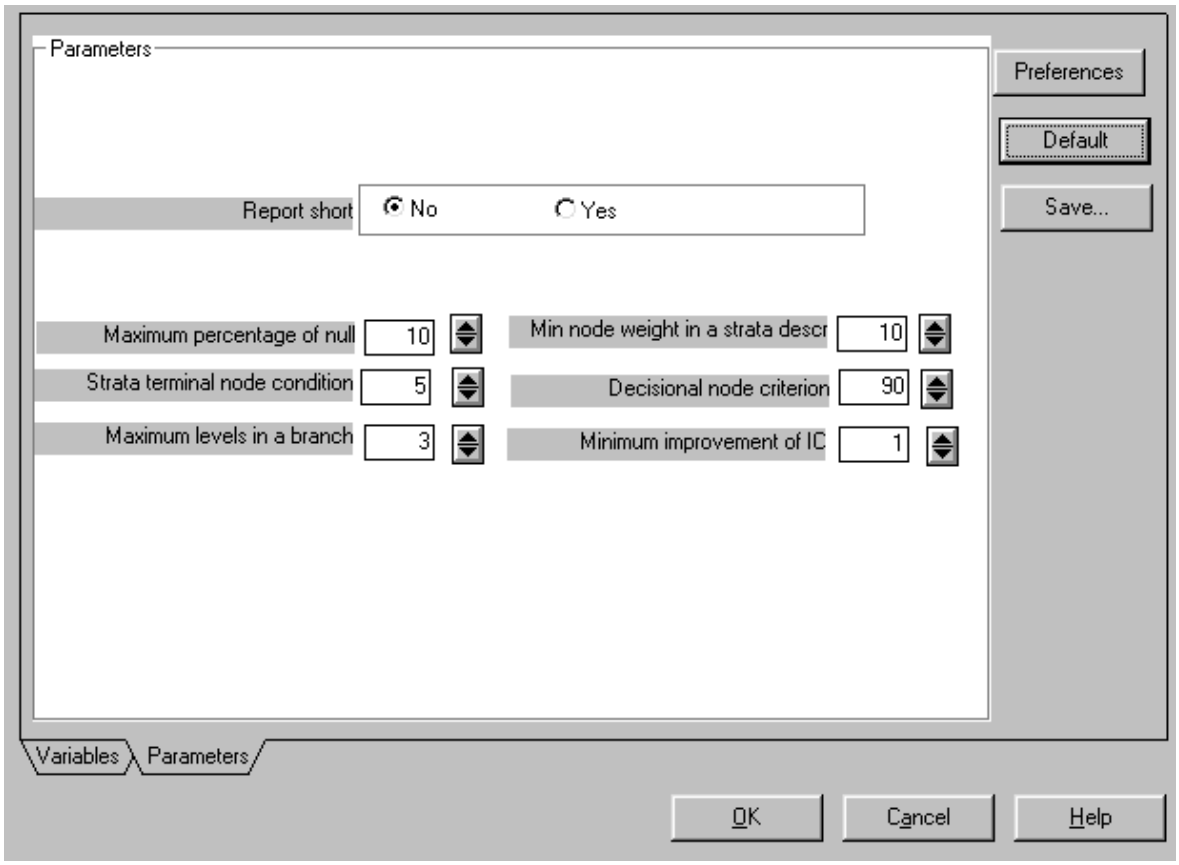
PREDICTIVE Predictive variable. This variable must be of CATEGORICAL SINGLE type and binary variables.

PREDICTORS Predictor variables. These variables must be of CATEGORICAL SINGLE type or MODAL (Multivalued probabilistic) type and binary variables.

SDT checks these requirements. If they are not fulfilled, then error or warning messages are produced.

Input SODAS / XML files can be generated through DB2SO.

SODAS(WB) user interface for specification of SDT parameters:



Note: Maximum percentage of null in this figure should be maximum percentage of missing

Report short *Printout of a short report*

Possible values: NO, YES. Default value: NO

Maximum percentage of missing Predictor variables with a percentage of missing values (*missing* values) in input objects bigger or equal to this parameter are not analysed. When the percentage of missing values (*missing* values) in input objects in a predictor is lower to this percentage, then the objects with missing values are removed from analysis. NOTE: Non-Applicable values in input objects without an associated hierarchical dependence (see DB2SO and SOM for more details) are treated as missing values.

Possible values: 0. to 100.

Specific value: 0. Predictors are not removed from analysis because of presence of missing values. Only input objects are removed from analysis.

Default value: 10.

Decisional node condition This number divided by 100 is the probability for decisional node criterion. *This means that strata for which the estimated probability for one of the classes is bigger than this value, then they are kept out from the partition method and form part of one or two "decisional nodes" (i.e. a prediction rule is obtained for these strata).*

Possible values: 0. to 100.

Default value: 80

Recommended values: [50,100]

Strata terminal node condition Minimum weight for terminal node condition for a stratum. Strata with weight lower than this value do not follow the recursive method, and then form a *"terminal node"*.

Possible values: A positive integer or zero value

Default value: 5

Maximum (number of) **levels in a branch**.Possible values: 1 to 15. Default value: 5

Minimum improvement of IC (Information Content measure) Minimum degree of information content improvement in a branch to follow the recursive partition method. This value is divided by SDT by 10000, to give a value between 0 and 1.

Possible values: 1. to 10000.

Specific value: 1. This option has no effect.

Default value: 1.

Example: An input value for *Minimum improvement of the IC* of 200, results a value of 0.02. An input value for *Minimum improvement of the IC* of 20, results a value of 0.002.

IC measure is a meausre of class uncertainty in the tree. Degree of improvement is measured as: (new_IC-old_IC)/old_IC.

Mininimum node weight in strata description. Minimum percentage of a decisional node to be in the printout of stratum description in the report file

Possible values: 0. to 100.

Specific value: 0. This option has no effect

Default value: 10.

Note: See SDT Repor file for a description of decisional, terminal and terminal-divide nodes.

Execution of the method and termination

After running of the method by SODAS user interface:

Normal termination of the SDT is shown by the presence of a white icon in the screen, placed at right side of SDT red icon. User can click twice this icon an edit the report file.

Abnormal termination of the SDT is shown by the presence of a brown icon in the screen, placed at right side of SDT red icon. User can click twice this icon an edit the log file and read the error message with an informative text. Input file and variables and parametes specifications must be checked.

Internal error messages (from I0001 to I0104) are caused by an abnormal execution of the SDT, for example a lack of memory. When an internal error message happens, user has to take note of the error code and error informative text and notify it to software distributor.

External error messages (from E0001 to E0055) are caused by incorrect data, for example incorrect input parameters. There are some warnings (W0001 to W0010).

For the complete list of error and warning messages produced by SDT, see SDT Software complete User Manual.

For Input limits to SDT see complete version of user manual.

19.3 Output of SDT

SDT Report file:

Short report contains :

List of input parameters

% of missing values in predictors

Description in SOL language of decisional nodes with their relative weights.

Description in SOL language of strata

Initial and final Information Content Measure

Long report contains :

Short report

The successive cuts obtained and the information content measure and its improvement in all steps

Report description:

Input parameters:

List of input parameters

Input variables and objects

List of input variables used in analysis

Information on weight and percentage of input objects with missing values (and NA values without an associated hierarchical dependence) for each predictor

Information on weight and percentage of input objects with missing values and NA values and for the strata variable.

Number and list of individuals not used in design sample because of:

- Missing values in a predictor or strata or predictive variables. (For predictors, see below)
- Na values in a predictor without an associated hierarchical dependence
- Na values in the strata or predictive variable.

Number and list of individuals not used in design sample and used to be predicted because of:

- Missing value in the predictive variable.

List of predictors not used because:

- the type is different from input requirements,
- the number of categories is bigger than a permitted or,
- the percentage of missing values plus NA values without an associated hierarchical dependence is bigger that the value of the parameter *Maximum percentage of missing*.
- When a variable is removed from analysis, objects with missing values or Na values in this predictor are not removed from analysis.
- Additional information is written in Log file as Warning messages.

Information content measure

Information on Initial and final content measure

Initial weight from observations in design sample that are used in the strata tree building algorithm.

From initial objects in the data matrix are removed:

the prediction sample objects (which have a missing value in the predictive variable)

objects with NA values in the predictive variable

objects with missing/NA values in the strata variable

strata with initial weights lower than *Strata terminal node condition*

for objects with missing/ NA values without an associated hierarchical dependence in predictors, see *Maximum percentage of missing* parameter and treatment of hierarchical dependences.

Number of explored nodes

Description of nodes

Description of decisional and terminal nodes are given by the symbolic objects describing them, the relative weight of their extensions in the stratum design sample, their internal Information Content measure and their contribution to the strata tree Information Content measure.

Short note on node identifications

Nodes are identified by n,m – XXXn with:

-n: level of the exploited parent node

-m: 'virtual' position in nth level for the node. It starts at 0 and finishes at $2^{n+1} - 1$

XXXn is one of these:

DEC0 Identifies a decisional node 'oriented to' first predictive class

DEC1 Identifies a decisional node 'oriented to' second predictive class

TD0 Identifies a terminal-divide node 'oriented to' first predictive class

TD1 Identifies a terminal-divide node 'oriented to' second predictive class

TER0 Identifies a terminal node

Decisional node descriptions

Description of decisional nodes obtained by the fullfilment of the decisional node condition. – for example:

3,0-dec0

[yy7 = Z71]^[YY5 = A52]^[YY1 = A2]^[MUNIC = {M3, M4}]^[ZZ = A1]

Weight: 13.00, IC: 0.000000, IC_SDT: 0.000000

Weight is node weight, IC is the internal node uncertainty of class variable and IC_SDT is contribution of node to final IC measure of the tree.

This node represents a prediction rule for Municipalities M3 and M4.

Terminal – Divide and terminal node descriptions

Terminal-Divide nodes are obtained by the fullfilment in the parent node of one of the following conditions. Parent node is then divided into two terminal nodes or it is itself a terminal node. The conditions are:

there are no more predictors for analysis

the increment of strata tree information content measure from this step to the following 'virtual' one is lower than the value derived from the *Minimum improvement of the IC* parameter

the level of the branch has reached the value of the *Maximum levels in a branch* parameter

For example:

3,6-td0 - [yy7 = Z72] ^ [YY5 = A51] ^ [YY6 = b622]^[MUNIC = {M1,M3,M4}] ^ [ZZ = 0.36 A2, 0.64 A1]

Weight node: 11.00, Internal IC measure: -0.655482, IC_SDT (contritubition to the IC to Strata Tree): -0.090129

Terminal nodes are obtained from the fullfilment of the strata stopping criterion.

Not relevant nodes

Here are the decisional, terminal-divide and terminal nodes with less weight than relevant threshold value.

Relevant threshold: It is computed as $\min_{\{strata\}} \{ 2.5 * \text{Initial weight of stratum} / 100\}$

Description of strata.

Each stratum is described by the decisional and terminal-divide nodes that characterise it and their respective relative weights (from 0 to 100). In this output description, nodes with relative weight (from 0 to 100) bigger than the value of *Min node weight ia a stratum descr* parameter are shown.

For long report, detailed description of the strata tree building process.

A new information shown in the new decisional nodes is given. For each stratum, its weight in the node and the weight of the first class of the predictive variable are shown.

Prediction

Prediction for objects of strata included in the design sample used in the strata tree building process are given.

Number of individuals predicted

Class *class 1 description:*

Objects predicted to be in first class are given. For each object, three elements of information are given:

object description

predicted probability for the class

decisional/terminal node identification with higher weight in this prediction

Class *class 2 description:*

Objects predicted to be in second class are given. For each object, the same information as for first class predicted objects is given.

Especial cases (Not fiable descriptions)

Prediction sample objects with some missing/NA values that makes not possible a completely fiable prediction.

Class *class 1 description:*

Objects predicted to be in first class are given. For each object, four elements of information are given:

object description

predicted probability for the class

decisional/terminal node identification with higher weight in this prediction

fiability (from 0 to 1)

Class *class 2 description:*

Objects predicted to be in second class are given. For each object, the same information as for first class predicted objects is given.

Unpredictible individuals

Prediction sample objects that cannot be predicted because:

they have a missing value or NA value in the strata variable

they have a missing value or NA value in a predictor which makes impossible the fullfilment of any decisional / terminal node

Log file:

Log file contains :

List of strata

Hierarchical dependences in input SODAS / XML file

Clock ticks between depth levels in tree building.

Information on node processing

Warnings and error messages.

Warning and error messages are described in Section 4 of this document.

Graph file:

Graph file contains :

Information of Strata Decision Tree to be visualised by VSDT in an ASCII or binary file. The graph file is visualised by VSDT in an automatic way by a double click of the mouse on VSDT icon.

19.4 SDT Visualisation

VSDT is a graph editor for Strata Decision Trees. It performs Strata Tree Visualisation. It has a menu driven interface with user.

Opening a strata tree graph from SODAS software.

The left upper corner of the window is shown.

Characteristics of the editor:

Initial and final Information content measure is shown.

Different colors depending on node types.

Different intensity of colors depending on class predictive category:

- Light color for first predictive class
- Dark color for second predictive class

These classes are identified in the left upper corner of the window when a tree is visualised.

Rounded nodes for exploratory nodes.

Squared nodes for terminal and decisional nodes. ***Only squared nodes with a weight bigger than threshold value*** (see Options/Change options/Threshold value) are shown.

Information of cut variables and categories.

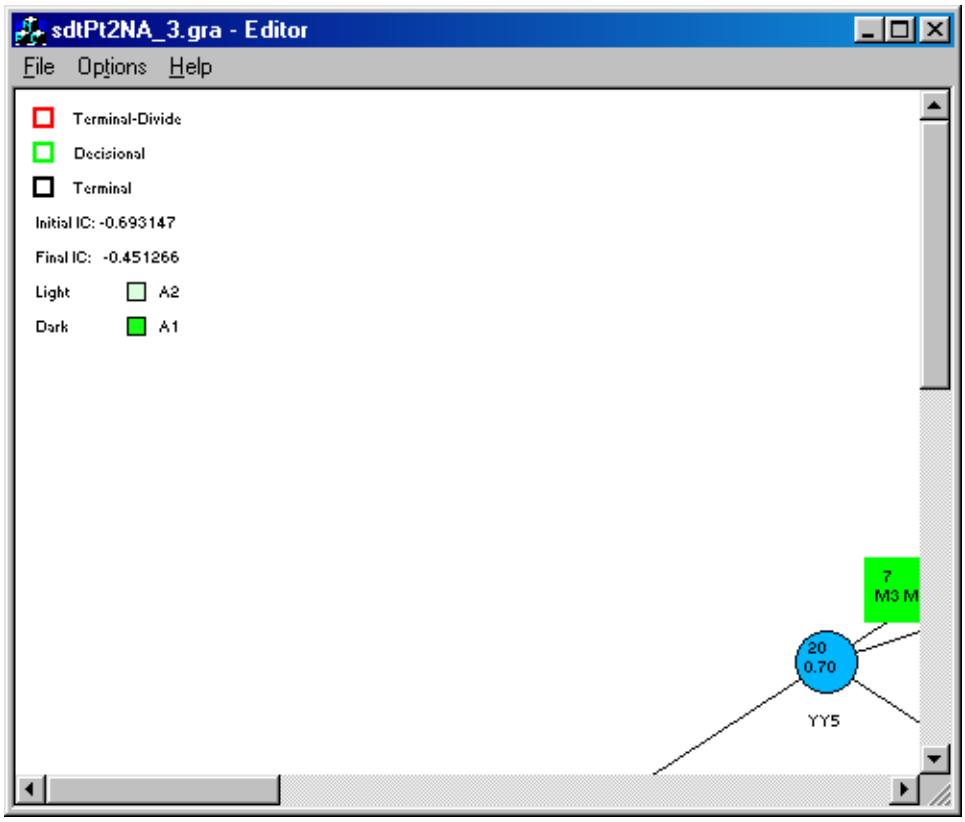
Information on nodes is:

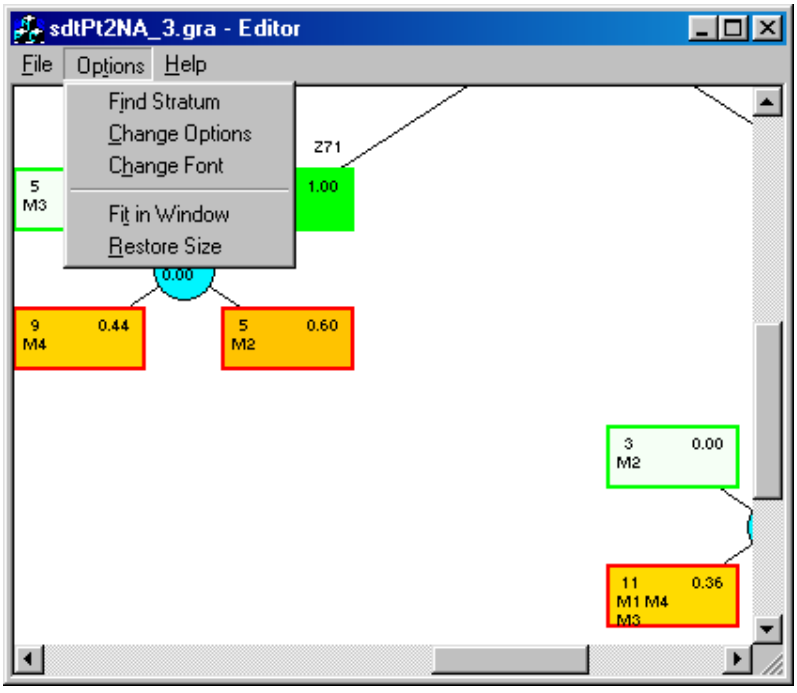
- For rounded nodes: Weight (without decimal information) and predictive first class probability.
- For squared nodes: Weight (without decimal information), predictive first class probability, strata.

Some functionalities:

- Find stratum implemented to signal terminal (decisional, terminal, terminal-divide) nodes with prediction rules for a stratum
- Fit in window or one-right click mouse button. To fit the strata tree in one screen.
- Options/Restore window or one-left click mouse button (from the fit in window). To go to the restored size window. The contents shown from the restored size window is connected with the place where the mouse is clicked in the 'fit in window' window. When the option Restore window is clicked, the left upper corner of the restored size window is shown.
- Double-left click mouse button (from the fit in window) or one-left click mouse button (from the restored size window) in one strata tree node. To show complete node information.

An Example of a strata tree opened by SODAS in two different views of the same window:





Left corner in first figure shows the node colors for Terminal-divide, decisional and terminal ndoes (See SDT Report file for a description of these nodes), Initial and final IC measures (IC measures before and after tree building) and the identification of light nodes with class A2 (first predictive class) and dark nodes with class A1.

In both figures we can see:
Exploitable nodes are rounded. Information contained inside is:

- its weight
- Estimated probability for first class of the predictive variable

Decisional and Terminal-Divide nodes are squared. Information contained is:

- its weight
- Estimated probability for first class of the predictive variable
- Strata defining the node, when possible.

Scrolling is permited in these screens.

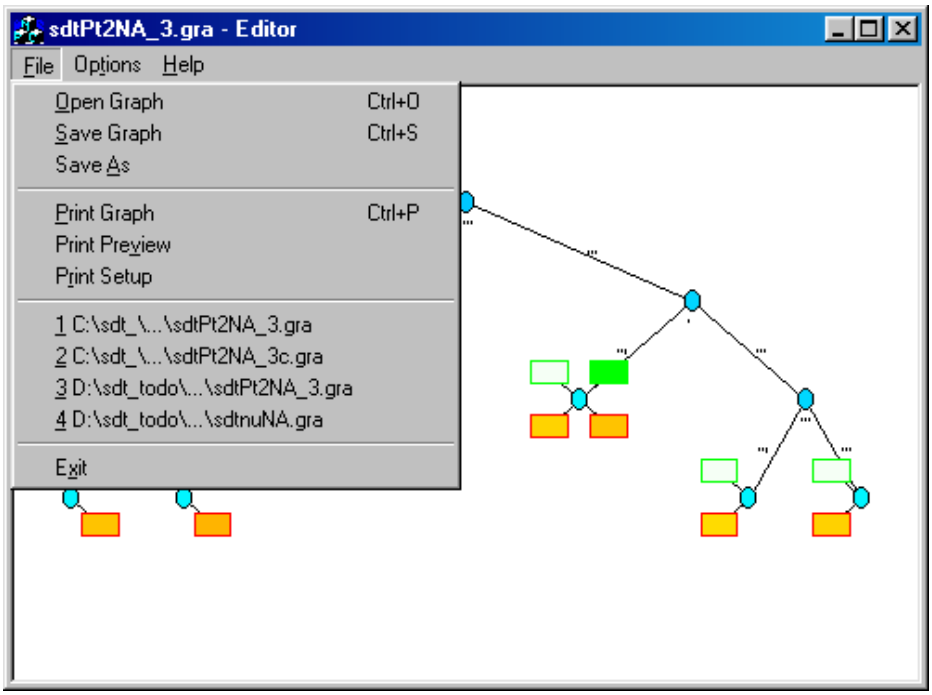
Main menu options are :

File
Options
Help

File menu options are :

Open graph	to select a strata decision tree graph file for visualisation
Save graph	to save a strata decision tree graph file
Save as	to change the name of the graph file
Print graph	
Print preview	
Print setup	it is related to user printer
Exit	to exit from VSDT

In File menu, the last four files opened with VSDT are shown.
File menu options in a 'Fit in Window' strata tree (see Options/Fit in window):



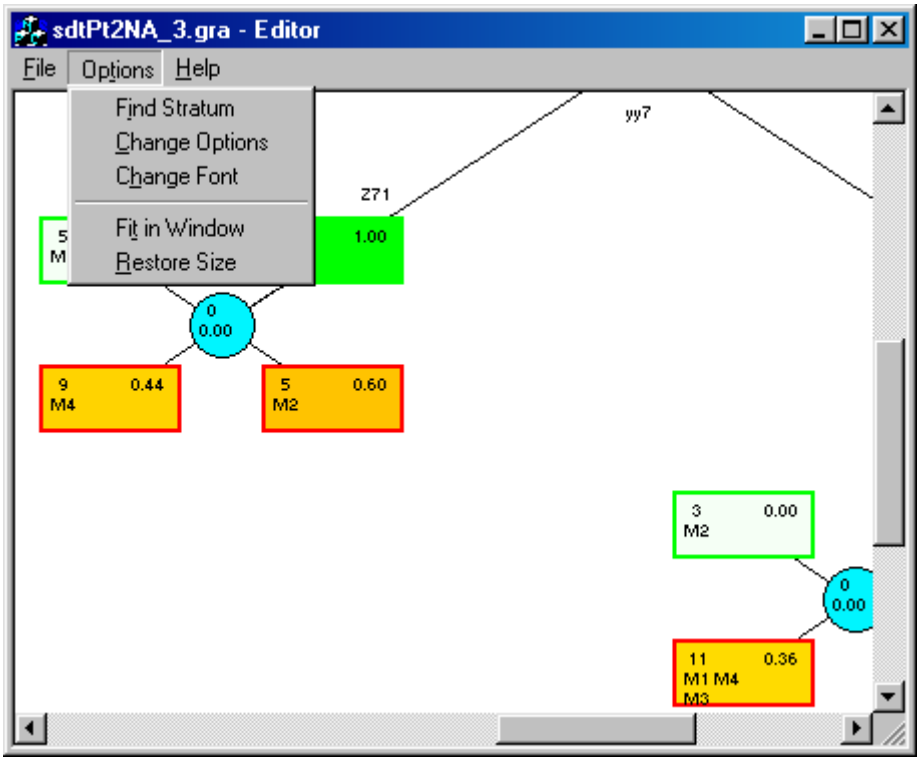
Options menu

Options menu options can be activated only from restored size windows, with exception made for Restore window option.

Options of Options menu are :

Find stratum	to select a stratum to be highlighted in the visualisation
Change options	It is a check text box to change options of visualisation
Change fonts	It is a check text box to change fonts and sizes
Fit in window	It draws the strata decision tree in one screen
	This option can be activated by one right-click mouse button
Restore window	It restores the strata tree in the original window
	This option can be activated by one left-click mouse button

Options menu in a restored size window:



For simplicity in posterior explanations Options/Fit in window and Options/Restore size are described first

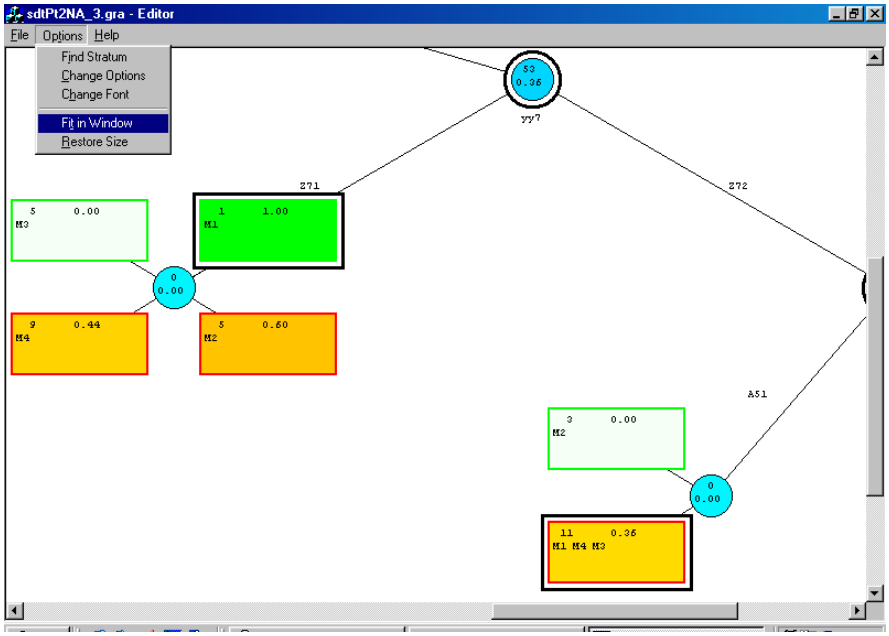
Options/Fit in window

This option makes the complete strata tree be in a screen.

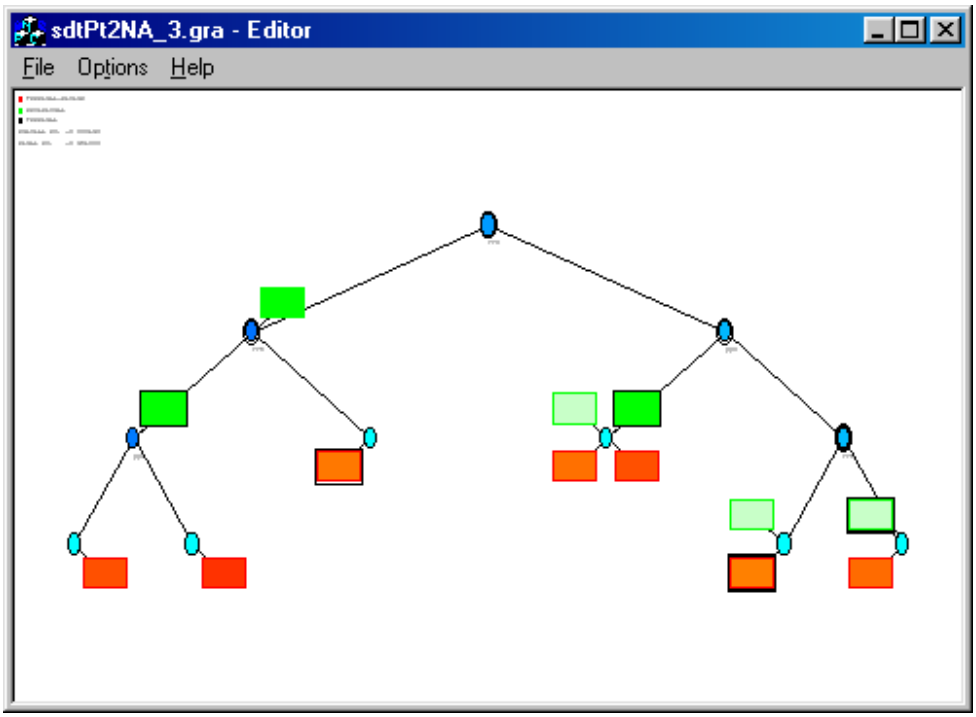
This option is equivalent to press the right-click mouse button.

The effect of this is shown in the following two screens:

Screen 1:



Screen 2. The effect of this option gives:



Options/Restore size

This option makes that strata tree fitted in a window be in the original window. When this option is clicked, the left upper corner of the restored size window is shown.

This option is equivalent to press the left-click mouse button. The contents shown from the restored size window is connected with the place where the mouse is clicked in the 'fit in window' window

Options/Find stratum text box:

The aim of this text box is to highlight nodes where a stratum is present.

As told before this option can be activated only from restored size windows.

It has effect for restore size windows as well as for fit in window screens.

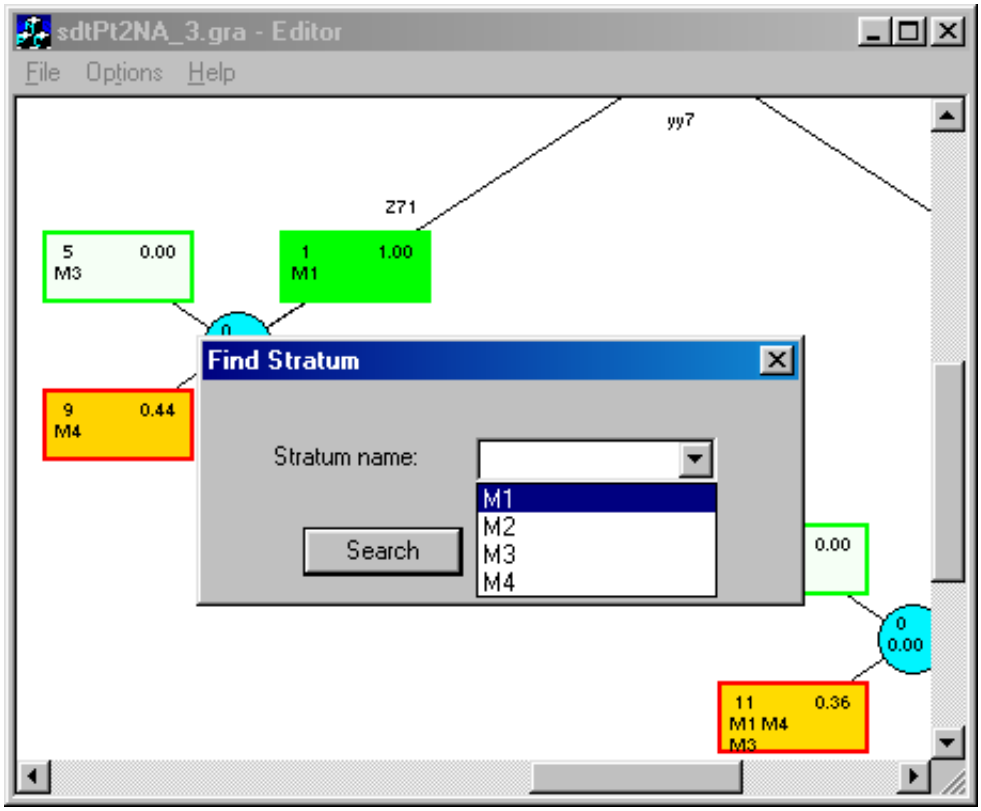
There are two alternative ways of using this text box:

-Stratum can be written in the text box. In this case the eight-character description has to be written. If the description has less characters, then they are written instead.

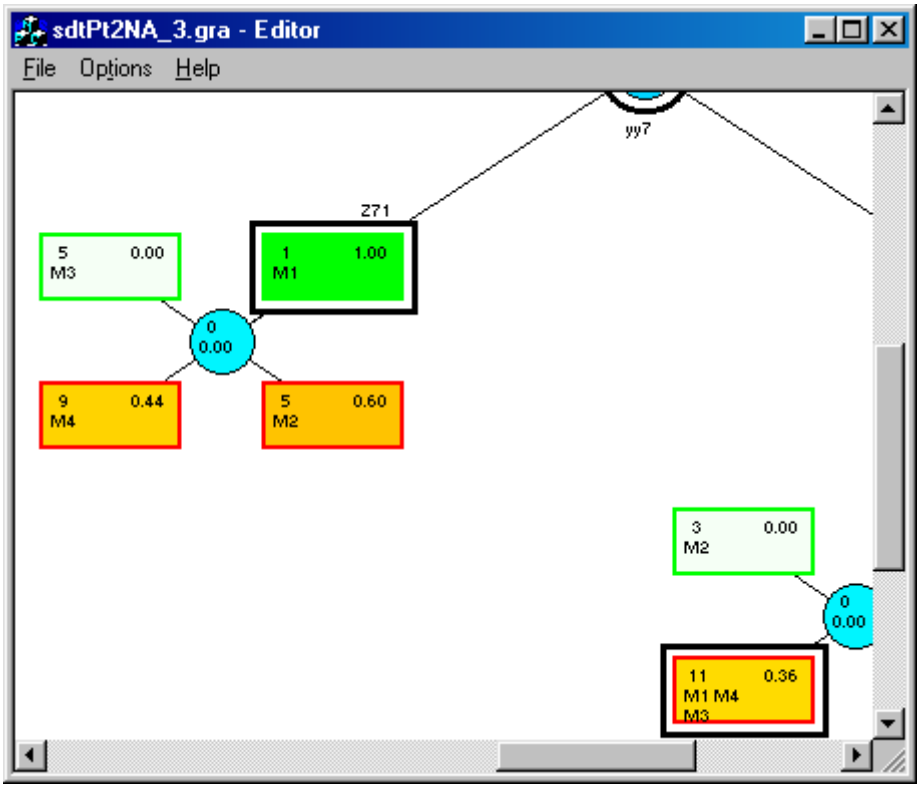
-Arrow can be displayed and a stratum can be selected from this list. SDT in previous versions to v2.21 generates graph files (.gra) which show an empty list here. In this case, previous way will be chosen.

-In both cases, search button must be clicked.

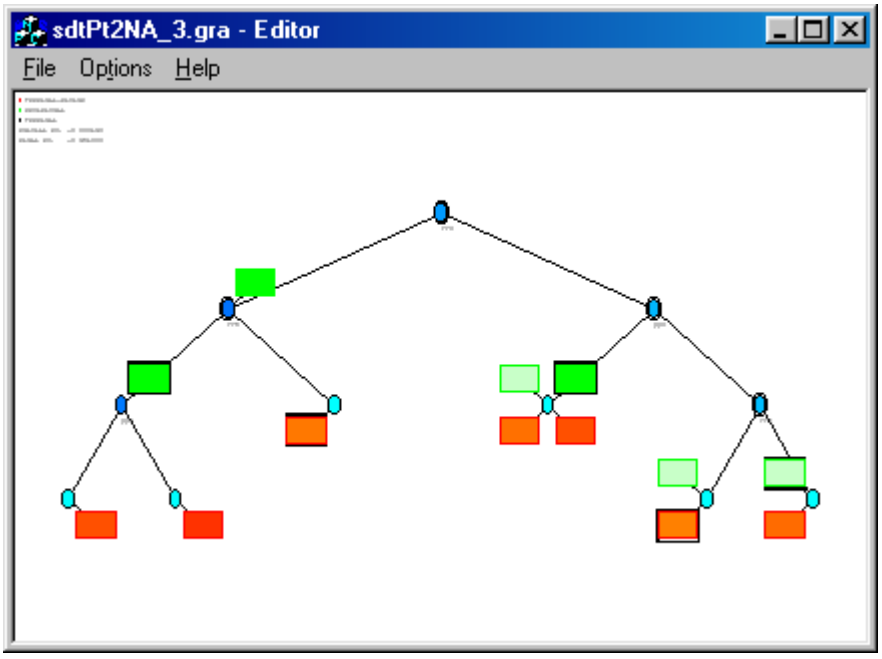
Example of Find stratum text box where 'M1' stratum is to be highlighted:



gives:



From this window, if Options/Fit in window is selected or the right-click mouse button is pressed, the following window is shown:



Options/Change options check box :

As told before these options can be activated only from restored size windows.

Only the effect for Relevant threshold text box and Show decisional/terminal nodes have an interesting result in fit in window screens.

Change options box has the following check text boxes :

Maximum number of strata : An integer number from 1 to 10. Default :4

It is the number of strata in a decisional/terminal node to be shown. If more strata are present in a node then an asterisc is shown and the information for this node has to be inspected inside the node.

Maximum number of characters per strata : An integer number from 0 to 4. Default :4

It is the number of character per strata in a decisional/terminal node to be shown.

Maximum number of characters per line : An integer number from 1 to 12. Default :10

It is the maximum number of characters per a line in a decisional/terminal node.

Relevant threshold : An real value Default : 0 or a value computed by SDT.

Decisional and terminal nodes with weight lower than this value are not shown. This option can have the effect of ***rounded (exploitable) nodes with weight 0 in the last depth level with no terminal / decisional nodes.***

Change options box has the following check thick boxes :

Show data : to show data in rounded nodes

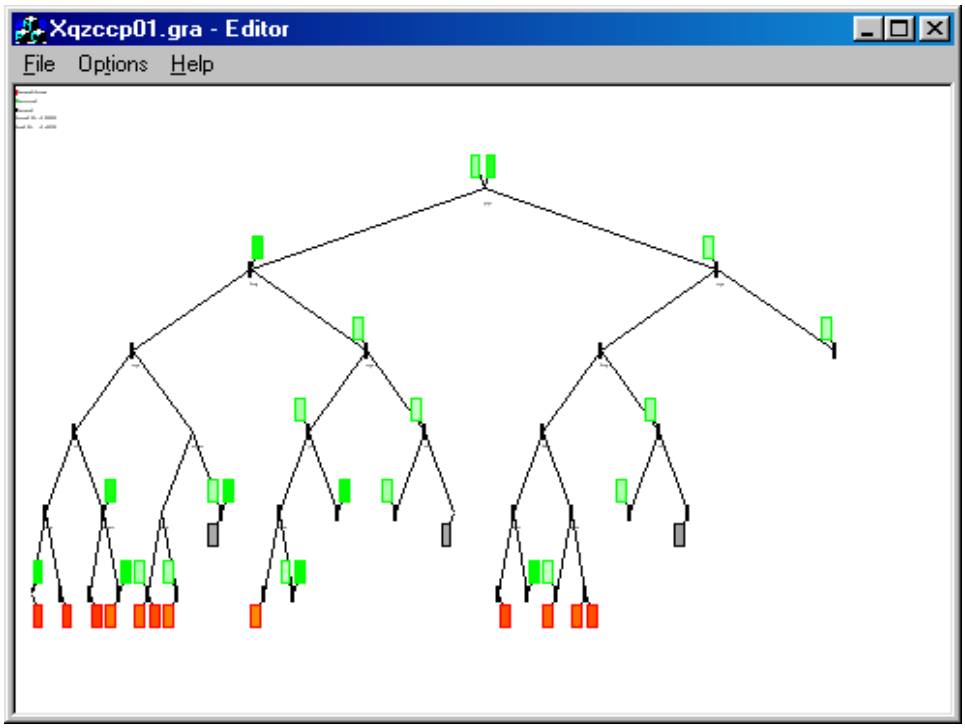
For decisional / terminal nodes :

Show decisional/terminal nodes

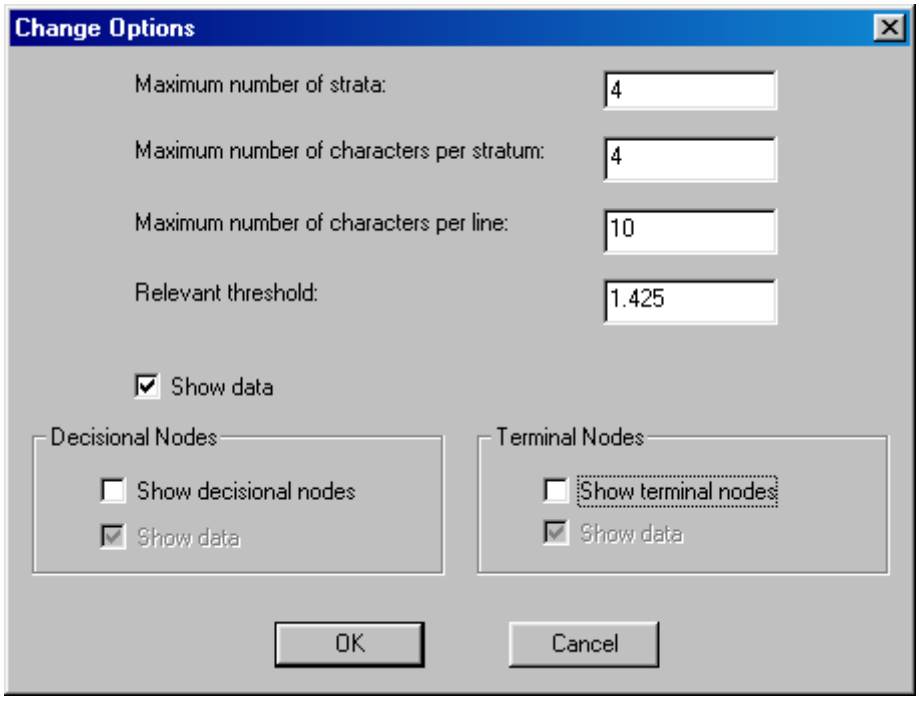
Show data

Note: For high depth level strata trees it can be interesting to unselect the show decisional/terminal nodes to have a more complete view of cut variables. This is applicable for fit inw window screens and for restored size windows. Three following screens show the effect of this option:

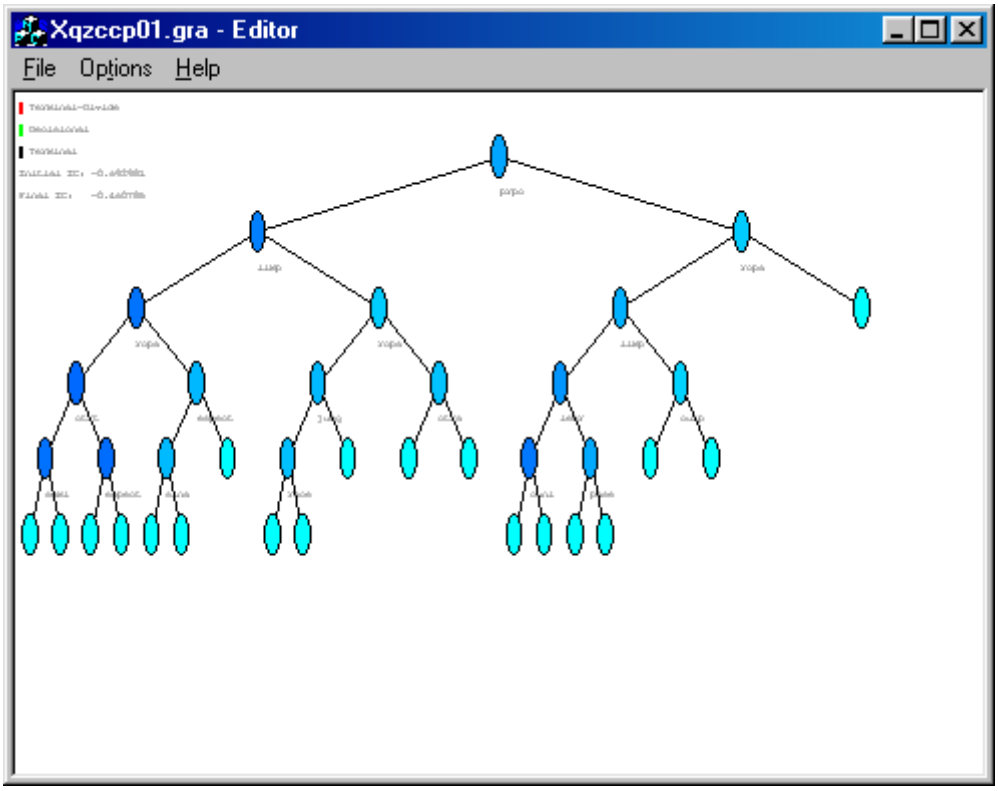
First screen: A 5-level-depth tree after Options/Fit in window:



Second screen: The Options/Change options text box:



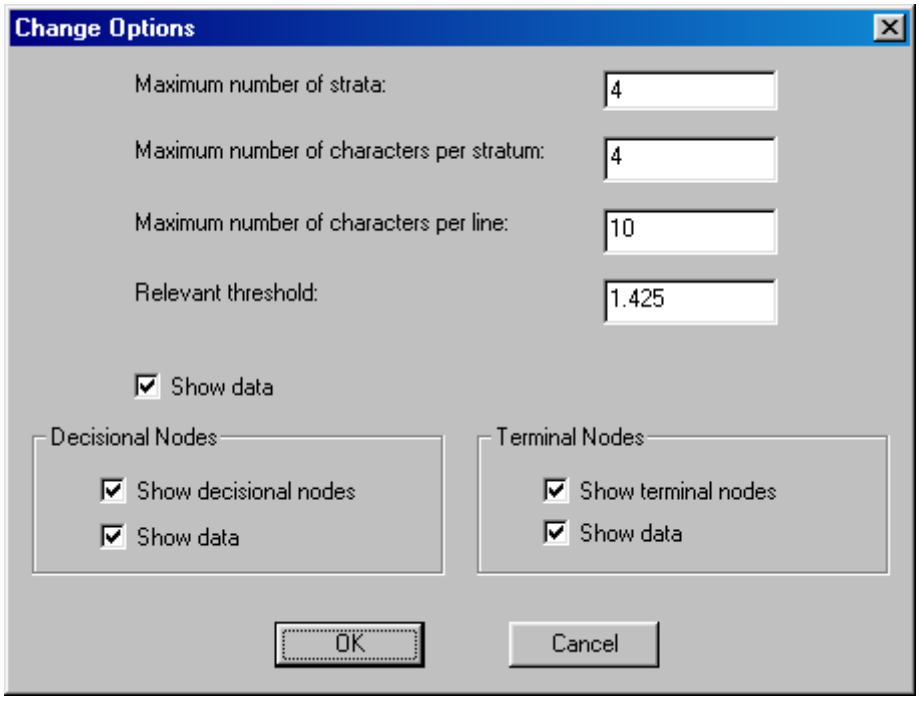
Third screen: The same strata tree from the first screen after Options/Fit in window:



Change options check text box has the following buttons :

Ok

Cancel



Options/Change fonts box

As told before this option can be activated only from restored size windows.

Change fonts box has the following check text boxes :

Font	to select the font
Font style	to select the style of font
Size	to select the size of the font
Color	to select the color of the font
Alphabet	to select alphabet

Change fonts box has the following check thick boxes :

Erasure

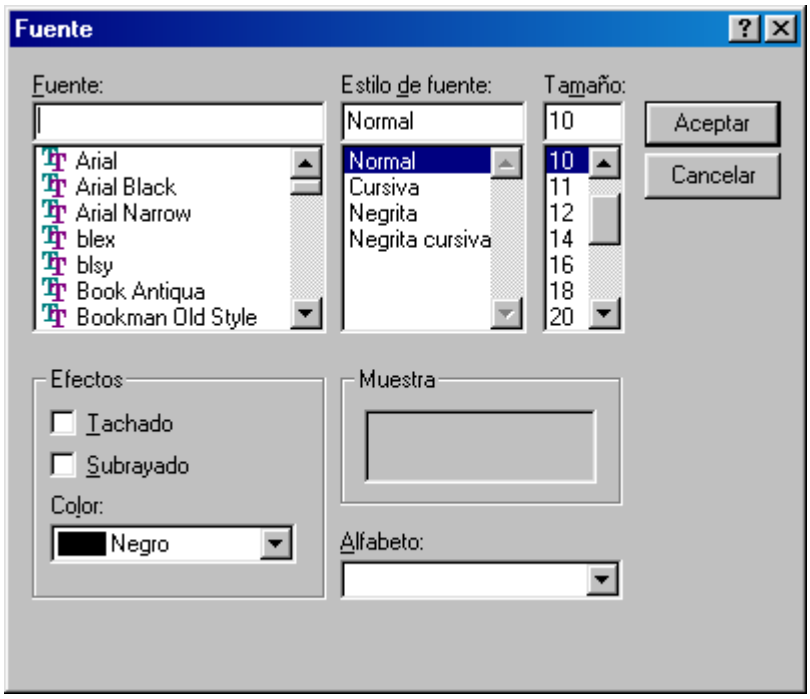
Underlined

Change fonts box has the following buttons :

Accept

Cancel

Change fonts box has an example output text box to show how the chosen font is.



Note: This printout is in Spanish because of the Windows operating system in Spanish version.

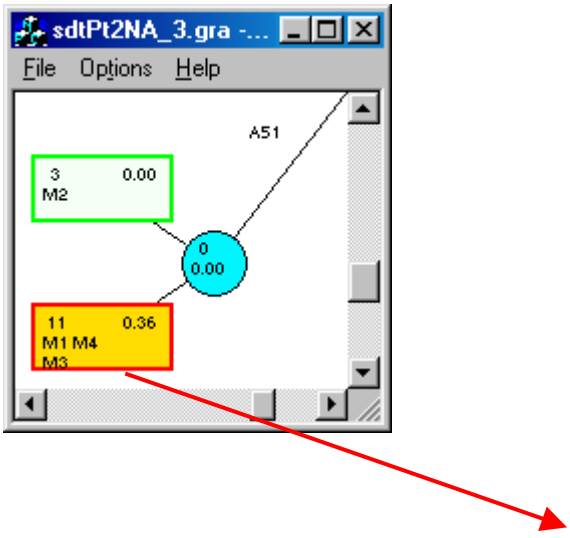
Double-left click mouse button (from the fit in window)

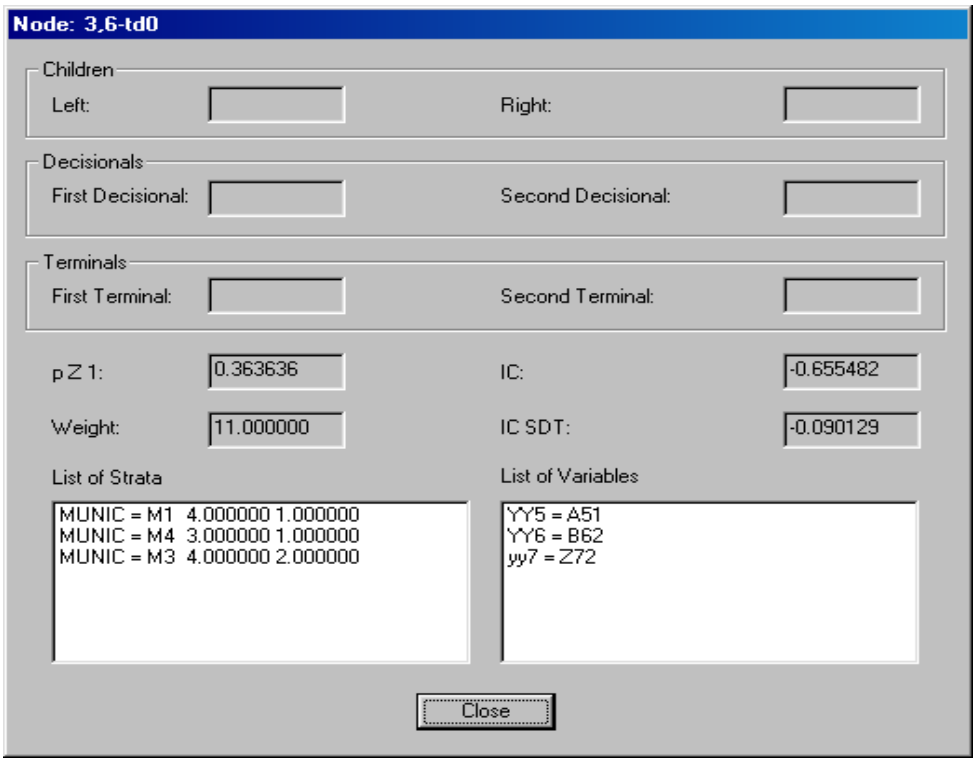
This click must be done on a node.

It is equivalent to one-left click mouse button (from the restored size window) in one strata tree node.

Node information is shown.

Node information for a decisional node:





This information corresponds to node 3,6-td0 represented by the symbolic object:

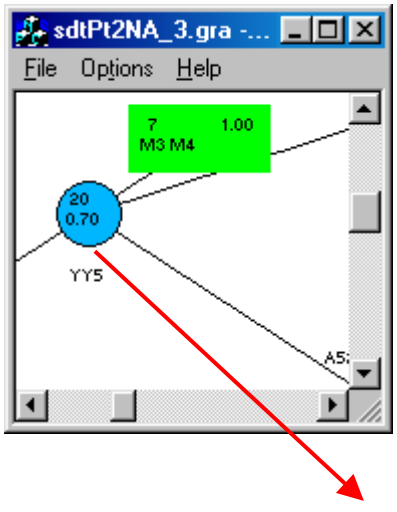
[yy7 = Z72] ^ [YY5 = A51] ^ [YY6 = b622]^[MUNIC = {M1,M3,M4}] ^ [ZZ = 0.36 A2, 0.64 A1]

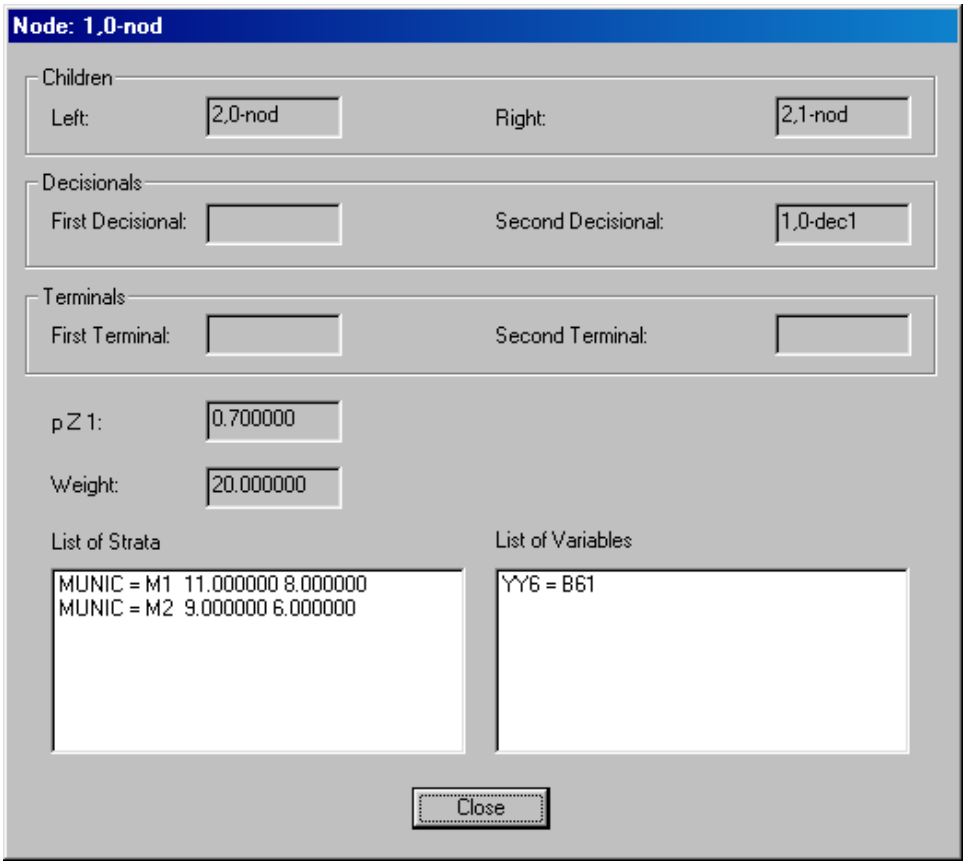
Weight node: 11.00, Internal node Information content (IC) measure: -0.655482, IC_SDT (contritubition of the node to the Strata Tree IC Measure): -0.090129.

The sum IC_SDT measures on decisional, terminal-divide and terminal nodes that compose the final tree is the Strata Tree final IC measure. The contribution of each node is their own internal node IC measure weighted by relative node weight.

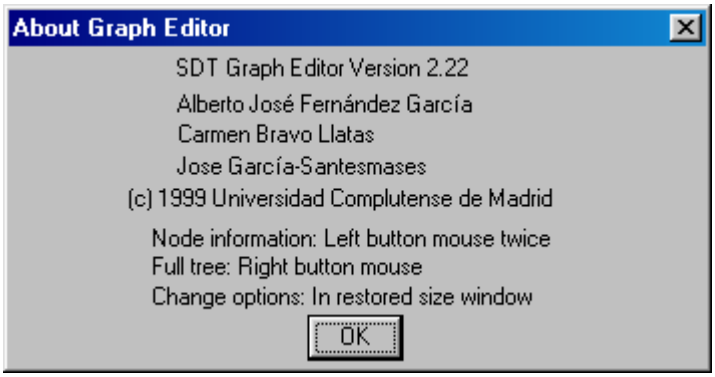
The two last columns in the list of strata show strata weight in the node and strata weight for first predictive class

Node information for an exploitable (rounded) node:





Help box



Hits for VSDT

Given that VSDT was not SODAS contractual software some functionalities could not be fully optimised. Some suggestions are given to manage the strata tree in a way to look at the visualisation in an easier way.

For an easier navigating in screen, some alternative suggestions are given:

- Select a smaller font size, let's say 6. For a restore size window
- Select the fit in window option
- Execute again SDT with a smaller number of levels in a branch and visualise the strata tree again

For an easier printout, these suggestions are given:

- Do it always from the restore size window
- Select a bigger font size, let's say 20 or more
- In print setup, choose the horizontal orientation

Termination :

If an error happens during VSDT execution, a separated window is shown to the user. When an internal error message happens or unexpected error message happens, user has to take note of the error code and error informative text and notify it to software distributor.

The list of error messages produced by VSDT can be seen in the complete version of SDT user Manual.

To exit from VSDT user interface, chose in File menu, Exit option.

ANNEX

Treatment of missing values and hierarchical dependences

SDT v2.23 treats missing values as follows:

- Objects with missing values in the strata variable are removed from analysis and prediction.
- Objects with missing values in the predictive variable are removed from analysis and considered as objects to be predicted.
- Objects with missing values in a predictor are either removed from analysis or not depending of the value of parameter *Maximum percentage of missing*. (See description of this parameter in this document).

Treatment of hierarchical dependences in SDT v2.23:

Note:SDT is able to process NA values without an associated hierarchical dependence.

- Objects with NA values in the strata or predictive variable are removed from analysis and prediction.
- Objects with NA values without an associated hierarchical dependence are treated as if they had a missing value for this predictor.
- Objects with NA values with an associated hierarchical dependence are treated following these specifications:
    - The antecedent is a predictor. A predictor that is a consequent in a hierarchical dependence is checked for segmentation in an exploitable node only when the predictor precedent of the hierarchical dependence is in the definition of this node and the value of the precedent is not the value that makes consequent to be NA in an object.
    - The antecedent is the predictive variable. Warning W0009 is produced. Predictor is removed from the list of predictors.
    - The antecedent is the strata variable. Warning W0009 is produced. Predictor is removed from the list of predictors. If it desired to analyse this predictor, then user is suggested to remove the strata in the hierarchical dependence premise from the data matrix.

Categorical single predictor variables

This is equivalent to analyse classical data, that is, one assertion per individual. Each group of individuals contains only one individual. Here, a group is a set of individuals for which an assertion will be built.

Treatment of categorical single predictor variables in SDT is a particular case of treatment of modal (probabilistic) predictor variables. For this reason, when analysing categorical single type variables they can be put as modal (with *modes* or *probabilities* being 1.0 or 0.0).

Nevertheless, computing time can be increased when using modal type variables instead of categorical single variables.

To build a SODAS file, the following steps can be followed (in a very schematic way). See DB2SO for more details:

Prepare the data base in ACCESS with a table and two related queries:

- Query1: It is composed by the variables:

  ID_Individual ID_group one or more text variables

For these text variables, DB2SO will generate modal or categorical multi-valued variables. At least one variable must be put in this query.

For SDT, in DB2SO in VIEW / VARIABLE PROPERTIES select the probabilistic option for these text variables.

- Query2: It is composed by the variables:

  ID_group text variables

  (with the option grouped by if there are several individuals with the same ID_group)

  These text variables are at least the predictive and the strata variables.

  Categorical single predictor variables can be in this list or in Query1. (Better in this list of variables) See above 'Categorical single Predictor variables' for more details.

  For these text variables, DB2SO will generate categorical single variables.

Hits:

When identifying groups in ACCESS by concatenation of several variables, the strata and the predictive variable usually are in this concatenation. (Given that they must be of categorical single type in the input SODAS file)

When all predictor variables are of categorical single type then the contains of ID_group variable can be the same as the contains of ID_individual variable

Predictor and predictive variables must be binary

Build the SODAS file with DB2SO :

- FILE / NEW
- Select * from Query1
- VIEW / VARIABLE PROPERTIES and select the probabilistic option for the text variables in Query 1
- MODIFY / ADD ONE-SINGLE VALUED VARIABLES
- Select * from Query2
- FILE / EXTRACT

Note: When the number of assertions is big (let's say for a pentium 166 Mz. 32 Mg., 10.000 assertions, 8 text variables in the second query) then the computing time for this second query and the file / extract operation can be long (let's say 10 minutes or more for each). These data are aproximated and derived from own examples, which means that they are not necessarilly data in all circumstances.

See DB2SO for more details.

Abbreviations and Acronyms :

- DB2SO: Short identification for the Data base to Symbolic Object
- IC: Information content measure
- Nu value: Internal name for a missing value for a Symbolic object in a variable.
- NA value: Internal name for a Non-applicable value for a Symbolic object in a variable. It usually comes from a hierarchical dependence. Nevertheless, SDT is able to treat NA values without an associated hierarchical dependence.
- SDT: Short identification for the STRATA_DECISION_TREE
- VSDT: Short identification for the STRATA_DECISION_TREE_GRAPH_EDITOR
- SO: Symbolic Object
- SODAS: Symbolic Object Data Analysis System
- SOM: Short identification for the Symbolic Object Management
- SOL.: Symbolic Object Language (SOM)
- WB: Short identification for the WorkBench

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SBTREE Help Guide

Bayesian Decision Tree

Edited by FUNDPMa

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: 02/10/2003

20 SBTREE : Bayesian Decision Tree

20.1 Introduction

The Bayesian decision tree, based on density estimation, aims to classify new objects to one class of a prior partition.

Each split is carried out selecting the best discriminant variable and the classification step is performed according to the Bayesian decision rule thanks to kernel density estimation.

This description requires a class-specific density estimate and an optimal choice of the corresponding window bandwidths. Suitable prior probabilities are also computed.

The proposed method gives rules in order to classify the news objects.

20.2 The input to the SBTREE module

SBTREE studies the case where a set of Symbolic Objects are described by p interval variables. The interval-valued variable is measured for each element.

Furthermore, the input symbolic objects must be labeled. A categorical single variable in the dataset (sds file) provides that information. The value of this variable is the identification number of the class for each training object and 99 for each object to be classified.

We take as an example an artificial database, with 20 symbolic objects, described by two interval variables and a categorical single variable. The chosen “BASE” is the file artificial.sds.

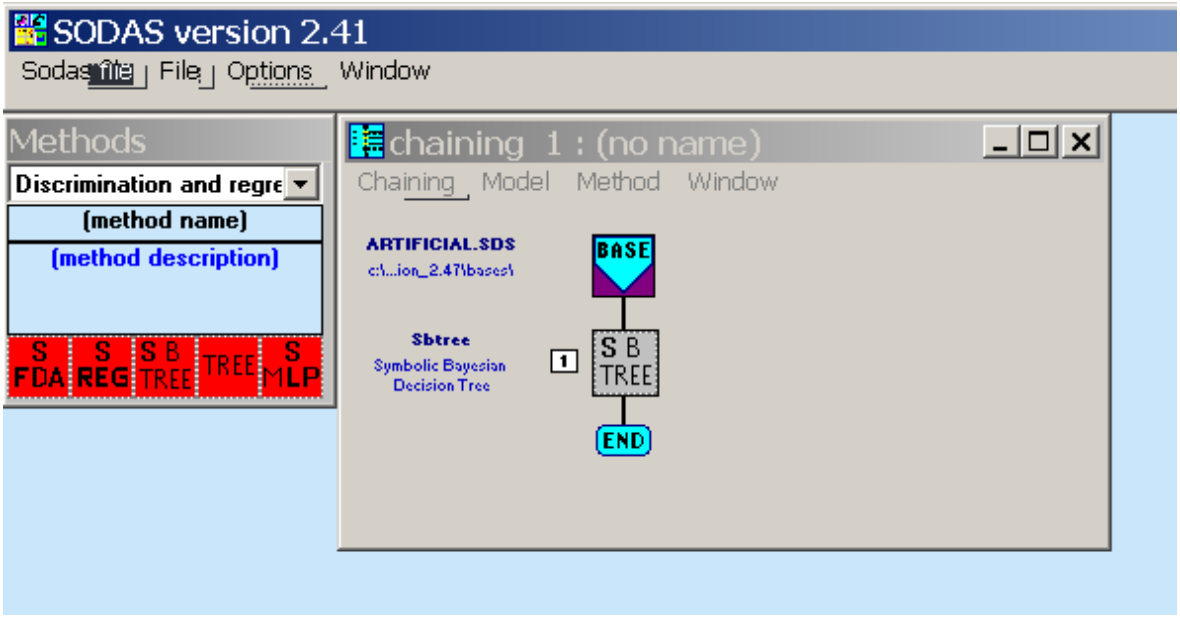


FIG. 1 Chaining example

20.3 The parameters of SBTREE

The user must specify the values of a variety of parameters which characterize special options. Insofar the user is able to tune the parameters such that the resulting clustering method fits optimally his special application problem.

20.3.1 Selected variables

The user has to choose the active variables and the class identifier among the dataset variables.

The active variables are in fact the ones which will serve to the affectation of the unclassified objects.

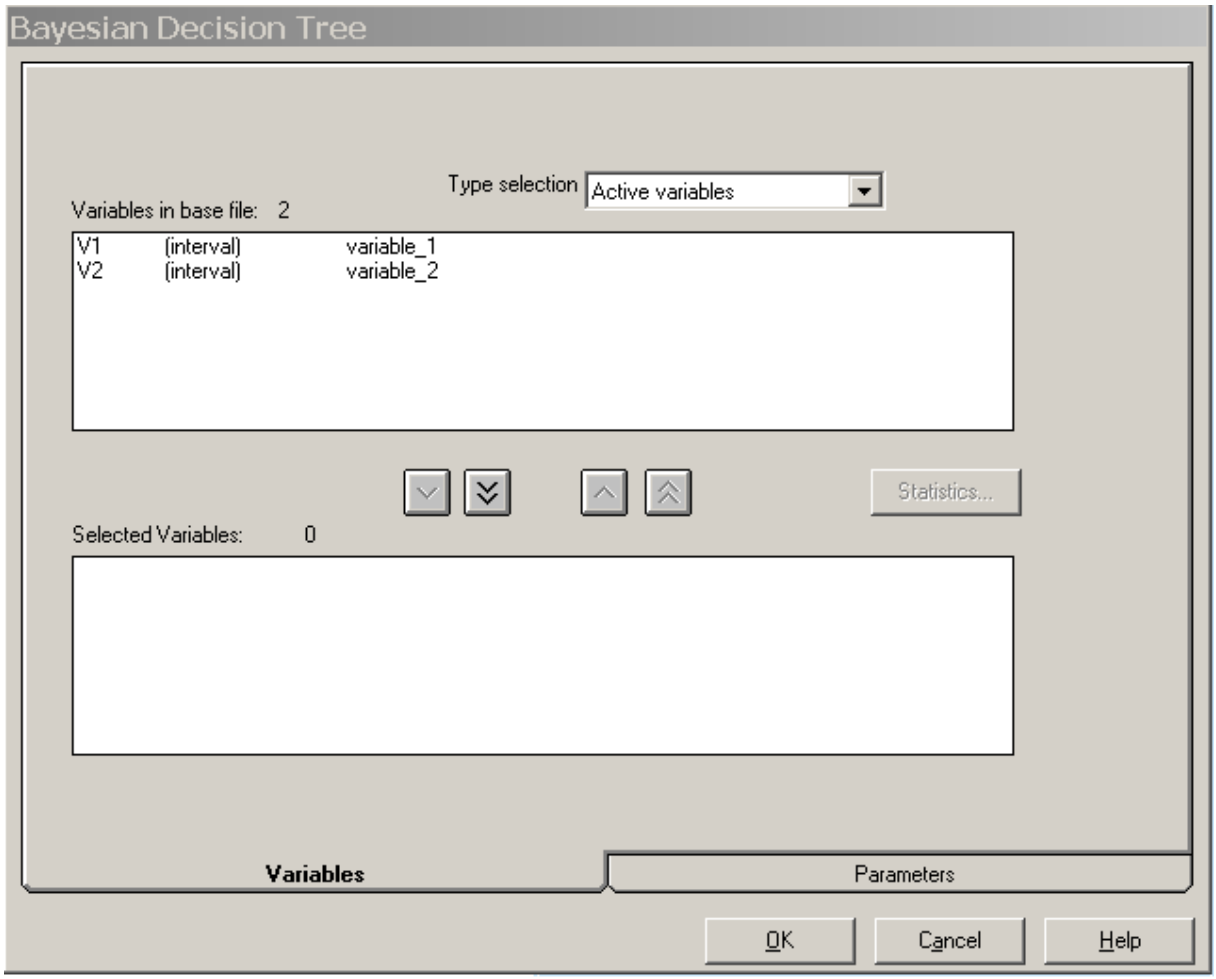


FIG. 2 The SODAS window for the selection of the active variables

The class identifier must be the categorical single variable in the dataset corresponding to the above description. Briefly, this is the variable which has two categories.

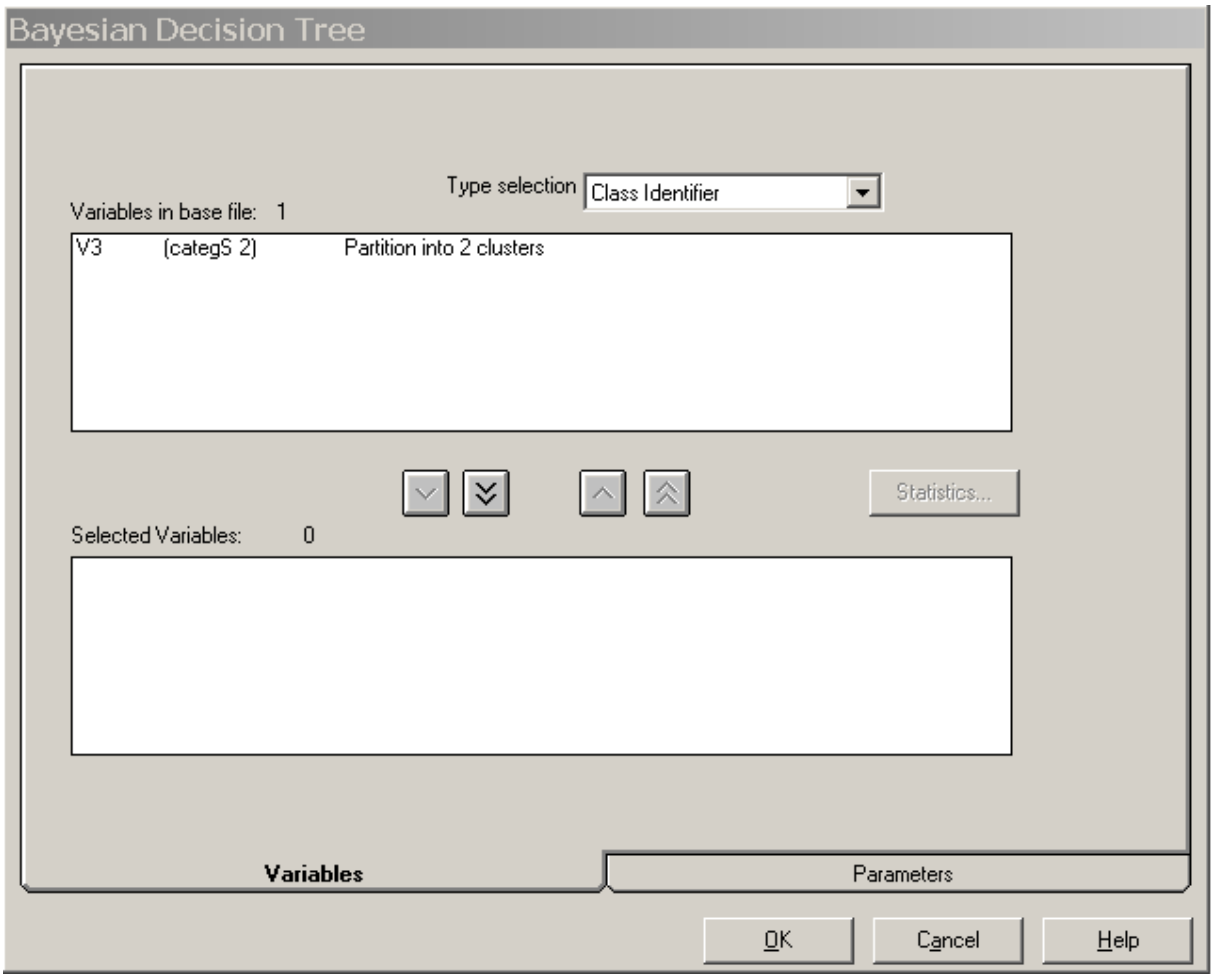


FIG. 3 The SODAS window for the selection of the class identifier

20.3.2 Parameters definition

Three parameters have to be defined:

- the value of the pruning parameter: it serves in the gap test, used during the pruning step. This value must be between 0 and 1. By default, it is set to 0.5.
- the minimum size to split the node: this parameter defines the minimum number of individuals in a node to split it. By default, this value is set to 3.
- the method to choose the prior probabilities: the user has to specify the prior probabilities for the bayesian rule. There are two possibilities: the probabilities can be based on the number of classes, or they can be based on the proportions observed in the training data sets. By default, they are computed in the first way.

In the artificial example, we have chosen the default values for the pruning parameter, that is 0.5, and for the minimum size to split a node, that is 3. We have also chosen the first way to compute the prior probabilities.

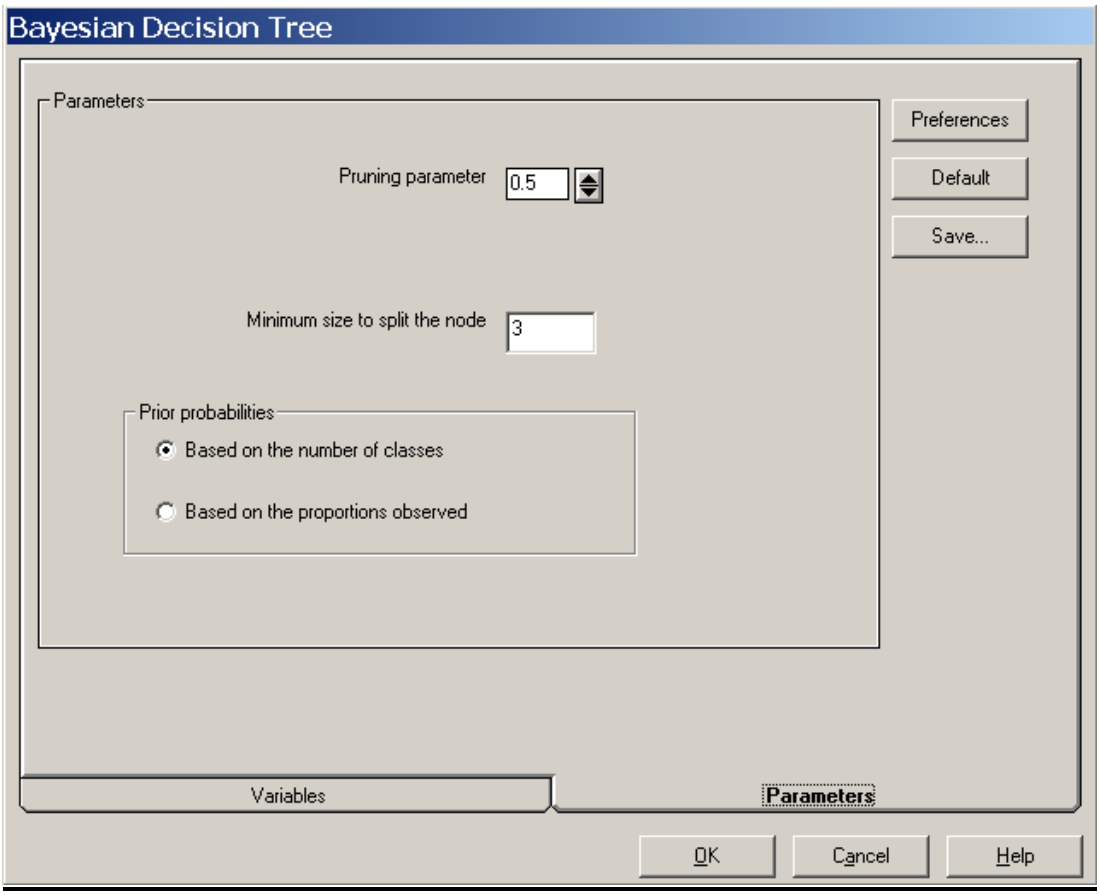


FIG. 4 The SODAS window for the selection of the parameters

20.4 Additional buttons

20.4.1 Preferences

The user is able to save his own default values for the parameters. He gets these parameters by pushing this button.

20.4.2 Default

If this button is pushed, all parameters will be set to their default values.

20.4.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

20.5 The output

After having defined the parameters and run the SBTREE method, there are two outputs:

- a listing file, containing the whole development of the method;
- a graphical representation of the tree corresponding to the discriminant analysis rule.

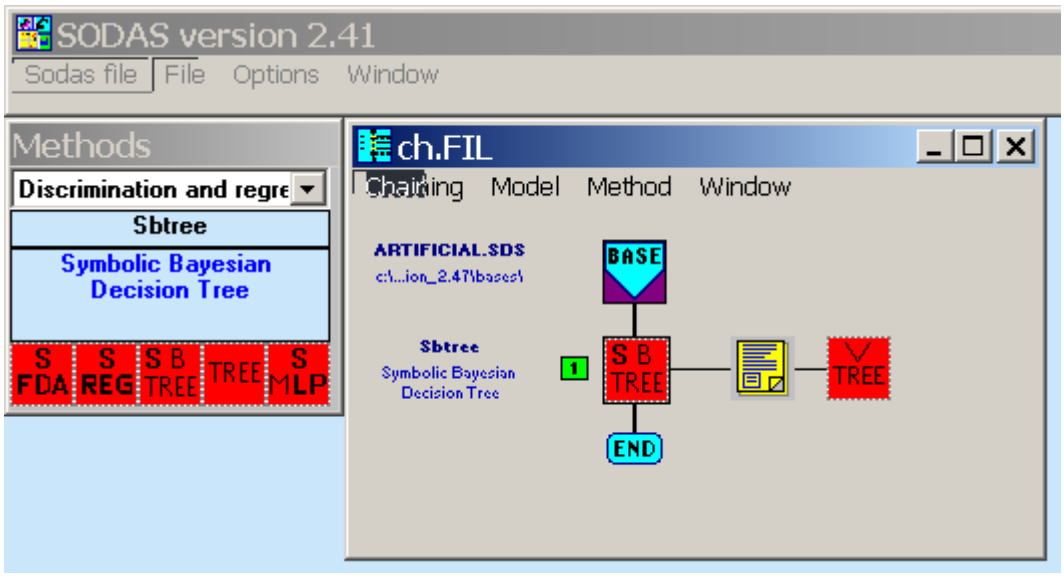


FIG. 5 The chaining after running the SBTREE method

20.5.1 The listing file

This file contains:

- The tree growing strategy. In fact, all the details of the construction of the bayesian decision tree, which leads to the symbolic discrimination rule. In our example, the beginning of the file is like this:

```
=========================
Split of the node :    1
=========================

Number of Symbolic objects in the node:     16pt
---------------------------------------

Criteria of cut :
-----------------

Cut variable :(  1) variable_1
Cut value :   6.00
Smoothing parameter CLASS 1 :   1.03
Smoothing parameter CLASS 2 :   0.95
Rule : if value of the SO <    6.00 -> the SO is in the left node (next even
       node)
```

```
        if value of the SO >    6.00 -> the SO is in the right node (next odd
       node)

Node :   2    Cardinal :     8pt
====================================
(0)  "artif_1"
(1)  "artif_2"
(2)  "artif_3"
(3)  "artif_4"
(4)  "artif_5"
(5)  "artif_6"
(6)  "artif_7"
(7)  "artif_8"

Node :   3    Cardinal :     8pt
====================================
(8)  "artif_9"
(9)  "artif_10"
(10)  "artif_1"
(11)  "artif_12"
(12)  "artif_13"
(13)  "artif_14"
(14)  "artif_15"
(15)  "artif_16"
```

- The symbolic discrimination rule: it is summarized at the end of the construction process

```
CLASSES:
--------

C1:     8 TRAINING OBJECT(S)
C2:     8 TRAINING OBJECT(S)
----------------------------
       16 TRAINING OBJECT(S)

     4 OBJECT(S) TO CLASSIFY


PRIOR PROBABILITIES:
--------------------

      C1: 0.500000
      C2: 0.500000

CUT RULE:
---------

CUT NUMBER   CUT VARIABLE   CUT VALUE
         1              1    6.000000
```

- The result of the symbolic discriminant analysis: at the end of the file the user can find the classification of the objects which were not assigned at a prior class:

```
CLASSIFICATION:
---------------

OBJECT        CLASS 1        CLASS 2
    16              0              1
    17              1              0
    18              0              1
    19              1              0
```

20.5.2 The graphical representation

In this representation, the user will find the bayesian decision tree. It shows clearly the nodes, the cut variables and their associated cut values, and the objects in each terminal node of the tree.

As far as our example is concerned, there is only one cut, and it is represented in the following figure:

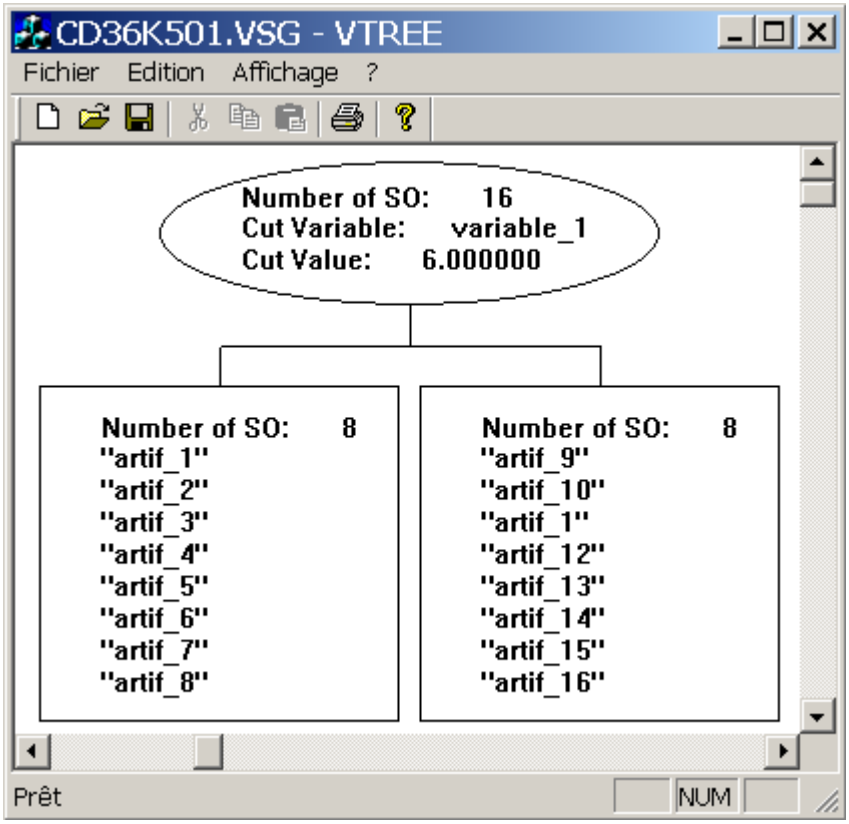


FIG. 6 Visualisation of the tree

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SFDA Help Guide

Factorial Discriminant Analysis

N. C. Lauro, R. Verde, A. Irpino, M. Guerra
DMS

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **24/06/2003**

21 SFDA : Factorial Discriminant Analysis

21.1 Introduction

The aim of the Factorial Discriminant Analysis on Symbolic Objects is to describe and visualize on factorial planes the relations between a set of predictors (*variables*) and a classificatory variable that identifies a partition of the objects into groups (*classes*), to define and validate a discriminant decision rule (*classification rule*) based on linear combinations of the predictor variables in order to classify new symbolic object into the membership class.

The SFDA method is based on a *numerical-symbolic-numerical* procedure that consist on a *numerical* transformation of the symbolic objects descriptors and a *symbolic* interpretation of the results.

21.2 The input to the SFDA.

The input for the SFDA procedure is a set of *n* objects described by symbolic variables (*predictors*) of different type: "quantitative single", "interval", "categorical single", "categorical multi-valued" and "modal".

The data matrix must contain a "categorical single" variable representing the classificatory variable that define the a priori partition of the objects into classes. This variable indicate the membership of the objects to one of the classes.

21.3 The output to the SFDA

As result, SFDA provide the coordinates of the vertices of the hyper cubes for the symbolic objects and classes on each factorial axis.

Others main results are: a table summarizing the absolute value, the percentage and the cumulative inertia explained by each axis, a distance matrix among objects computed according to the geometrical approach selected for the definition of the classification rule (Ichino-De Carvalho, PDI, Hausdorff, par. 4.4.1). If an automatic selection of active variables is performed, in the textual result file there will be a table synthesizing the results of the automatic selection based on Goodman Kruskal predictivity index $\boldsymbol{\tau}_{Xj}$

21.4 The options for SFDA.

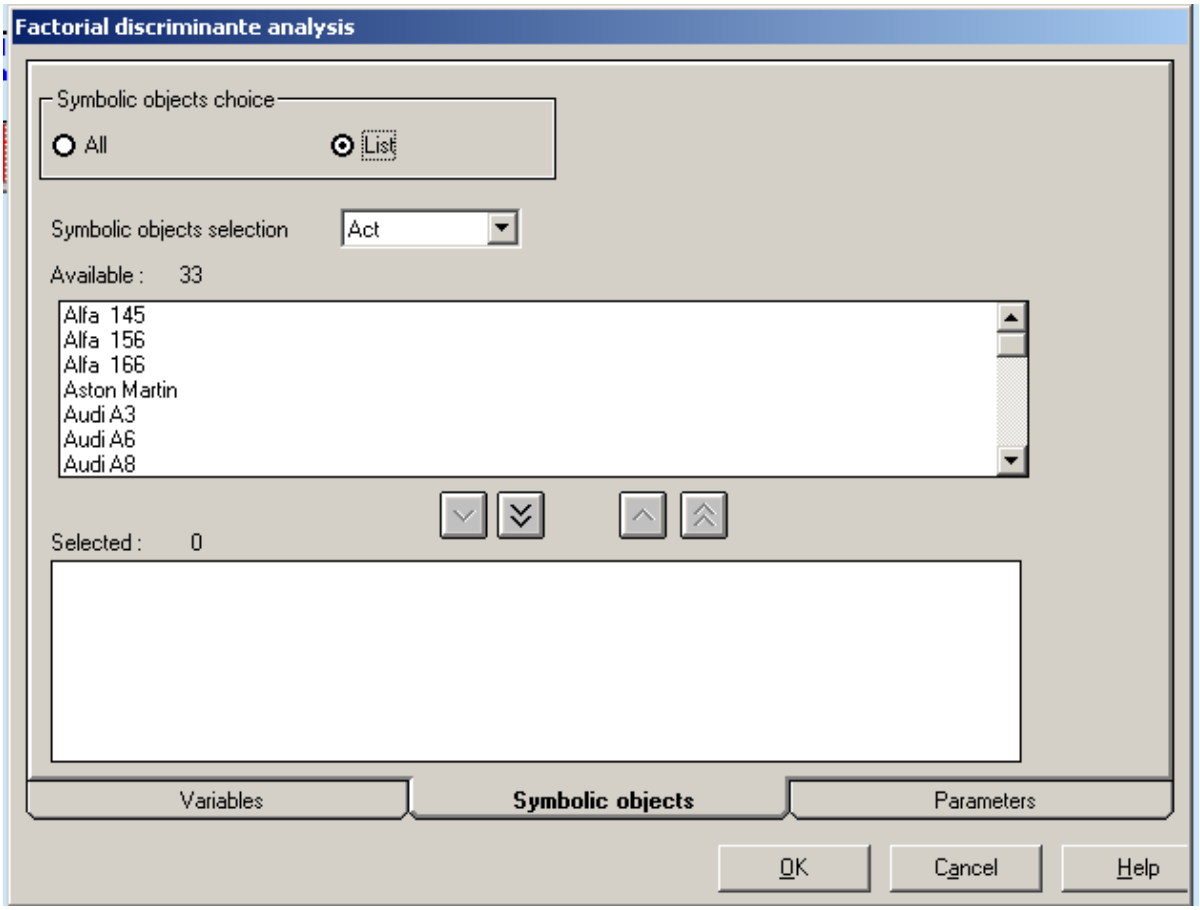
The SFDA module provides several options to perform the analysis. The user can specify the values of several different parameters, such that the classification rule resulting from the analysis is the better one in terms of higher correct classification ratio (CCR).

It is also possible, to run SFDA in automatic way without choosing any options for the parameters or selecting about it only a part, since SFDA provides default value for almost all parameters.

The SFDA module present four window, in which select options and parameters for the analysis: Symbolic Object", "Variables", "Rules" and "Parameters".

21.4.1 "Symbolic Objects" options.

The window "Symbolic objects" presents a drop list that allows to decide which symbolic object to select as Active or Illustrative for the analysis. In this case is provided a default value that select all objects as Active. All the Symbolic Objects selected as active compose the training set.

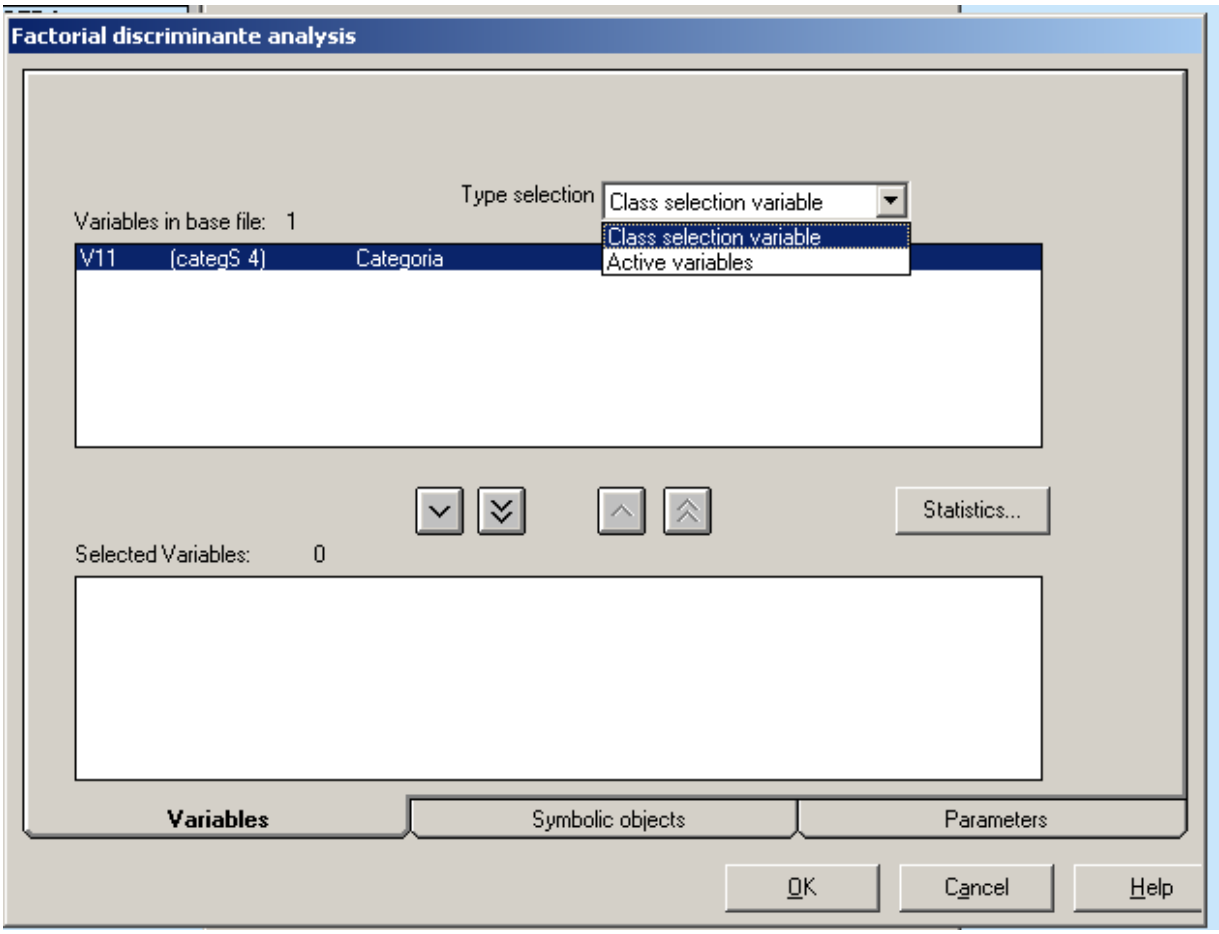


21.4.2 "Variables" options.

The "Variables" options window specifies which variables to use in the determination of the factorial discriminant axes. The selection is mandatory,; in this case it is not possible to opt for the default value.

The window "Variables" require the selection of the Active variables and the Class Selection variables. The variable selected as "Class Selection variables" must be a "categorical single" type variable and never an active variable.

For the selection of Active variable it is possible to select one to one the variables present in the list. Moreover, SFDA module provides a variables selection criterion performed by a generalization of Goodman-Kruskal predictivity index $\boldsymbol{\tau}_{Xj}$. By selecting "Active Variables" into the drop list, the "Automatic Selection" button is available, and therefore the list of selected variable is deactivated.



21.4.3 "Rules" options.

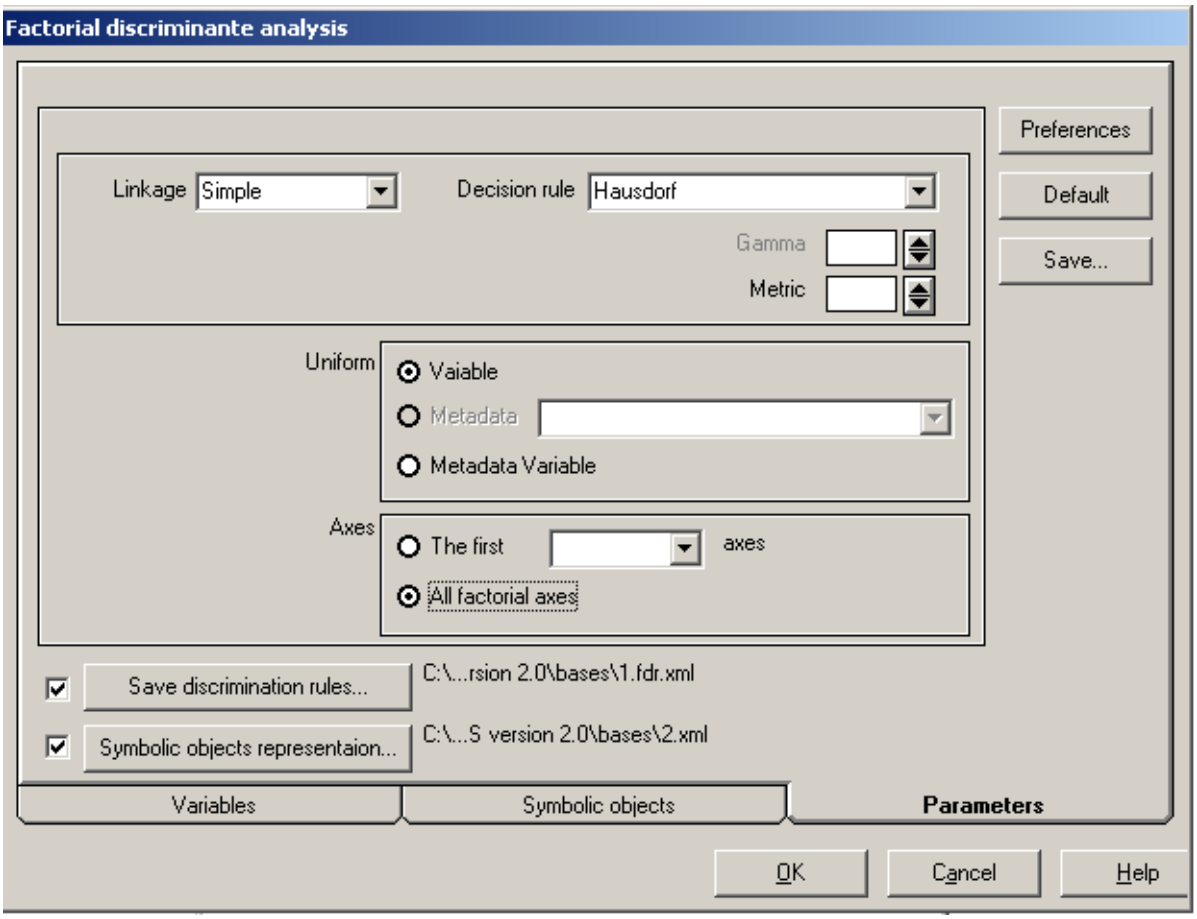
This window "Rules" allows to the user to select rules if they are presents in the file. The rules represents logical relationships between symbolic descriptors of the objects, and they can be Hierarchical Dependence rules, Logical Dependence rules and Taxonomies.

The SFDA module allows to the user to select a maximum of 5 rules as active, whereas by default no rules are selected.

21.4.4 "Parameters" options.

The "Parameters" window of SFDA looks as follows:

In this display it is possible to choose all the parameters needed for the method. The several options and parameters are explained below.



Decision Rule

In the box **"**Decision Rules**"** it is possible to choose one of the three different approaches for defining a geometrical classification rule based on particular type of distances defined between the images of the objects in the factorial plane. The resulting classification rule can be applied to classify either the symbolic objects selected in the analysis or a new set of symbolic objects in order to *predict* their membership to one of the a priori classes.

The three approaches to define the decision rule provided by SFDA module are:

1. **Ichino-De Carvalho**; this dissimilarity measure is a generalization of a dissimilarity measure proposed by Ichino & Yaguchi to the symbolic case by De Carvalho-Diday (1998).
2. **PDI**; this measure is based on the minimum increase of the potential descriptor of the image of the class in order to include a new object in its area.
3. **Hausdorff**; this measure is based on an extension of the Hausdorff distance between convex polygons to the symbolic data.

Gamma and Metric

These options allow to the user to tune the parameters of the Minkowsky dissimilarity measure. They are active only if the Minkowsky approach is selected in the "Decision Rules" box.

Gamma is a "weight" defined into the function $\varphi(S_{v\alpha}, S_{u\alpha})$ of the Minkowsky measure, its range is]0,1[(extremes excluded).

Metric require a integer bigger than 1 that define the metric to use for the dissimilarity measure (Metric = 1 defines a L_1–norm; Metric = 2 defines a L_2–norm (Euclidean); ……).

The default value for Gamma is 0.5, whereas for Metric is 2 (Euclidean metric).

Linkage

The options provided in the "Linkage" box are active only if "Ichino-De Carvalho" or "Hausdorff" distance approach are selected in the "Decision Rule" drop list.

In this box is possible to choose three different methods to determine the assignment of an object to a class, and they are:

- ✓ **Single**; if this option is selected, the dissimilarities between the object and every class is computed considering the minimum distance between the new object and each objects belonging to the class. The object is assigned to the class for which the dissimilarity measure is minimum.
- ✓ **Average**; if this option is selected, an object is assign to the class for which the average dissimilarity measure between the object to assign and each objects belonging to the class, is minimum.
- ✓ **Complete**; if this option is selected, , the dissimilarities between the object and every class is computed considering the maximum distance between the new object and each objects belonging to each class. The object is assigned to the class for which it presents the minimum the dissimilarity measure.

SO – Weights

In the **"SO-Weights"** box it is possible to assign a weights system to symbolic objects in order to take into account the different role of the objects in the analysis.
The alternatives for this option are:

- ➢ **Uniform weights**; all objects has the same weight
- ➢ **Variable**; the variable that represent weights need to be a "quantitative single" variable. The drop list contains only the "quantitative single" variables present in the file.
- ➢ **Metadata**; if a metadata it is present in the input file, it can be used as weights.

The default value for this option is the Uniform weight.

Axes

The **"**Axes**"** option provides the opportunity to choose a number of factorial axes smaller than the meaningful factorial axes. By default all factorial axes are selected

21.5 Additional buttons.

21.5.1 Preferences.

The user is able to save default values for the parameters. He gets these parameters by pushing this button.

21.5.2 Default.

If this button is pushed, all parameters will be set to their default values.

21.5.3 Save

The user is able to save his own default values for the parameters. If this button is pushed, the current values will be saved.

21.5.4 Save Discrimination Rule

This option allows users to choose an alternate name and path for the “discriminant rule”. The rule is stored in a SODAS file (with “.fdr” extension) which name can be defined by the user.

21.5.5 Save Symbolic Objects Representation

This option allows users to choose an alternative name and path to save the symbolic objects factorial coordinate file. This factorial coordinates are stored in a SODAS file (with “.sds or .xml” extension) which name can be defined by the user.

References

Bock, H. H. and Diday, E. (eds): 2000, *Analysis of Symbolic Data*, Springer-Verlag, Heidelberg.

Goodman, L.A.,Kruskal, W.H. (1954). Measures of association for cross-classifications. *Journal of the American Statistical Association* 49, 732-764.

Lauro, C., Verde, R., Palumbo, F.: 2000, Factorial Discriminant Analysis on Symbolic Objects, in Bock, H.H. and Diday, E. (Eds): 1999, Analysis of Symbolic Data, Springer Verlag. Heidelberg.

Lauro, C., Palumbo, F.: 2000, Factorial Methods with cohesion constraints on Symbolic Objects. In Proceeding of IFCS2000, Springer-Verlag, .

Lebart, L., Morineau, A. and Piron, M.: 1995, Statistique exploratorie multidimensionelle, Dunod, Paris.

Verde, R.: 1997, Symbolic object decomposition by factorial techniques. Indo-French Meeting, LISE-CEREMADE, Université Paris IX Dauphine.

Verde, R. and De Angelis, P.: 1997, Symbolic objects recognition on a factorial plan, NGUS'97, Bilbao Spain.

Verde, R. : 1999 Generalised Canonical Analysis on Symbolic Objects. In *Classification and Data Analysis, Theory and Application*. Vichi M. - Opitz O. (Eds), Springer-Verlag, Heidelberg, 195-202.

Verde, R. and Lauro, C.: 1993, Non symmetrical data analysis of multiway fuzzy coded matrices, ISI, Florence.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SDD Help Guide

Discriminant Description towards Interpretation

Edited by DAUPHINE

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **10/11/2003**

22 SDD : Discriminant Description towards Interpretation

22.1 Introduction

Marking and Generalisation by Symbolic objects (MGS approach) can be used for:

- building a new symbolic data table which summarise an initial given one
- interpretation aid in a symbolic clustering method
- interpretation aid in symbolic factorial method

In ASSO the corresponding procedure is named SDD, Symbolic Discriminating Descriptions.

In a first step, 'Marking Cores' are discovered to characterise a subset of statistical initial units using specific variables (the supervisor variable is the partition which is the indicator variable for the subset)

In a second step, the resulting descriptions are completed to generate a new data base which contains the whole set of initial variables.

Symbolic Discriminating Descriptions in ASSO are special symbolic descriptions of a subset: each of them characterises the subset by distinguishing it as well as possible from its complementary part in the whole set and the union of their extension elements fits as well as possible with the whole subset.

For the interpretation aids, the supervisor variable is

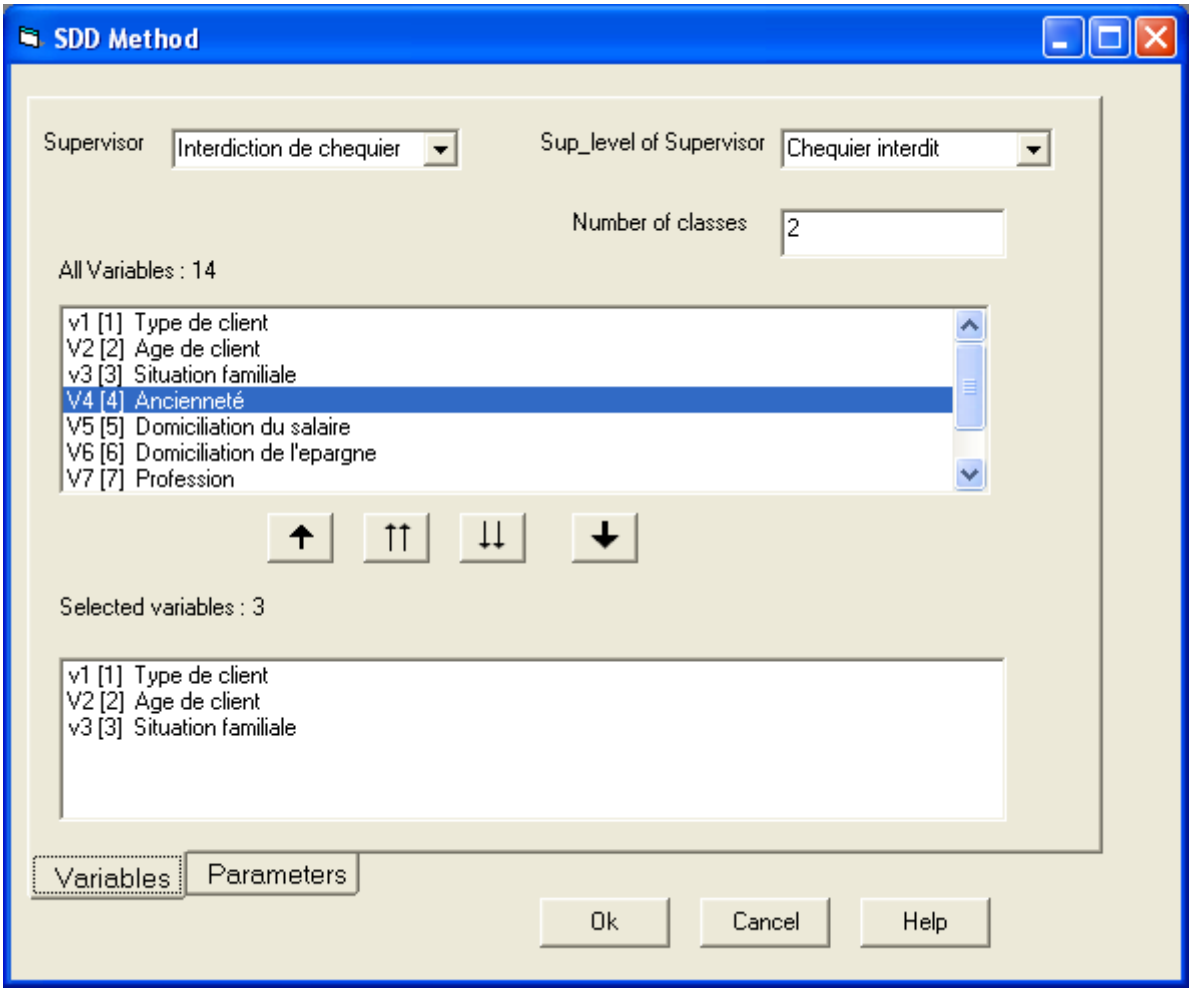
- the resulting partition for cluster analysis
- a binary partition of the elements (chosen by the end user) on a factorial plane, for factorial analysis.

22.2 The input to the SDD module

The input for the module SDD is a data array (any input format announced for ASSO procedures: ' .sds' , XML etc.) which can contain all the following types of data (if necessary in a mixed way) :

- quantitative single valued
- categorical single
- intervals
- categorical multi-valued

The window allows to select the supervisor variable in an interactive way, and the variables to take into account in the final descriptions.



22.3 The parameters of SDD

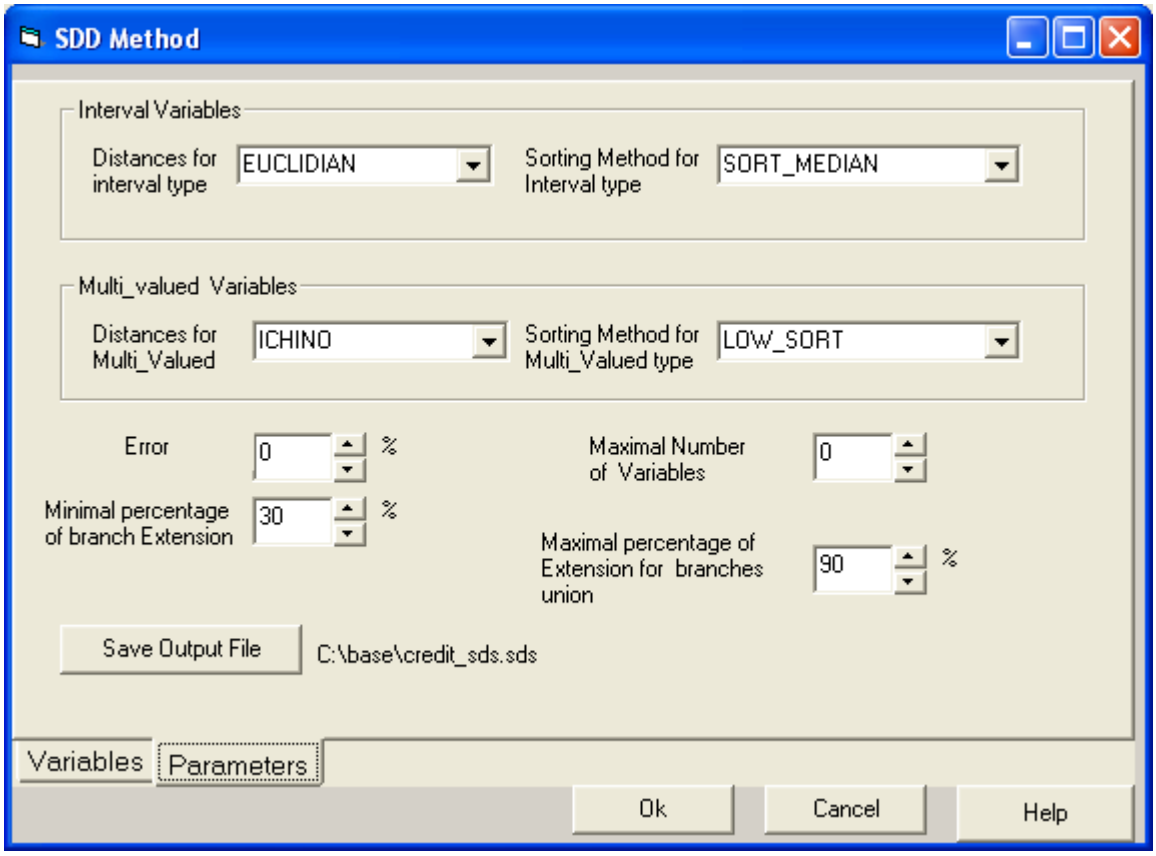
SDD provides several different options for treating data according to their type.

Some parameters are in common for all the possible situations. Some differ, e.g., by the definition of dissimilarities and the sorting methods, according to the type of data. Therefore, the user must specify the values of a variety of parameters which characterize special options. Insofar the user is able to tune the parameters such that the resulting method fits optimally his special application problem.

On the other hand, it is possible to run SDD in a automatic way because it detects by itself the different types of variables to be recoded and proposes in a single window the different optional parameters.

SDD provides default (standard) values only for some parameters, and replaces non-specified parameter values by standard values.

The parameter window of SDD looks as follows:



The options and parameters are described therby in the same order as they appear in the window. For a more complete explanation of the methods and the parameters please look at the specification of this module ("SDD, An outline with options and formulas").

The original SDD module applies upon categorical single variables. The interval variables and categorical multi valued variables are recoded through an implicit method.

For such a re-coding on intervals and categorical multi-valued variables, some parameters are to be chosen: dissimilarities and sorting method criteria. Continuous variables are transformed into categorical single ones by using the V-test criterion through dynamic programming.

22.3.1 Intervals variables

Distances of interval type

If the input data contain interval variables, it is essential to provide some parameters which appear in the previous window. The following different distance measures between n-dimensional intervals (hypercubes) can be selected:

- "khi2 distance"
- "Euclidian distance"
- "Kullback_leibler distance"
- "Bhattacharrya distance "

Sorting method of intervals type

Four different sorting methods are then available to represent the intervals:

- "Average sorting"
- "Average weighted sorting"
- "Median sorting"
- "Standard deviation sorting"
-

22.3.2 Multi valued variables

Distances for multi valued type

In this case, the multi valued variables are re-coded by using some specific parameters to be used during the dynamic processing of this kind of variables. The following four different distance measures can be selected :

- "Ichino distance"
- "Euclidian distance"
- "Gowda_diday distance"
- " Haussdorff distance "

Sorting method for multi valued type

Three different sorting methods can be chosen:

- "average sorting"
- "upper sorting"
- "low sorting"

22.3.3 Maximal Number of Variables

On a first phase, SDD runs on all the selected variables which verify the threshold that is chosen for the V-test criterion (default value is equal to 0 if all variables are selected at the very beginning of the process).

An option is available to limit the number of initial variables if the end user does not want to use the whole set of variables.

22.3.4 Minimal percentage of branch Extension / Error

The main elements to be defined in order to run the algorithm in ASSO are the following :

- The choice of a criterion to build branches from each selected node (in SDD: V-test)
- Two thresholds of acceptance (percentages) for two criteria (EXTension and ERRor for each node).

These two scores allow for checking the homogeneity in each branch and for distinguishing each branch from its complementary part. 'EXTension' is the percentage of elements that the description associated to a branch recognize, and 'ERRor is the percentage of 'wrong' elements (false positive) that belong to the extension of the description associated to a branch.

.

22.3.5 Maximal percentage of Extension for branches union

It is a parameter that can stop the developing of the algorithm: when the extension of the union of the developed branches reaches a certain percentage respect to the class to be described, the research ends.

22.3.6 Save partition base …

With this option, the set of marking cores, which have been completed with their probabilistic values on all the missing initial variables (produced after running the SDD module), is stored in a new SODAS file which name is to be defined in the window.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SREG Help Guide

Regression

Edited by DAUPHINE

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

23 SREG : Regression

23.1 Introduction

The module provides methods and tests for multiple linear regression on symbolic data like intervals, quantitative single, modals, taxonomies, mother-daughters variables, categorical single and multi-valued variables.

23.2 Input

The input of the method is a classical data matrix or a symbolic data matrix from a sodas file .sds. This input is a sequence of n individuals described by variables which could be of the following types :

- Explanatory variables can be quantitative single variables, intervals, modals, taxonomic, mother-daughters, categorical single and multi-valued variables.
- The Dependent variable can be quantitative single, interval or modal.

We present in figure 1 an example of the parameterisation file “XXXXXXX.PAD” that we propose :

```
SDS_IN = "D:\SREG\taxonomie.sds"
LOG = "D:\SREG\ taxonomie.log"                    =>  first section
OUT = "D:\SREG\ taxonomie.lst"

PROC SREG
REGRESSION

SELECT_VAR 2--8
DEPEND = 1
FS = 1
CLEVEL = 2                                              => second section
MTAXO = 2
PREDSDS = "D:\SREG\taxonomie.sds"
```

Figure 1 : example of a .PAD

The first section contains the full name of the input SODAS-ASSO base, the .LOG file and the .LST file.

The beginning of the second section contains the variables selection.

23.3 Variable selection

"SELECT_VAR 2--6" : enumerates the selected *explanatory* variables in the base defined in SDS_IN line. We have the following information :

- Default value : no, and, the user must select variables.
- Explanatory variables can be quantitative single variables, intervals, modal, categorical single, taxonomic, mother-daughters and categorical multivalued.

"DEPEND = 1" : gives the selected *dependent* variable in the base defined in SDS_IN line. We have the following information :

- Default value : no, and, the user must select a variable.
- This variable should not belong to the set of the *explanatory* variables selected above.

The Dependent variable can be quantitative single, interval or modal.

The figure 2 shows a simulation of the frame "Variable selection" :

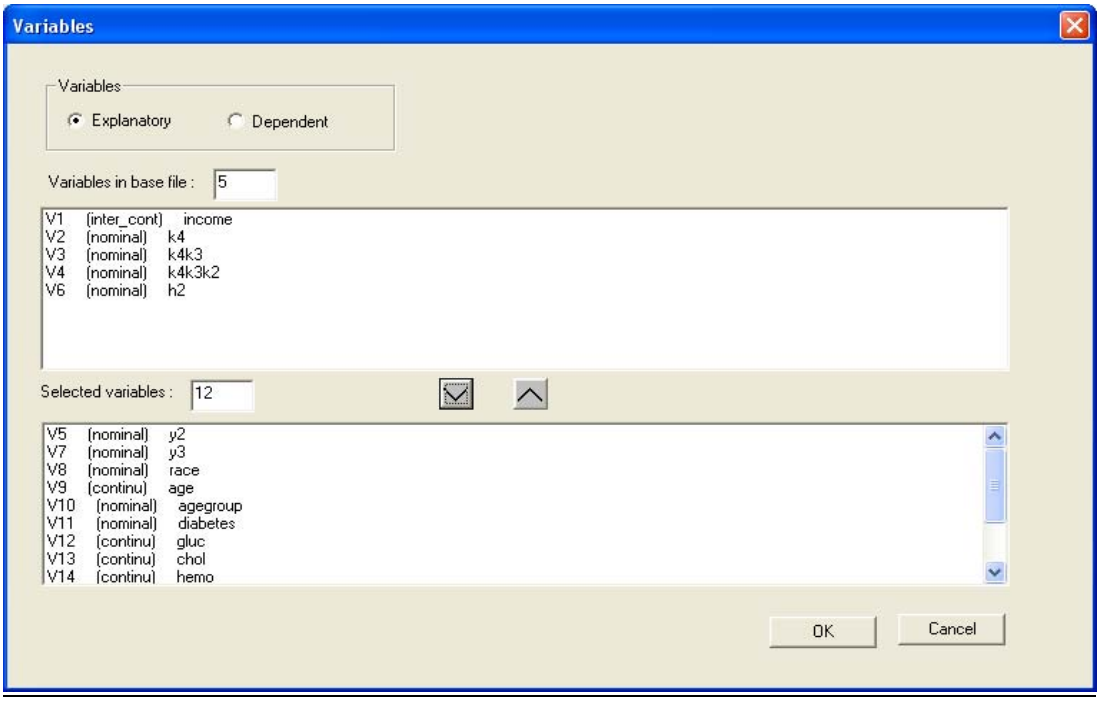


Figure 2 : The Variables

The last part of the second section of the parameterisation file "XXXXXXX.PAD" shown in figure 1 contains the parameters.

23.4 Parameters

The figure 3 shows a simulation of the frame “parameters selection”, we have 5 parameters :

“FS = 1” : corresponds to the choice between the Student test and the Fisher test for the test of the variables (values 0 or 1). The Student test is only for classical or symbolic quantitative variables. We have the following characteristics :
Student test correspond to the value 1 and fisher test correspond to the value 0.
The default value is fisher test (value 1).

“CLEVEL = 2“ : corresponds to the confidence level of the test, could be 0(95% default value) 1(99%) or 2(99,9%) :
the default value is 0.

“MTAXO = 2” : corresponds to the choice of the method of treating a taxonomy (decomposition, aggregation, divisive or multi-levels divisive method) :

- 4 integer values : 0 for the first method (decomposition) , 1 for the second method (aggregation) , 2 for the third method (divisive) and 3 for the forth method (multi-levels divisive).
- The default value is the forth method (value 3).

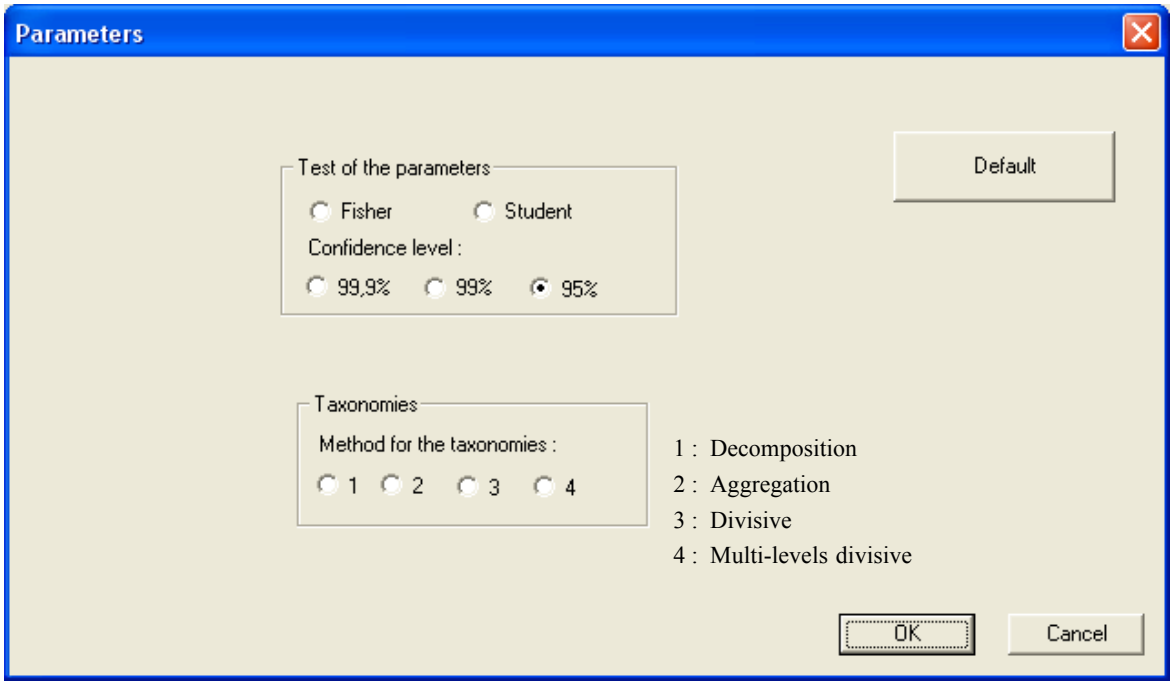


Figure 3 : The parameters

“PREDSDS = "D:\SREG\taxonomie.sds"” : correspond to the full name of the prediction SODAS-ASSO base.

23.5 Environment

The program works as a console application under windows and was developed on visual C++.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

SMLP Help Guide

Multi-Layer Perceptron

Edited by F. Rossi
DAUPHINE

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **28/11/2003**

24 SMLP : Multi-Layer Perceptron

24.1 Introduction

A multi-layer perceptron (MLP) is a particular type of neural networks. It is composed of a number of units connected together to form a networks. We define links between the units by weights.

The module SMLP is used to construct behaviour model from a set of examples. The neural networks method is constructed from a set objects described by inputs and output variables.

Given an initial specification matrix of weights, the goal of the method is to determine the optimum weights according a given criteria.

24.2 The input of the SMLP module

The input for the module SMLP is 2 sequences of variables .

The first sequence is composed of p input variables denoting the entries of neural networks.

The second sequence characterizes the target variable.

These variables can be numerical, interval valued, nominal, multi nominal or probabilistic valued variables.

SMLP module is able to process all type of variables, converting non numerical value to a numerical value using specific coding model.

Therefore, SLMP module accepts, in the window "variables", numerical, interval-type, nominal type, multi nominal type and probabilistic valued variables.

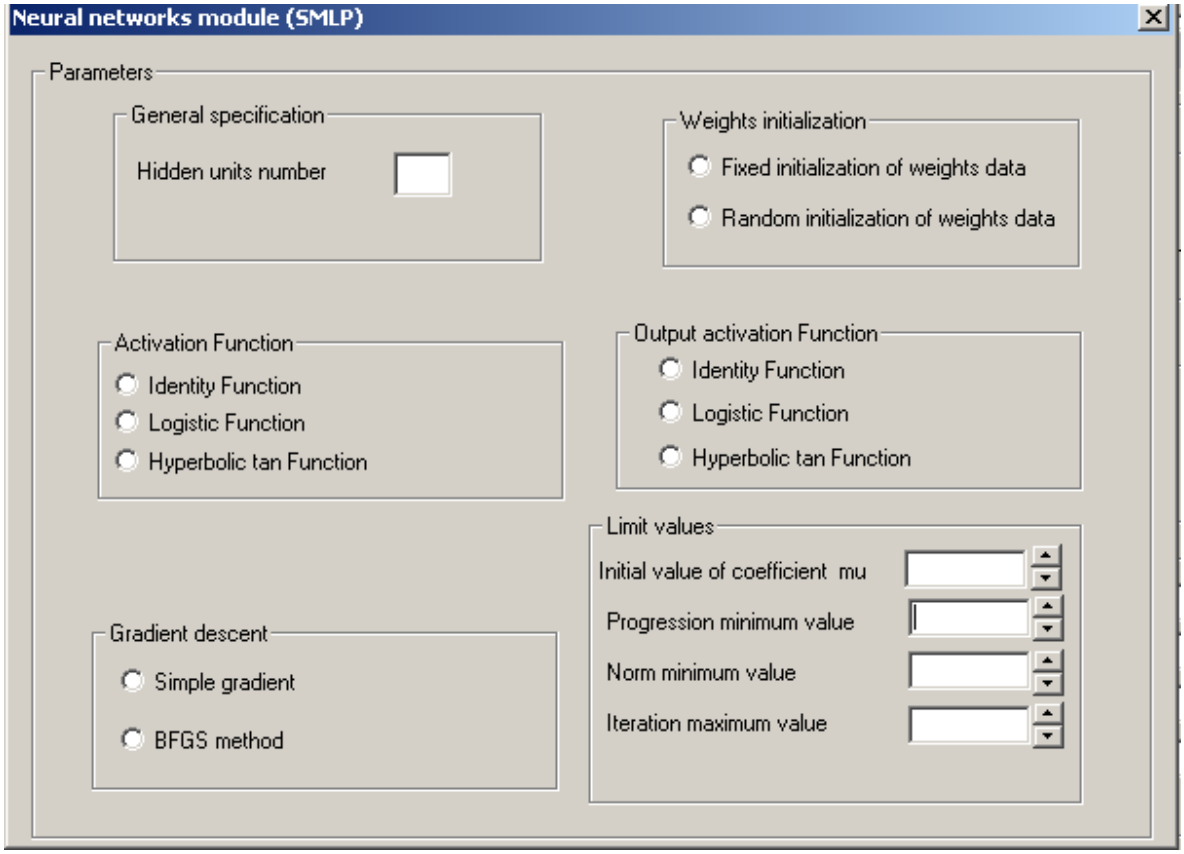
The appearance and the operation of the **window "variables"** is the same as in all other ASSO modules.

Remarks : In our program we do not deal with **"missing values".**

24.3 The parameters of SMLP

We give in the following window the parameters that the module SMLP needs to work. Their descriptions and correspondence with the PAD file is given after the parameters window.

The parameter window of SLMP module looks as follows:



General specification :

- Default value of hidden units number is *2*.

24.3.1.1 Weights initialisation

- Default value is *Random initialisation of weights data*.

Activation function :

- Default value is *Logistic function*

Output function :

- Default value is *Identity function*

Gradient descent :

- Default value is *BFGS method*

Limit values :

- Default value of initial coefficient mu is : 0.1
- Default value of progression minimum value is : 0.00001
- Default value of norm minimum value is : 2
- Default value of iteration maximum value is : 4

PAD PARAMETERS	Dialog name	Type/range
SELECT_X	Predictive variables selection	List of integers
SELECT_Y	Target variable selection	An integer
SELECT_WEIGHTS	Weights initialization	Integer : 1 if fixed initialization 2 if random initialization
NUMBER_HIDDEN_UNITS	General Specifications	An integer
ACTIVATION_FUNCTION	Activation Function	Integer : 1 if Identity Function 2 if Logistic Function 3 if Hyperbolic tan Function
OUTPUT_FUNCTION	Output Activation Function	Integer : 1 if Identity Function 2 if Logistic Function 3 if Hyperbolic tan Function
INITIAL_MU	Limit Values	Double
MIN_NORM	Limit Values	Double
METHOD_CHOICE	Gradient descent	Integer : 1 if Simple Gradient 2 if BFGS
NUMBER_ITERATION	Limit Values	An integer
MIN_PROGRESSION	Limit Values	A double
SDS_IN The base-file with data from variables of the variables (inputs and outputs)	NOT ON THE PANEL	A string quoted (between apices "")
TXT_IN The base-file of weights	NOT ON THE PANEL	A string quoted (between apices "")
LOG The base-file of all the error messages contained in the program	NOT ON THE PANEL	A string quoted (between apices "")
SDS_OUT The output base-file	NOT ON THE PANEL	A string quoted (between apices "")
LST_OUT The base-file of the general results of the program	NOT ON THE PANEL	A string quoted (between apices "")

VISUALISATION

Visualisation Exec Modules

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

VSTAR Help Guide

Zoom Star Visualisation

Edited by FUNDP

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

25 VSTAR : Zoom Star Visualisation

25.1 Introduction

The symbolic objects visualisation Module (see Figure 36) allows users to view in a table all symbolic objects present in a SODAS file and to perform graphical visualisation on individual SO. The application provides functionalities for viewing the graphical representations (2D and 3D), and the SOL representation of each symbolic object present in the table. We will present all functionalities menu by menu and item by item.

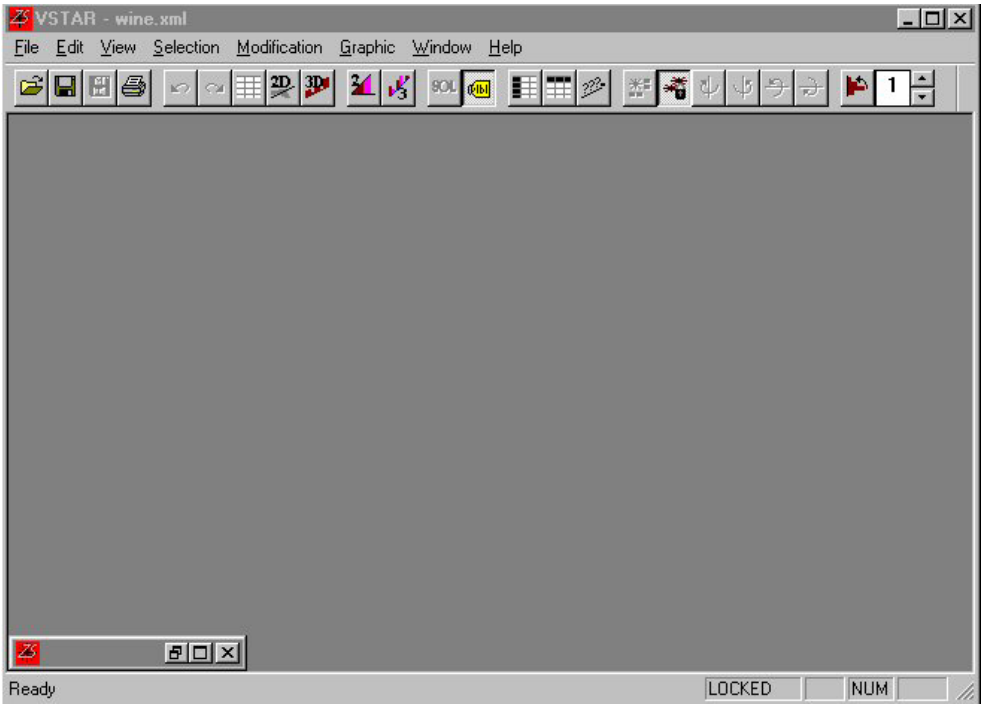
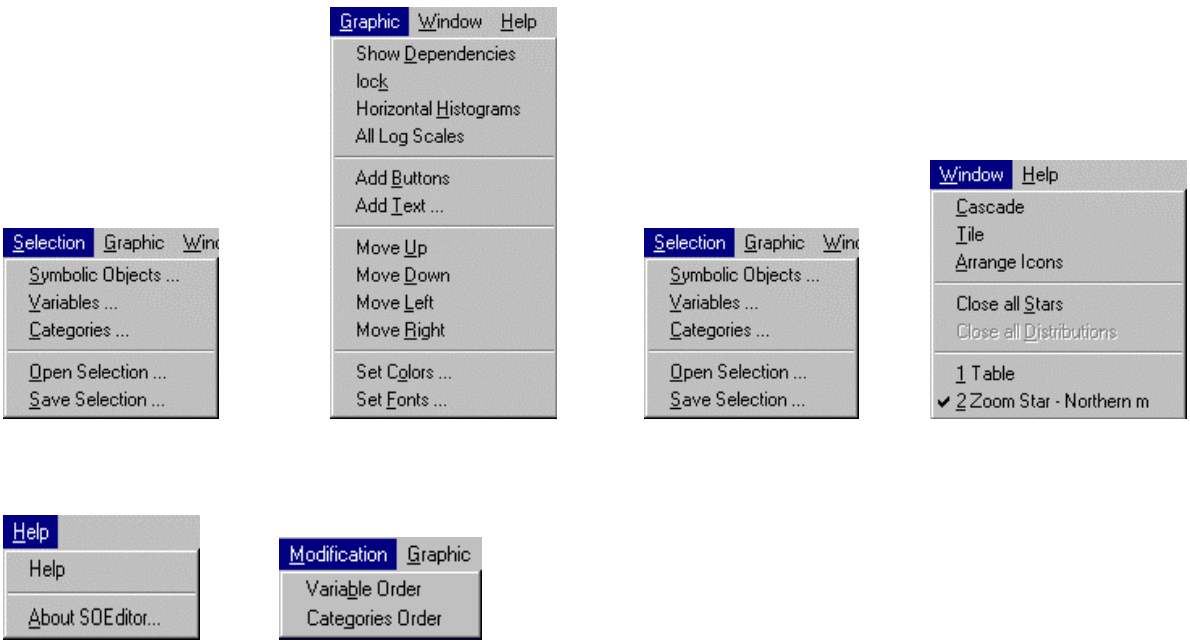


Figure 36: The Symbolic Data Visualisation



25.2 The «File» menu

The «File» menu corresponds to a classical Windows ™ File menu. It allows SODAS files to be opened (), closed and saved ().

The «Save» and «Save as ...» items save the content of the table in a sodas file. It implies that the symbolic objects and the variables present in the table will be the only one to be written in the file (in this case, a warning message will be given to the user), and the order defined in the table will be respected (rules and taxonomies are also updated according to the selected variables). If a new order has been defined for categories, this order will also be taken into account. The « Save » item is available only if the application is called outside of a chain.

It also allows the content of any window (Zoom Star, table, Los) to be printed (). The program checks if the selected printer is a colour printer. If not, colours providing distinct greys will be automatically selected by the program. These colours correspond to the colours recommended in the ASSO style guide. When printing a star, it is necessary to reduce the size of label characters.

For example, choose 5 for variable and categories labels.

The «Save Image File» item saves the Zoom Star or the table content in a printable format such as portable network graphics (*.png), graphic interchange (*.gif) or JEPG etc...

The «File» menu also contains the list of the last opened files.

25.3 The «Edit» menu

The «Edit» menu contains the «Undo» () and «Redo» () functions. These functions only apply to modifications done inside the table.

The «Copy» item allows the content of any window to be copied to the clipboard. So, the image can be paste inside any other application (word processor, image processor ...).

25.4 The «View» menu

In the VSTAR application, three distinct types of window are available. The table containing the Symbolic Objects and the variables located in the SODAS file, the graphical representation of a Symbolic Object (2D or 3D) and the SOL (Symbolic Object Language) description of a Symbolic Object. The first items of the «View» menu correspond to these different windows.

25.4.1 The «Table» item

The «Table» item sends the table window to the front, i.e. it displays the table when the window is hidden by other windows.

25.4.2 The «2D Graphic...» item

The «2D Graphic» item () is context sensitive:

- if the active window corresponds to the table, and if Symbolic Objects are selected inside the table (see 0), then the selection of this item will lead to the display of the 2D Zoom Stars of all selected objects. If variables are selected in the table, these variables will be represented on the graphical representation. If none of the variables is selected, all variables present in the table will be represented on the graphical representation.
- If the active window is a 3D Zoom Star representation, the 2D Zoom Star representation replaces the 3D graphic. If graphics are locked (see 25.6.3), all graphics are replaced by the 2D representation.

25.4.3 The «3D Graphic...» item

See 25.4.2, but for the 3D zoom Star representation ()

25.4.4 The Superimpose 2D & 3D items

The superimpose items display all the selected Symbolic Objects on the same picture. It is then possible to compare several symbolic objects by superimposition.

25.4.5 The «SOL» item

The «SOL» item displays the SOL description of all Symbolic objects selected in the table. The active window must be the table and objects have to be selected (see 0) to enable this item.

25.4.6 The «Taxonomy» item

The «Taxonomy» item displays the taxonomy corresponding to the active Zoom Star according to the Windows [TM] hierarchy representation (see Figure 37). The active window must be a graphic to enable this item.

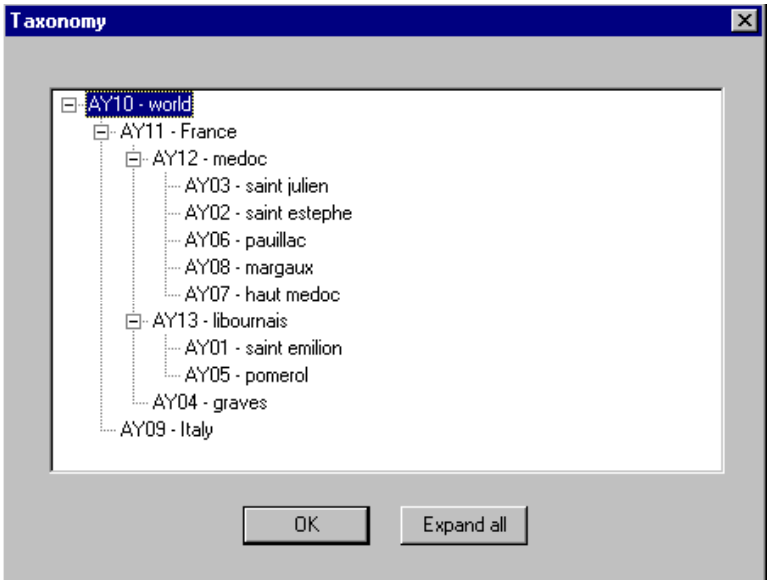


Figure 37: Representation of Taxonomy

In the «Taxonomy» dialog box, the «Expand all» button displays the complete taxonomy.

25.4.7 The «Tool bar» and «Status bar» items

The «Tool bar» and «Status bar» items allows display/hide respectively the status bar located in the bottom of the screen and the tool bar containing all shortcut icons.

25.4.8 The «Labels ...» item

The «Labels...» item makes appear a dialog box (see Figure 38) which provides the user with the opportunity to work with the first letters of labels instead of the identifiers for Symbolic Objects, variables, and categories. The maximum length of labels can be defined for each element. The chosen options are saved.

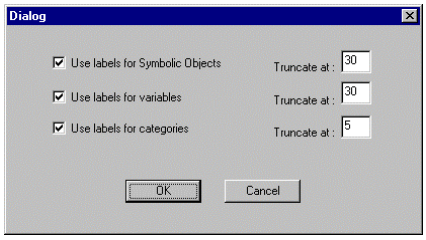


Figure 38: The "Labels" dialog box

A button located in the tool bar () can be used to switch from the identifiers to the labels. This button acts on symbolic objects, variables and categories in the same time.

25.5 The «Selection» menu

The «Selection» menu allows the user to select the Symbolic Objects, the variables and the categories that he wants to see in the table and on the Zoom Stars. It also allows a configuration to be saved/loaded. The selection is made by way of a dialog box containing three «tags». The first one corresponds to Symbolic Objects (see Figure 39), the second one corresponds to variables (see Figure 40), and the third one corresponds to Categories (see).

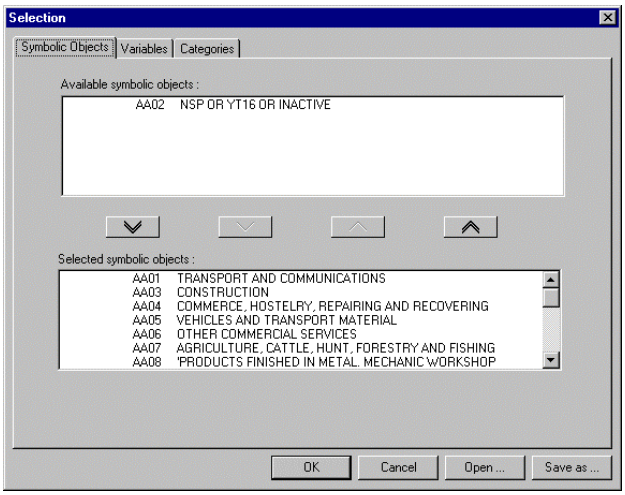


Figure 39: Symbolic objects selection

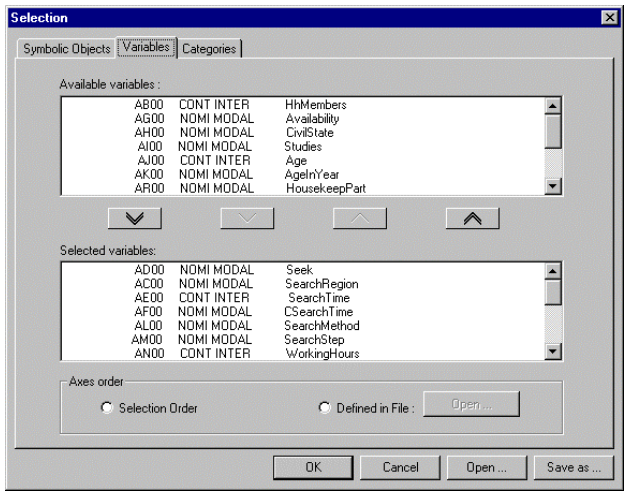


Figure 40: Variables selection

In the case of symbolic objects and variables, each «tag» contains two lists. The list located on the bottom of the «tag» contains the selected elements. Elements can be added and/or removed by using the four buttons. The order defined in the list will be respected in the table. On the bottom of the dialog box, there are two buttons which can be used to open and to save a configuration.

The configuration corresponds to choices made for symbolic objects, variables and categories (i.e., it is not possible to save the symbolic objects selection and the variables selection in two distinct files).

The «Tag» corresponding to the variables selection has an additional element which gives the opportunity to use the order defined in the list, or to use the order defined in a file generated by a statistical method (this possibility is not yet implemented).

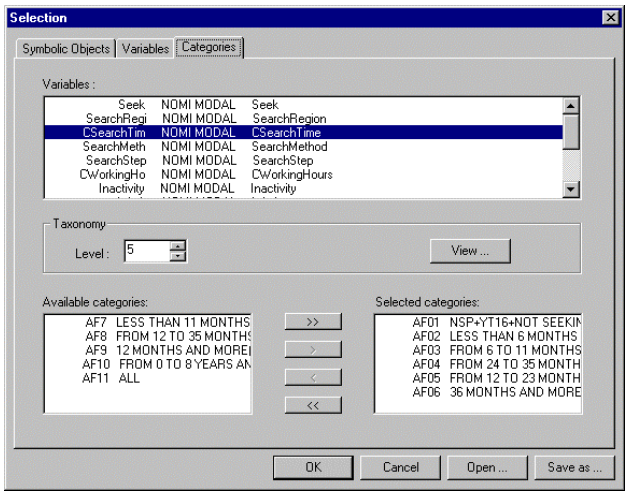


Figure 41: Categories selection

In the «tag» corresponding to categories selection, all categorical variables which are currently selected are placed in the list located on the top, and the selected categories are located in the lower right list. Categories, of the selected variable, can be added and/or removed by using the four buttons.

The order defined in lists will be respected on the Zoom Star. If taxonomy is defined for a variable, the user can select the level he wants to work with. The complete taxonomy is displayed when the «View ...» button is pushed.

For the star display, the number of variables and categories for any configuration must be such as:

$$3 \leq \text{number of variables} \leq 500$$

$$1 \leq \text{number of categories} \leq 500$$

Nevertheless it is recommended, for visual quality, to use not more than 24 variables and 15 categories.

25.5.1 The «Select So...» item

The «Select So...» item ( ) displays the selection dialog box and activates the «tag» corresponding to Symbolic Objects selection (see Figure 39).

25.5.2 The «Select Variables...» item

The «Select Variables...» item ( ) displays the selection dialog box and activates the «tag» corresponding to variables selection (see Figure 40).

25.5.3 The «Select Categories...» item

The «Select Categories...» item ( ) displays the selection dialog box and activates the «tag» corresponding to categories selection (see).

25.5.4 The «Open Selection...» item

The «Open Selection...» item displays a dialog box which allows a file containing a configuration to be opened. This file has a «.ovc» extension.

25.5.5 The «Save Selection...» item

The «Save Selection...» item displays a dialog box which allows the current configuration to be saved in a file. This file has a «.ovc» extension.

25.6 The «Graphic» menu

25.6.1 The «Show/Hide Dependencies» item

The «Show/Hide Dependencies» item displays/hides dependencies. Axes for the dependent variable is attached to the "mother" variable. If graphics are locked (see 25.6.3), dependencies are displayed/hidden on all graphics.

25.6.2 The «Add/Remove Buttons» item

The «Add/Remove Buttons» item () displays/hides, on all graphics, buttons allowing Zoom Star to be moved up, down, right and left (see Figure 7).

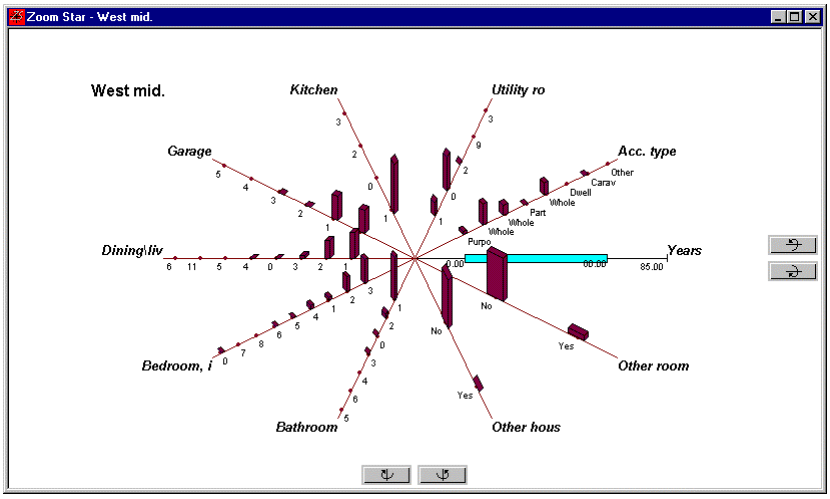


Figure 42: Zoom star with buttons allowing Zoom Star to be moved

25.6.3 The «Lock/Unlock» item

The «Lock/Unlock» item () allows a function to be applied to a single Zoom Star (if unlocked) or to all Zoom Stars currently displayed (if locked). This locking system applies to the «2D graphic...», «3D graphic...», «Show/Hide dependencies», «Add Text ...» items and all move functions. It also applies to visualisation of a distribution, i.e. if the user selects an axis corresponding to a variable with weighted values when graphics are locked, the variable distribution corresponding to all Zoom Stars will be displayed. The status bar continuously indicates if graphics are locked or not.

25.6.4 The «Horizontal/Vertical Histograms» item

This item () can be used to switch from the horizontal to the vertical representation of histograms.

25.6.5 The «All Log Scales» item

The «All Log Scales» item allows the user to ask for the use of logarithmic scales for all quantitative variables. Of course, the logarithmic scale will be applied only if the minimum value is greater than 0. Logarithmic scales can be selected to any axis individually (see 25.10.8).

25.6.6 The «Add Text ...» item

The «Add Text...» item provides the user with the possibility to add a text to a graphic. Once the user has entered the text and pushed on the « Ok » button of the dialog box presented on Figure 43, the text is displayed on the upper-left side of the graphic. The user can then drag the text (by pushing on the left button of the mouse) and drop it at the desired location (by releasing the left button of the mouse). To modify or to remove a text, the user has to double click on the text and the dialog box presented on Figure 43 will be displayed. The dialog box also contains a «Font...» button which allows the user to select a particular font for each text.

If graphics are locked (see 25.6.3), the text is displayed/moved/removed on all graphics in the same time.

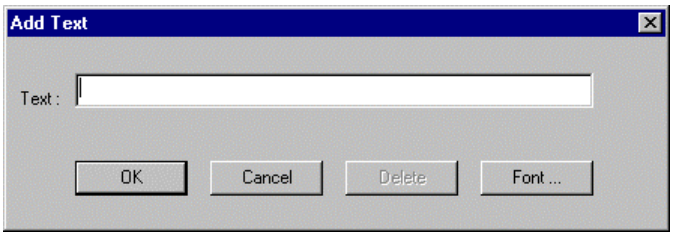
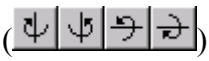


Figure 43: The Add Text dialog box

25.6.7 The «Move Up», «Move Down», «Move Left», «Move Right» items

The move items () allow Zoom Stars to be moved around a vertical and a horizontal axis. However, the four arrow keys can also be used and it represents of course the easiest way to move graphics. If graphics are locked (see 25.6.3), each move is applied to all graphics.

25.6.8 The «Set Colours...» item

The «Set Colours...» item makes appear a dialog box (see Figure 44) allowing Zoom Stars colours to be modified. The selected colours are saved when the application is closed.

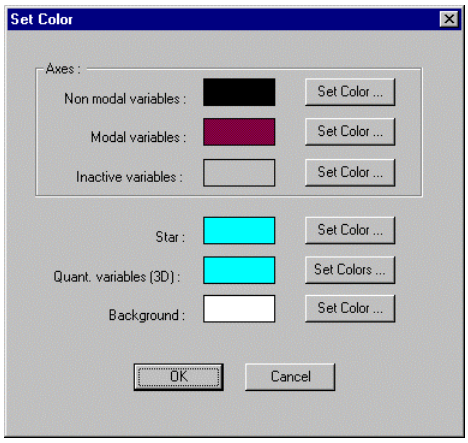


Figure 44: The "Set Colours" dialog box

25.6.9 The «Set Fonts...» item

The «Set Fonts...» item makes appear a dialog box (see Figure 45) allowing Zoom Stars fonts to be modified. The selected fonts are saved when the application is closed.

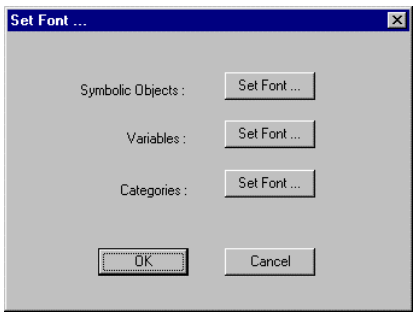


Figure 45: The "Set Font..." dialog box

NB: When printing a star , it is necessary to reduce the size of label characters.

For example, choose 5 for variable and categories labels.

25.7 The «Window» menu

The «Window» menu is a standard Windows™ menu making the windows managing easier.

25.7.1 The «Cascade» item

The «Cascade» item displays windows the one behind to others (see Figure 46).

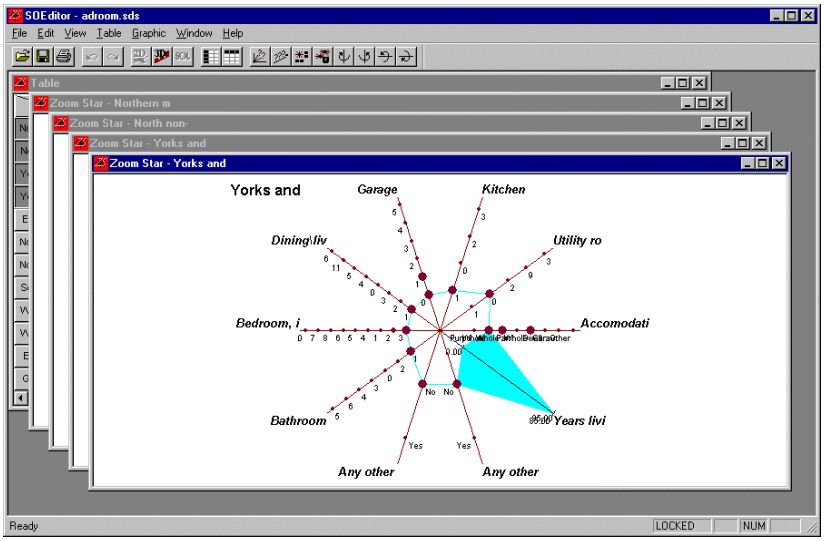


Figure 46: Example of windows in cascade

25.7.2 The «Tile» item

The «Tile» item displays windows so that they are all completely visible (see Figure 47).

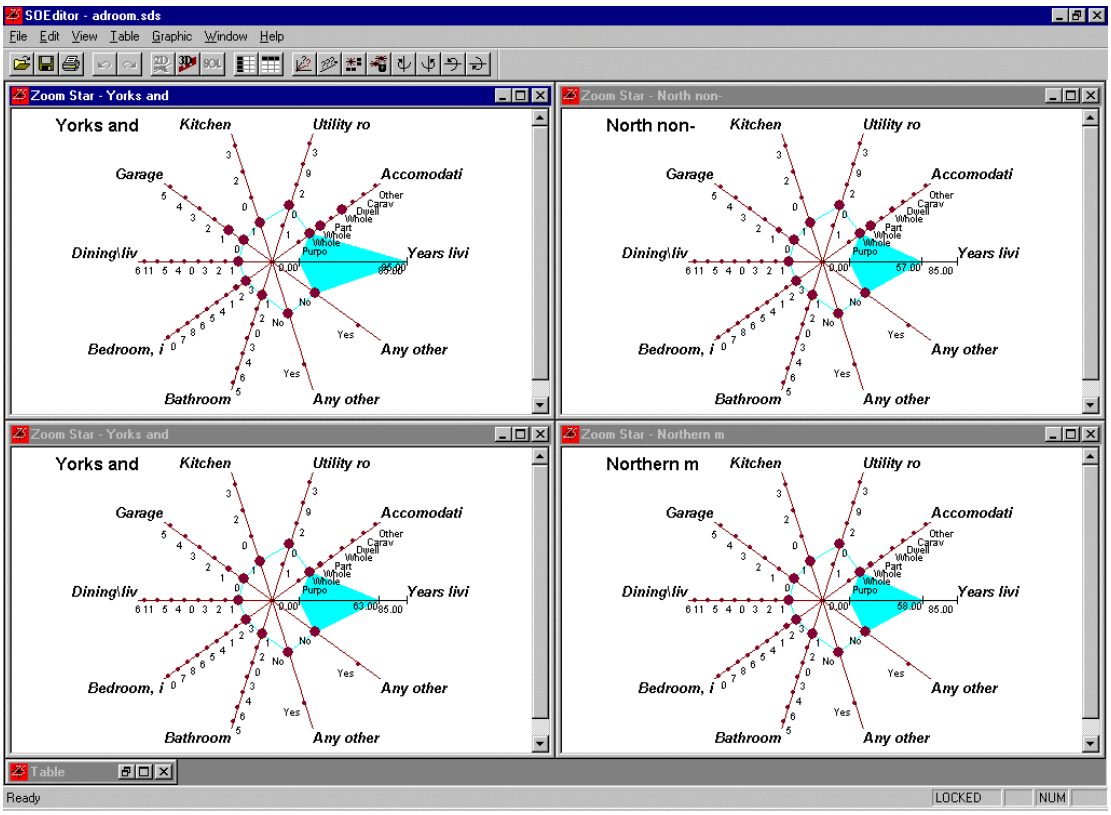


Figure 47: Example of tiled windows

25.7.3 The «Arrange Icons» item

The «Arrange Icons» item displays side by side minimised windows in the bottom of the screen (see the «table» window on Figure 47).

25.7.4 The «Close All Stars» item

The «Close All Stars» item closes all windows containing Zoom Stars.

25.7.5 The «Close All Distributions» item

The «Close All Distributions» item closes all windows containing a distribution.

25.8 The «Help» menu

The «Help» menu contains the reference to the on-line help and an «about» dialog box.

25.9 The table

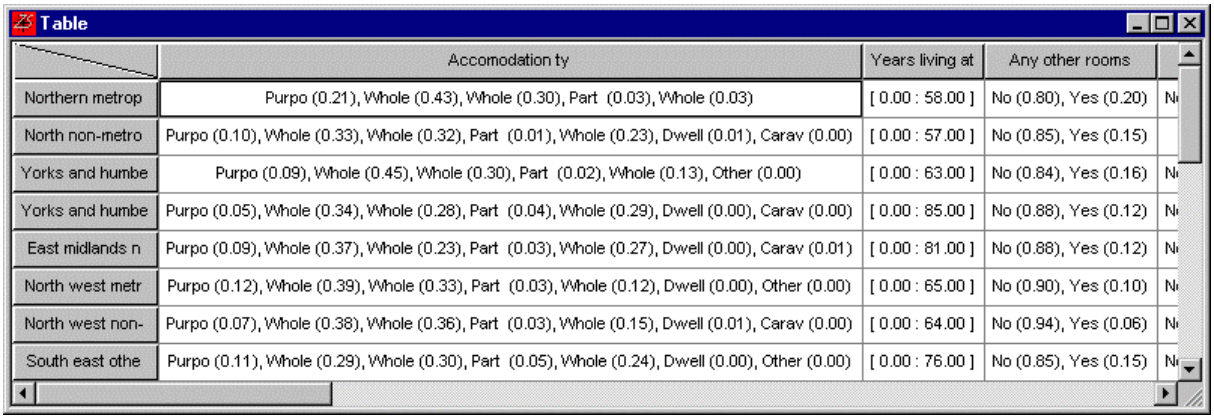


Figure 48: Table representation

The user has the possibility to scroll vertically and horizontally thanks to scroll bars or arrow keys, and to modify column widths by «dragging and dropping» a column with the mouse (in this case, the mouse icon changes into ↔).

Selection of lines and columns

It is possible to select lines and/or columns by selecting the header cell (grey cell) with the mouse. The entire line or column is coloured in grey as shown on Figure 49. The upper-left cell is separated in two parts; the upper-right part allows all variables to be selected/deselected, while the lower-left part allows all SOs to be selected/deselected.

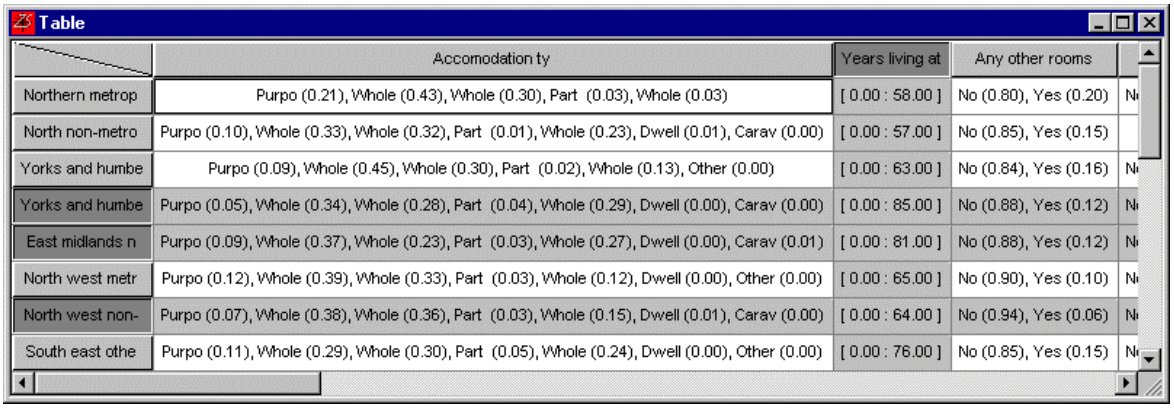


Figure 49: Selection of lines and columns

25.10 Visualising the graphical representation

25.10.1 The Zoom Star representation

The following table describes the conventions used for axes representation:

Variable type	Axis description
Quantitative	Graduated axis
Categorical	Dots equally distributed on the axis
Not weighted	Axis drawn in black
Weighted	Axis drawn in claret
Not applicable	Axis drawn in grey

Remark: colours given here correspond to default colours.

25.10.2 The 2D Zoom Star

In the 2D Zoom Star, axes are linked according to each variable value. The following table describes the conventions used to link all axes according to the value which is illustrated on Figure 50:

Variable value	Link type	Example of variable
Single	the current value is linked	Sex
Multiple	each values are linked	Action
Interval	the limits are both linked and the whole surface is filled	Age
Weighted values	The value with the highest weight is the only one to be linked	Place

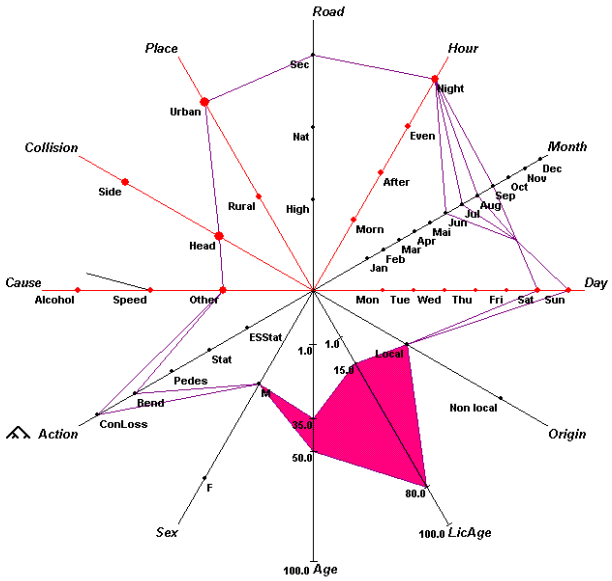
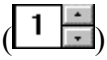


Figure 50: Example of a 2D Zoom Star

25.10.3 The 3D Zoom Star

On the 3D Zoom Star representation, distributions corresponding to each variable with weighted values are added, as show on Figure 51 (this figure represents the same object as Figure 50). In this representation, axes are not linked. Therefore, the conventions are not the same to represent variables values. In this case, values are written in claret.

The height of histograms represented on the 3D Zoom Stars can be chosen by the user thanks to the small arrows located in the tool bar (). The edit box located beside the small arrows indicates the current level. The level ranges from 1 to 10.

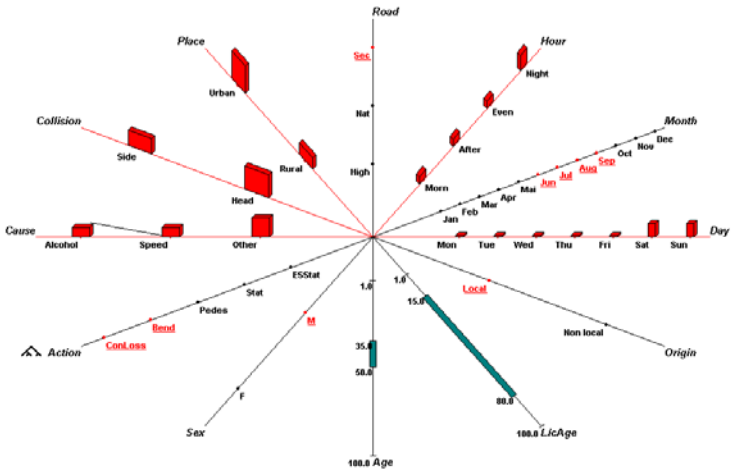


Figure 51: Example of 3D Zoom Star

25.10.4 Moving the object name and the variables names

The user has the possibility to move the object name (located in the upper-left corner by default) and any variable name by a «drag and drop» method.

25.10.5 Visualisation of distributions

When the mouse points on an axis corresponding to a variable with weighted values, the mouse icon changes (Figure 52a). This indicates to the user that he can select the axis (with the left button of the mouse) in order to view the distribution in another window. This possibility is available for both the 2D and the 3D representations.

If graphics are locked (see 25.6.3) , the distribution of the selected variable are displayed for all symbolic objects represented by Zoom Stars.

25.10.6 Visualisation of dependencies

When the Zoom Star is first displayed, dependencies are not completely displayed in order to avoid screen overloading. A small line is only displayed to indicate the existence of a dependency (see «Speed» category on «Cause» axis on Figure 51). When the mouse points on this small line, the mouse icon changes (Figure 52b). If the user pushes on the right button of the mouse, the entire axis is drawn. Once the entire axis is drawn, the user can push again on the right button of the mouse in order to replace the axis with the small line.

25.10.7 Visualisation of taxonomies

A small icon representing taxonomy is placed beside any axis name corresponding to a variable with taxonomy. When the mouse points on this icon, the mouse icon changes (Figure 52c). This indicates to the user that he can select the icon (with the left button of the mouse) in order to view the taxonomy in another window.

25.10.8 Scales modification

For quantitative variables, it is possible to modify the bounds of the displayed interval. When the mouse points to a quantitative axis, the mouse icon is modified to indicate that the axis can be selected (see Figure 52d).

Figure 52: Mouse icons

When the user selects a quantitative axis, a dialog box is displayed (see Figure 53). This dialog box allows the user to modify the minimum and the maximum values, but also to use a logarithmic scale (the minimum value must be greater than 0 to allow the use of the logarithmic scale). If the user extends the bounds of the interval, the new values will be taken into account when the sodas file will be saved. If the user restricts the interval, the original

values will be taken into account when the sodas file will be saved (because this could lead to a loss of information).

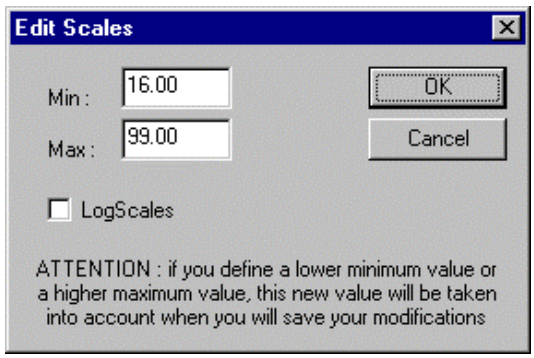


Figure 53: The "Edit scale" dialog box

25.11 Breakdown or Drill down

Breakdown or Drill down is a new interactive way to create symbolic object inside a given class. When selecting a category on one axis of a zoom star, a search in the individuals data base is generated and the corresponding object is automatically created. It represents the set of individuals who have the selected characteristic. For example if we import the data on sociological investigation, we can interactively create the objects corresponding to gender (women or men), or the class of individuals corresponding to status (single person, widower, cohabitation, divorced). The two symbolic objects will be displayed on the same graph (superimposition).

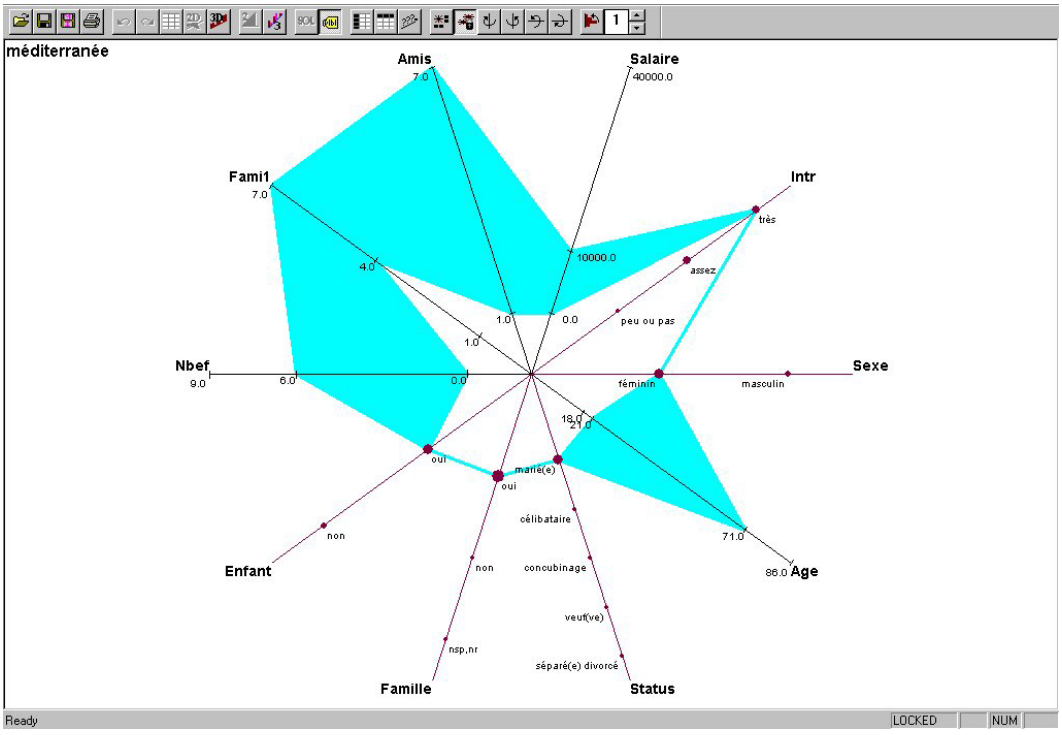


Figure 54: Original object for breakdown

To perform a Breakdown, the following sequence of actions has to be done:

1. Display the desired object in 2D/3D (Figure 19).
2. Click with the right button on the desired category of modal variable (for example “Masculin” on “Sexe” variable).
3. A window appears where it is asked to choose the original data base (Figure 20).
4. The new symbolic object is automatically created and represented by superimposition process.
5. It is added in the original file but saved with another name (BkDown + Original name).

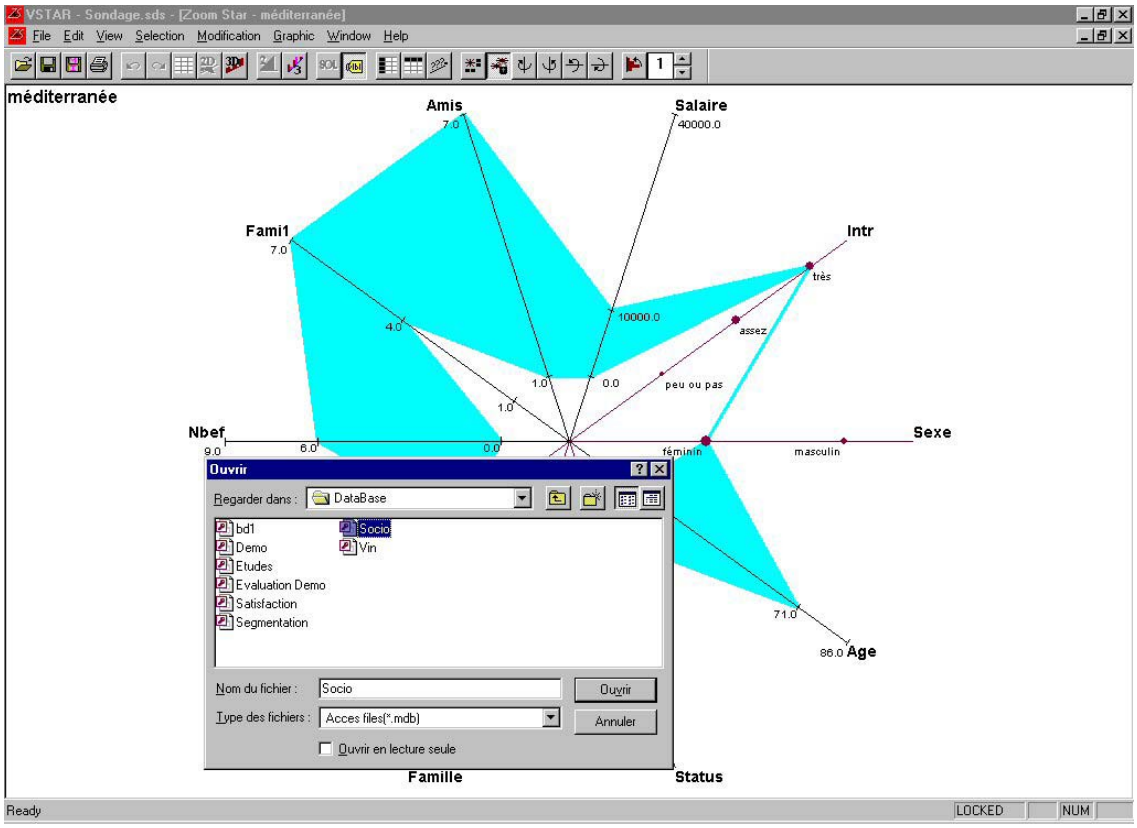


Figure 55: Breakdown - Choice of the original data base

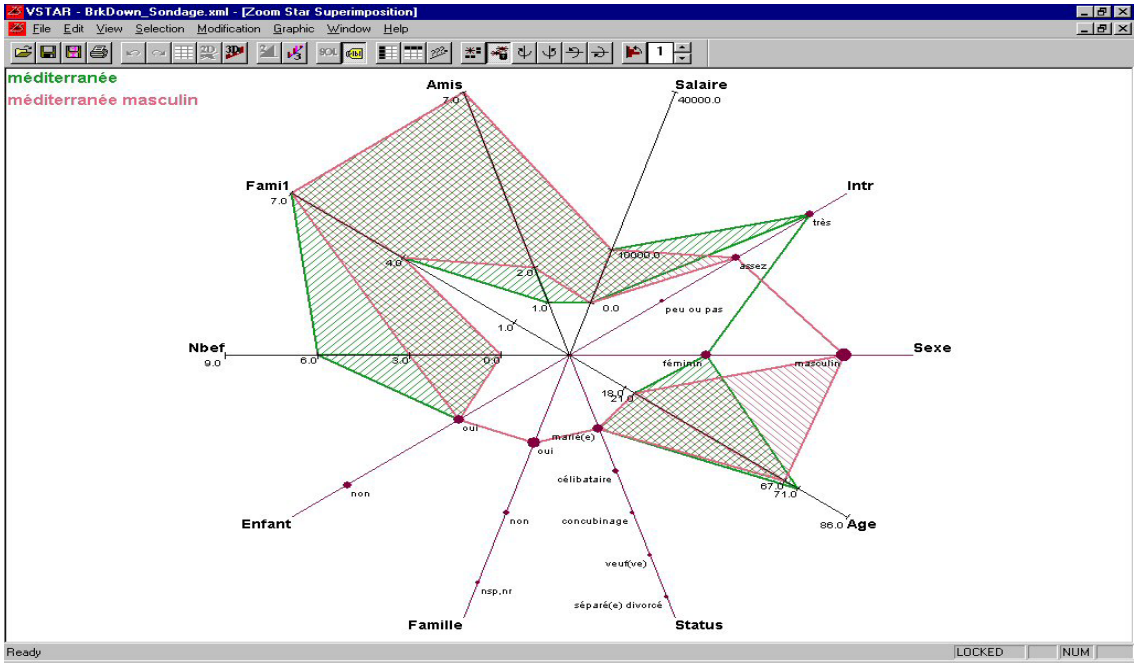


Figure 56: Breakdown - Superimposition of the new and the old objects

The last figure shows in green the initial object (Méditerranée) and in pink the set of men inside the object Méditerranée.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

VSTAT Help Guide

Descriptive Statistics Visualisation

Edited by DAUPHINE

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

26 VSTAT : Descriptive Statistics Visualisation

26.1 Introduction

VSTAT is the visualisation module handling the graphs for the following methods of DSTAT:

Frequencies for categorical multi-valued
Frequencies for interval
Capacities

VSTAT does not handle the Biplot, which relies on a separate module called VPLOT, because that module is shared with other methods (see the VPLOT Internal Help Guide).

VSTAT incorporates many interactivity features for changing the variable(s) selection, customizing the graph, adding texts, ...

26.2 Input

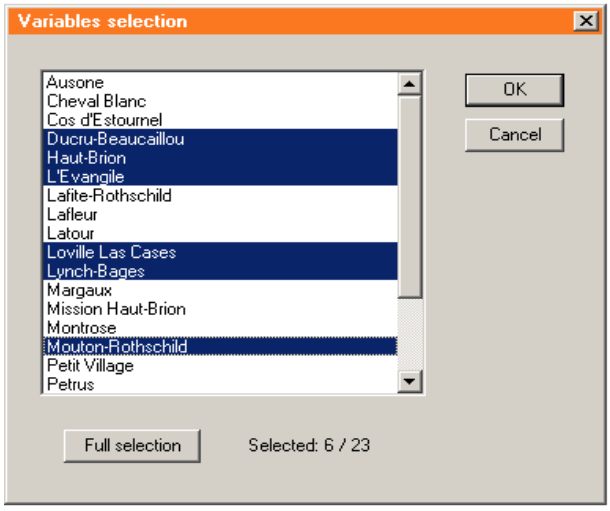
The user does not have to care about the input to VSTAT, which is totally managed by DSTAT and the workbench.

26.3 Starting VSTAT

The VSTAT module is started when the user clicks on the graph icon in the workbench.

Before the graph may be drawn, the user has to select the variable (or variables) that VSTAT must process. The selection dialog then shows up first.

Basically, one graph relates to a single variable, but VSTAT allows selecting several ones in order to draw several graphs in parallel, to ease comparisons between variables.

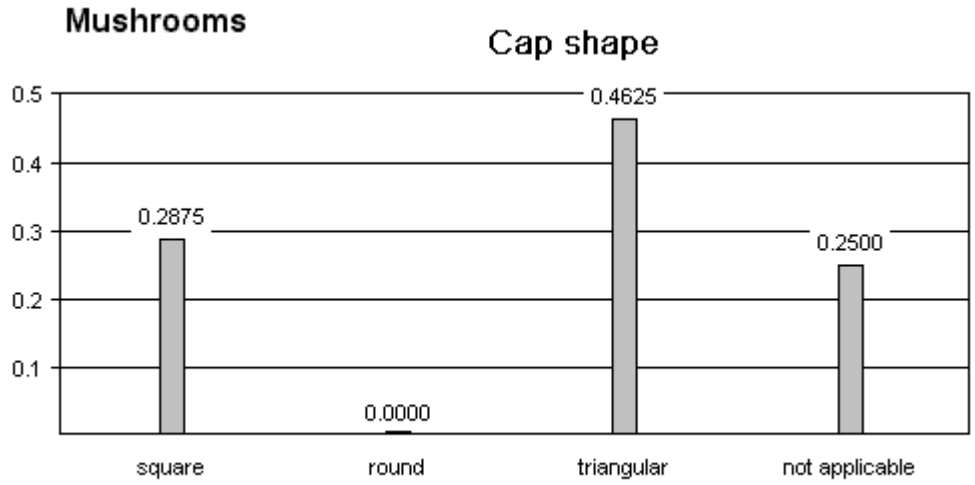


The selectable variables are limited to the set defined in the workbench parameters, i.e. all or only part of the variables of the applicable type.
If a graph for the same chaining had already been saved, it is directly recalled and this dialog is bypassed.

It is possible to later change the selection at any time using the relevant menu command.

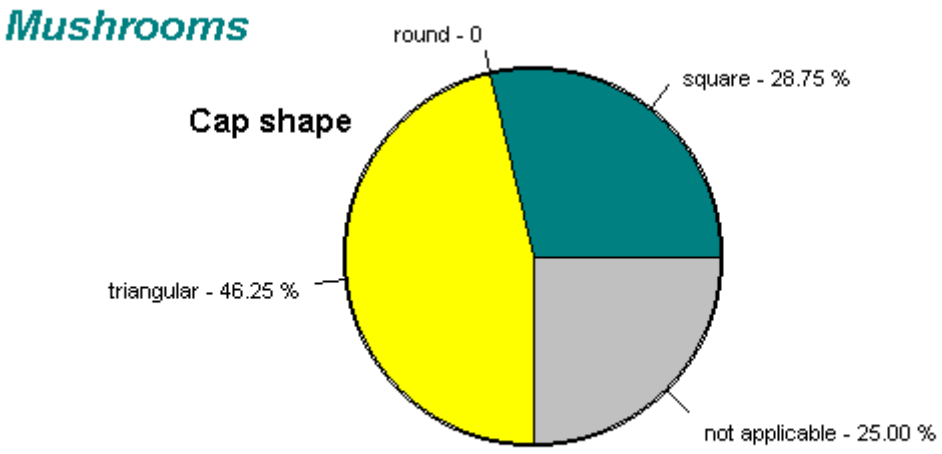
Upon clicking on OK, the graph is displayed.
There are five basic types of graphs:

Frequencies for categorical multi-valued variables – bar chart version



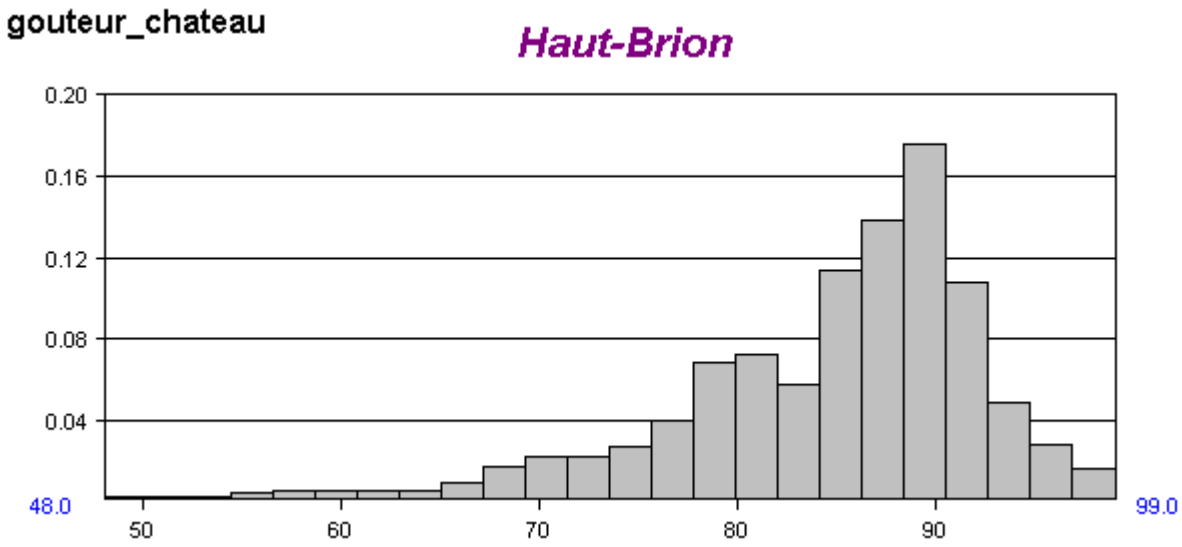
Relative frequencies (multinominal) - file MUSHROO2.SDS - variable 'Cap shape'

Frequencies for categorical multi-valued variables– pie chart version

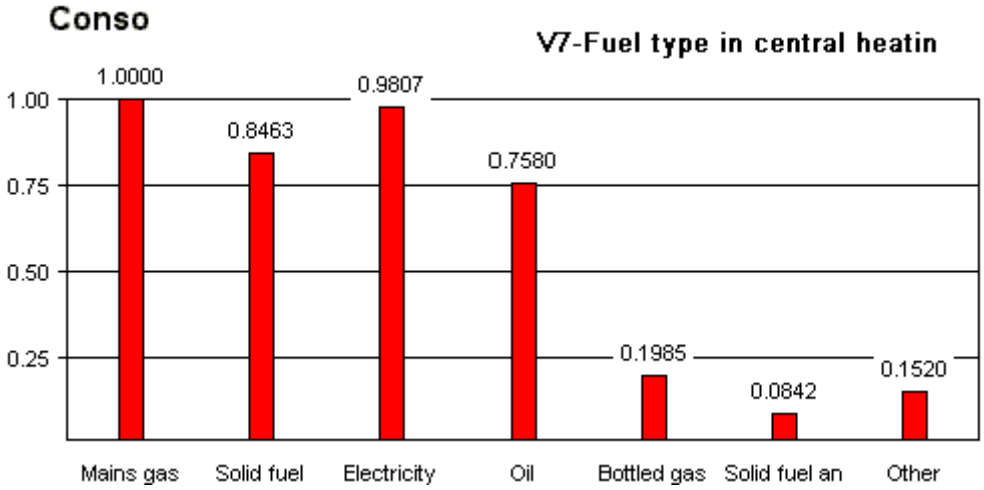


Relative frequencies (multinominal) - file MUSHROO2.SDS - variable 'Cap shape'

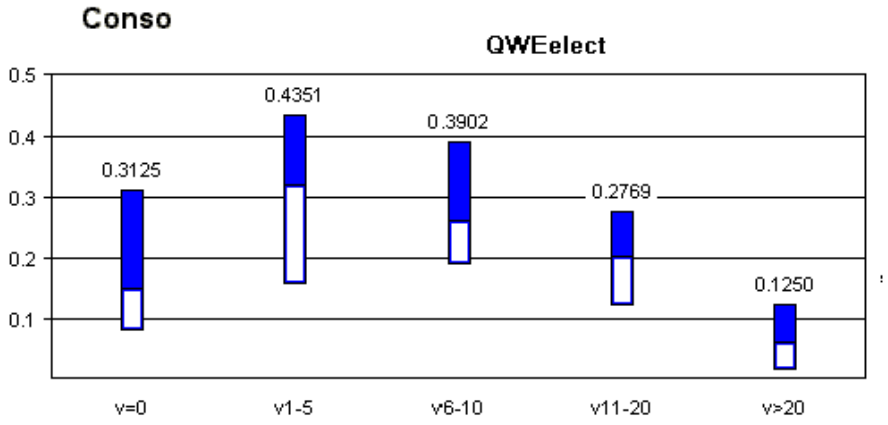
Frequencies for interval variables



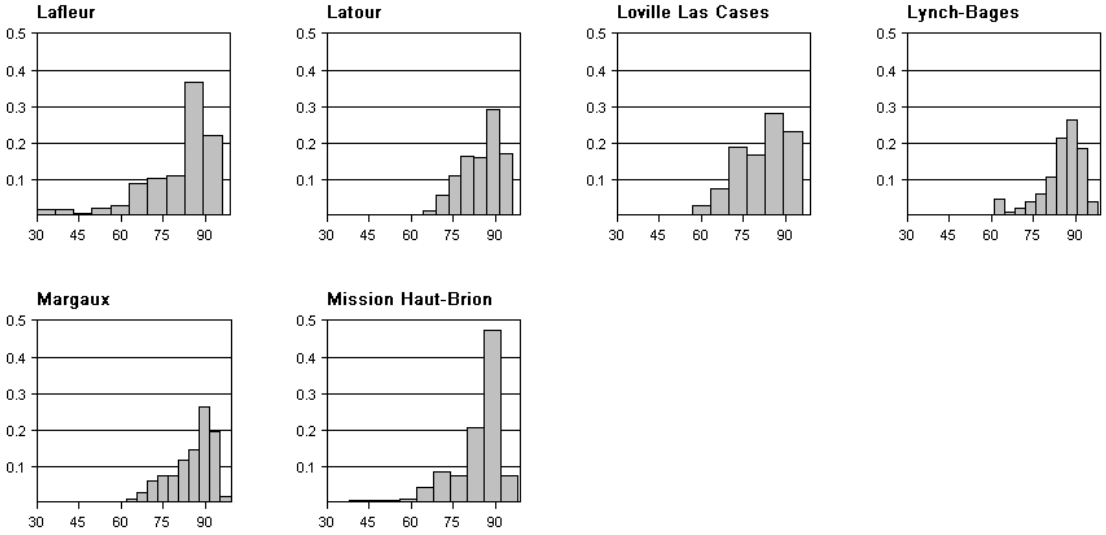
Capacities for modal variables

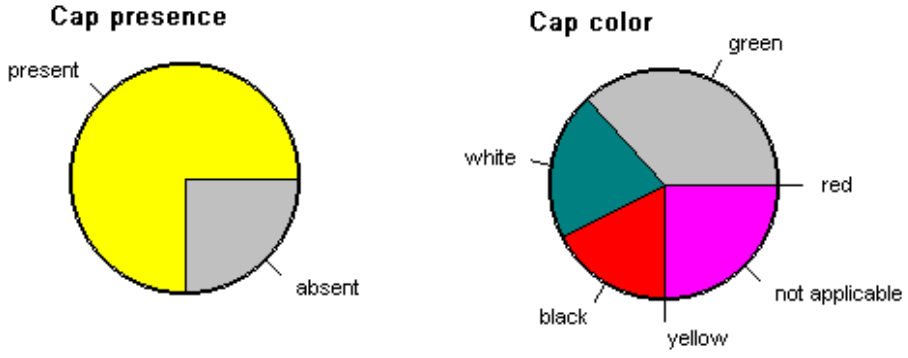


Min / mean / max for modal variables



If several variables have been selected, VSTAT displays multiple graphs, as illustrated by the two examples below:





The multiple graphs layout is automatically computed by VSTAT. The graph readability obviously requires that the number of variables keeps reasonable.

26.4 Main menu commands

In addition to the main menu and the toolbar, there are contextual menus that pop up, when applicable, upon clicking with the right button. When a command of the main menu has an associated toolbar button, it is displayed next to the function label; similarly, if the function can be also accessed from a contextual menu, an asterisk (*) is appended to the label.

26.4.1 File

26.4.1.1 New 

This command may be used if the user wants to start a new graph while already processing another one.

Before leaving the curent graph, the user is prompted for saving it.

26.4.1.2 Save 

Creates a save file (in an internal format) that will automatically be retrieved at the next start of VSTAT for the same chaining (the initial variable selection dialog will then be bypassed).

The current data and environment are saved, so that the user will be able to resume the work as if it had not been interrupted. And if (s)he does not want to reuse it, (s)he may simply use the New command (the current saved graph is kept, and will be lost only if the new graph is saved).

26.4.1.3 Export as

This function saves the graph in formats allowing to import it in word and image processing applications such as Word, PowerPoint, Paint, IrfanView (to name a few).

Two types of files save the display image as such:

1. Windows bitmap (uncompressed pixel map).

2. Portable Network Graphic (lossless compressed pixel map).

The third type, Metafile, saves in a kind of vectorial mode, i.e. records the drawing commands (Windows Graphics Device Interface calls), so that the using application may "replay" those commands and draw the graph as if it had created it by itself. This is less widely supported than the two others.

26.4.1.4 Print 

Stores the bitmap of the graph and copies it to the printer device after rescaling.

26.4.1.5 Printer setup

If the default printer options do not match the user needs, this command allows to change them for the whole session instead of changing them at each print request.

This does not change the default printer options, nor the options selected by other applications, and is lost when the application ends.

26.4.1.6 Exit

Ends VSTAT, and prompts for saving the current graph (same as Save above).

26.4.2 Edit

26.4.2.1 Copy 

Copies the entire graph bitmap to the clipboard.

26.4.2.2 Paste

Imports the contents of the clipboard (bitmap) into the graph.

If the bitmap is too large, the operation is cancelled after warning the user, because such bitmaps are not supposed to take much space in the graph.

The bitmap is initially placed at the top left corner of the graph, but may be dragged with the mouse to adjust its position. And it is possible to move it later at any time in the same way.

26.4.3 View

26.4.3.1 Tool Bar

Shows or hides the Tool Bar (flip/flop)

26.4.3.2 Status Bar

Shows or hides the Status Bar (flip/flop)

26.4.4 Process

26.4.4.1 Select variable(s) 

Allows to change the variable(s) selection and restart a new graph.

The selection dialog is the same as described under «Starting VSTAT».

The new graph is displayed upon clicking on OK.

This command is disabled if a single variable has been selected in the workbench.

N.B.: contrary to the New command, the new graph there does not use the default customization options; it reuses what was current before the reselection, including

the texts inserted by the user.

26.4.4.2 Display listing

Opens a window and displays the listing.

This avoids going back to the workbench just for that.

26.4.5 Draw

This popup menu deals with the graph display characteristics.

Some commands depend on the type of graph processed; they are disabled when not applicable.

26.4.5.1 Pie chart and Bar chart  

These are two mutually exclusive options for displaying the "Frequencies for categorical multi-valued" type of graph.

26.4.5.2 Min/mean/max and Capacities  

Under «Capacities» are in fact two types of graphs:

- the basic «capacities» type, taken by default
- the companion «Min/mean/max» type

This command allows to switch between both, and carries the label of the alternate one,

to indicate the way of the switch.

26.4.5.3 Display values (*)

Shows or hides the values over the bars or next to the pie sectors.

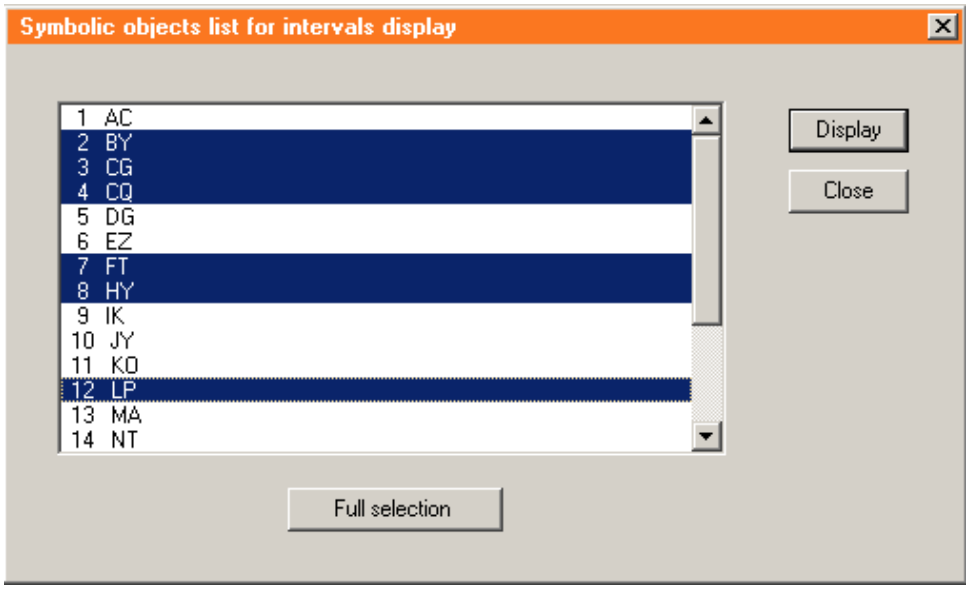
Hiding the values may be necessary when the bars are not spaced enough in a bar chart, or not wide enough in an histogram, given the value string length..

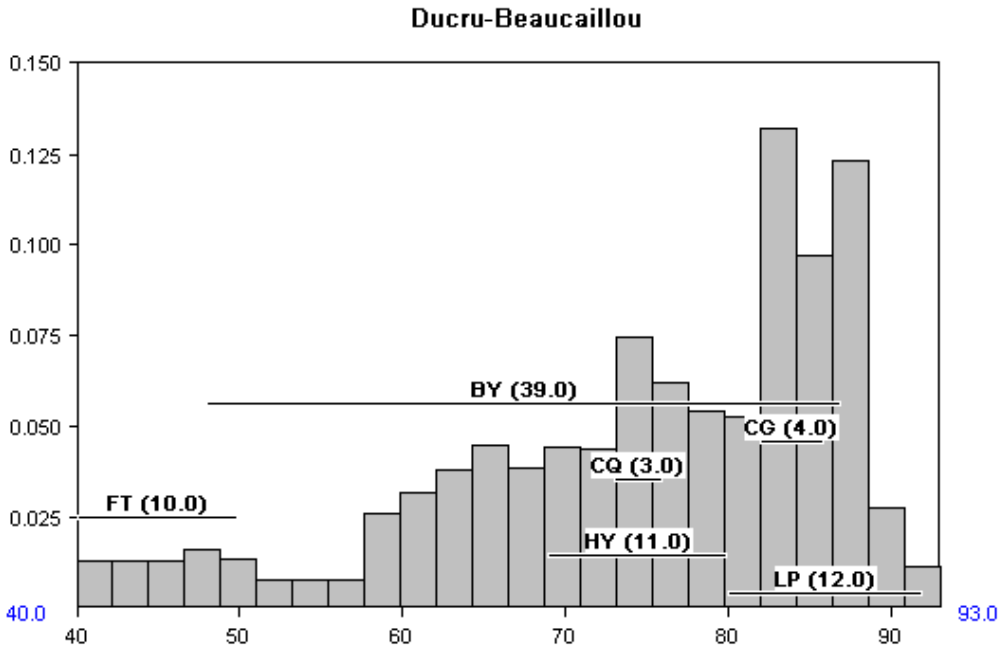
26.4.5.4 Display objects intervals (*)

In the "Frequencies for interval" type of graph, allows to display in overlay the intervals of the symbolic objects for the variable(s) being displayed.

This is a flip/flop type command.

The following selection dialog is used:





The numbers in parenthesis are the width of the object interval.

If the number of objects selected is too high to correctly fit in the graph frame, scrolling using the arrow keys is enabled.

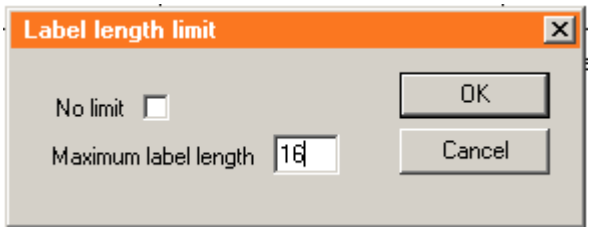
26.4.5.5 Force origin (*)

For interval histograms, directs VSTAT to include the origin in the graph even if the variable overall interval does not encompass it. This is a flip/flop type function.

26.4.5.6 Set labels length limit

In case of bar chart, the labels of the categories are displayed under the bars. If the bars are too close or the labels too long, there is overlapping between labels. Limiting the length aims to avoid this problem.

The following dialog is used:



26.4.5.7 Zoom subgraph

When in multi-graph mode, this command allows to select one of the graphs and have it displayed like in single graph mode.

When it is selected, the mouse cursor changes to a cross; the user has to move the cursor over the target graph and click there (or cancel with the Escape key).

Double clicking inside the subgraph does the same.

26.4.5.8 Return to multiple graphs

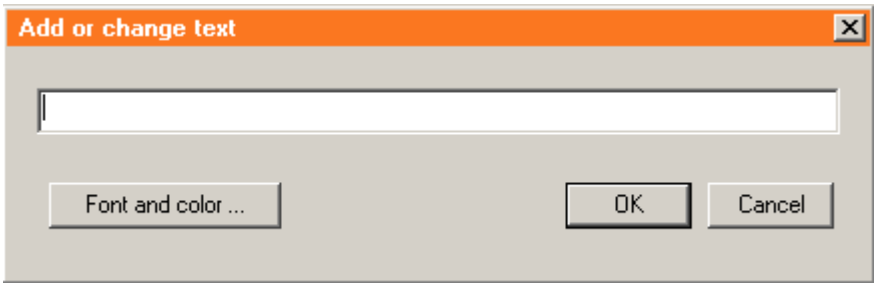
This command cancels the previous one.

26.4.5.9 Insert text

This command allows to insert a text anywhere in the graph area.

When it is selected, the mouse cursor changes to a cross; the user has to move the cursor at the insertion point (top left corner) and click there (or cancel with the Escape key).

The following dialog shows up:



The «Font and color» button calls the standard Windows dialog for selecting another font and/or another text color.

Once displayed, the text can be moved to adjust its position, by dragging it with the mouse.
It remains possible to move it later at any time in the same way.
N.B.: When a text is moveable, the mouse cursor changes to a quadruple arrow when passing over it, so indicating that possibility.
All the texts inserted by the user are moveable, as well as the titles inserted by VSTAT; the texts linked physically to the graph are not moveable: values or labels next to the bars, and scale tags.

A text can be changed via its contextual menu, which pops up when right clicking on it.
The same dialog as above is used.

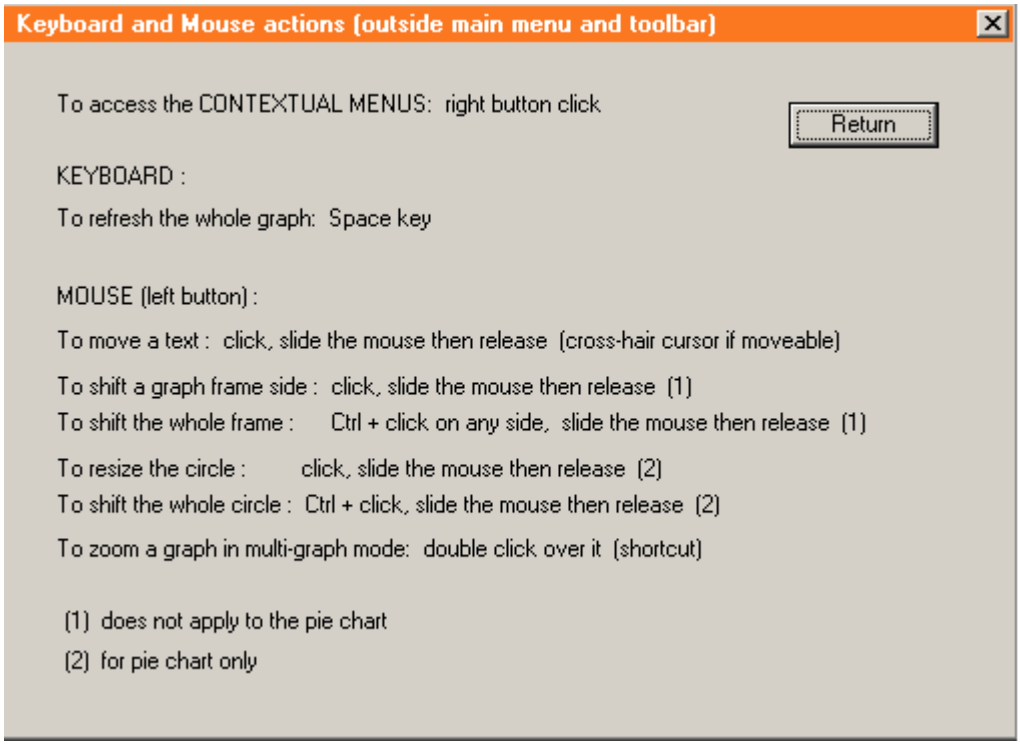
26.4.5.10 Redraw 

This command clears the display and redraws the graph.
Hitting the space bar does it too.

26.4.6 Help

26.4.6.1 Keyboard and mouse actions

Some functions are directly triggered from the keyboard and the mouse.
This command displays the following panel, summarizing those possibilities:



26.4.6.2 Direct keyboard and mouse commands

See above command description.

26.4.7 Contextual menu commands

Contextual menus have three objectives:

1. Provide an alternate and quicker access to some commands available in the main menu.
2. Filter the set of commands according to the nature and status of the target.
3. Add specific commands that would not be easily manageable via the main menu.

26.4.8 Commands duplicating main menu commands

Set bars default color
Sel labels length limit
Display objects intervals
Display values
Force origin

Refer to section 4 for the description of these commands.

26.4.9 Commands for display layout and look

Hide vertical scale (multi-graph mode only)

Used in case of numerous graphs, to make the display less crowded and better looking.

This is a flip/flop type command.

Hide horizontal scale (interval histograms and multi-graph mode only)

Used in case of numerous graphs, to make the display less crowded and better looking.

This is limited to the interval histograms, only graphs where there is an horizontal scale (for the bar charts, where there are labels displayed horizontally under the graph frame, the user may limit their length with the relevant command if deemed necessary).

This a flip/flop type command.

Force overall limits (interval histograms and multi-graph mode only)

Basically, VSTAT computes each graph scaling so that the graph frame space is used up.

For interval histograms in multi-graph mode, it may be valuable to compare the variables with a common scaling; this command then directs VSTAT to recompute the graphs scales with setting the lower and higher limits of each one to the overall lower and higher limits of the selected variables, both vertically and horizontally.

This a flip/flop type command.

26.4.10 Commands for texts

This covers the texts inserted by the user or by VSTAT; not the values or labels next to the bars, nor the scale tags, which must not be modified nor deleted.

26.4.10.1 Change text

Allows to change the text contents, font and color.
The same dialog as for inserting a text is used (see «Insert text» in section 4).

26.4.10.2 Delete

Self-explanatory.

26.4.11 Command for pasted bitmaps

26.4.11.1 Delete bitmap

Self-explanatory.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

VDISS Help Guide

Matrix Visualisation

Edited by DIB

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

Date: **6/06/2003**

27 VDISS : Matrix Visualisation

27.1 Commands

File menu

Tools menu

View menu

Help menu

27.2 File menu commands

The File menu offers the following commands:

Open	Opens an existing Sodas/Xml file.
Exit	Exits VDiss.

27.2.1 Open command (File menu)

Use this command to open a SODAS or XML File (*.sds/*.xml).

Shortcuts

Toolbar:

Keys : CTRL+O

File Open dialog box

The following options allow you to specify which file to open:

File Name

Type or select the filename you want to open. This box lists files with the extension you select in the List Files of Type box.

Select the type of file you want to open:

Drives

Select the drive in which VDISS stores the file that you want to open.

Directories

Select the directory in which VDISS stores the file that you want to open.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

27.2.2 Exit command (File menu)

Use this command to end your VDISS session.

Shortcuts

Mouse: Double-click the application's Control menu button.

Keys: ALT+F4

27.3 Tools menu commands

The Tools menu offers the following commands:

Bi-Dimensional Mapping	Shows the Sammon Representation of symbolic objects.
Graphic Representation	Shows the Graphic Representation of dissimilarities between symbolic object.

27.3.1 Bi-Dimensional Mapping command (Tools menu)

Use this command to create a graphic display of Sammon Representation of symbolic objects.

In the bi-dimensional scatterplot, the results of a nonlinear mapping of the Symbolic Objects into a bi-dimensional space are shown.

The nonlinear mapping is based on Sammon's algorithm that takes in input a matrix of distances between N objects and returns a collection on point in the bi-dimensional space such that their Euclidean distances preserve the "structure" of the original distance matrix.

Sammon's nonlinear mapping: Let Xi, i=1, …, N, be N vectors in an M-dimensional space, and =dist(Xi,Xj) the distances between two vectors Xi and Xj. Then Sammon's nonlinear mapping determines N vectors Yi in the bi-dimensional space such that the distances dij=dist(Yi,Yj) preserve the "structure" of the distances.

27.3.2 Graphic Representation command (Tools menu)

Use this command to create a graphic display of Graphic Representation of dissimilarity matrix.

In the line graph dissimilarities are reported along the vertical axis, while Symbolic Objects are listed along the horizontal axis, in the same order in which they are reported in the Sodas/Xml file.

In the Partial Lines each line represents the dissimilarity between a given Symbolic Object and the subsequent Symbolic Objects in the file. The number of line in each graph is equal to the number of Symbolic Objects minus one.

In the Total Lines, Bars and Pies each line, bar or pie represents the dissimilarity between a given Symbolic Object and the others Symbolic Objects in the file. The number of lines, bars or pies, in each graph is equal to the number of Symbolic Objects.

27.4 View menu commands

The View menu offers the following commands:

Toolbar	Shows or hides the toolbar.
Status Bar	Shows or hides the status bar.

27.4.1 Toolbar command (View menu)

Use this command to display and hide the Toolbar, which includes buttons for some of the most common commands in VDiss, such as File Open. A check mark appears next to the menu item when the Toolbar is displayed.

See Toolbar for help on using the toolbar.

Toolbar

The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in VDISS,

To hide or display the Toolbar, choose Toolbar from the View menu (ALT, V, T).

Open an existing document. VDISS displays the Open dialog box, in which you can locate and open the desired file.

27.4.2 Status Bar command (View menu)

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed.

See Status Bar for help on using the status bar.

Status Bar

The status bar is displayed at the bottom of the VDiss window. To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them. If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right areas of the status bar indicate which of the following keys are latched down:

Indicator	Description
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.

27.5 Help menu commands

The Help menu offers the following commands, which provide you assistance with this application:

Help Topics	Offers you an index to topics on which you can get help.
About	Displays the version number of this application.

27.5.1 Help Topics (Help menu)

Use this command to display the Help.

27.5.2 Index command (Help menu)

Use this command to display the opening screen of Help. From the opening screen, you can jump to step-by-step instructions for using VDiss and various types of reference information.

Once you open Help, you can click the Contents button whenever you want to return to the opening screen.

27.5.3 Using Help command (Help menu)

Use this command for instructions about using Help.

27.5.4 About command (Help menu)

Use this command to display the copyright notice and version number of your copy of VDiss.

Context Help command

Use the Context Help command to obtain help on some portion of VDiss. When you choose the Toolbar's Context Help button, the mouse pointer will change to an arrow and question mark. Then click somewhere in the VDiss window, such as another Toolbar button. The Help topic will be shown for the item you clicked.

Shortcut

Keys: SHIFT+F1

Title Bar

The title bar is located along the top of a window. It contains the name of the application and document.

To move the window, drag the title bar. Note: You can also move dialog boxes by dragging their title bars.

A title bar may contain the following elements:

- Application Control-menu button
- Document Control-menu button
- Maximize button
- Minimize button
- Name of the application
- Name of the document
- Restore button

Scroll bars

Displayed at the right and bottom edges of the document window. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the document. You can use the mouse to scroll to other parts of the document.

Size command (System menu)

Use this command to display a four-headed arrow so you can size the active window with the arrow keys.

After the pointer changes to the four-headed arrow:

1. Press one of the DIRECTION keys (left, right, up, or down arrow key) to move the pointer to the border you want to move.
2. Press a DIRECTION key to move the border.
3. Press ENTER when the window is the size you want.

Note: This command is unavailable if you maximize the window.

Shortcut

Mouse : Drag the size bars at the corners or edges of the window.

Move command (Control menu)

Use this command to display a four-headed arrow so you can move the active window or dialog box with the arrow keys.

Note: This command is unavailable if you maximize the window.

Shortcut

Keys: CTRL+F7

Minimize command (application Control menu)

Use this command to reduce the VDiss window to an icon.

Shortcut

Mouse: Click the minimize icon on the title bar.

Keys: ALT+F9

Maximize command (System menu)

Use this command to enlarge the active window to fill the available space.

Shortcut

Mouse : Click the maximize icon on the title bar; or double-click the title bar.

Keys : CTRL+F10 enlarges a document window.

Close command (Control menus)

Use this command to close the active window or dialog box.

Double-clicking a Control-menu box is the same as choosing the Close command.

Note: If you have multiple windows open for a single document, the Close command on the document Control menu closes only one window at a time. You can close all windows at once with the Close command on the File menu.

Shortcuts

Keys: CTRL+F4 closes a document window

ALT+F4 closes the VDiss window or dialog box

Switch to command (application Control menu)

Use this command to display a list of all open applications. Use this "Task List" to switch to or close an application on the list.

Shortcut

Keys: CTRL+ESC

Dialog Box Options

When you choose the Switch To command, you will be presented with a dialog box with the following options:

Task List

Select the application you want to switch to or close.

Switch To

Makes the selected application active.

End Task

Closes the selected application.

Cancel

Closes the Task List box.

Cascade

Arranges open applications so they overlap and you can see each title bar. This option does not affect applications reduced to icons.

Tile

Arranges open applications into windows that do not overlap. This option does not affect applications reduced to icons.

Arrange Icons

Arranges the icons of all minimized applications across the bottom of the screen.

Modifying the Document

<< Write application-specific help here that provides an overview of how the user should modify a document using your application.>>

No Help Available

No help is available for this area of the window.

Dissimilarity Matrix Dialog Help Index

Commands

Tools_menu

Options menu

Help menu

Tools menu commands

The Tools menu offers the following commands:

Bi-Dimensional Mapping	Shows a graphic display of Sammon Representation of symbolic objects.
Graphic Representation	Shows a graphic display of Graphic Representation of the dissimilarities between symbolic objects.
Exit	Exits Dialog.

Options menu commands

The Options menu offers the following commands:

SO's Label	Shows in the matrix the label of symbolic objects.
SO's Name	Shows in the matrix the name of symbolic objects.
Property	Shows the Dissimilarity Property Form.

Dissimilarity Properties menu commands

The Dissimilarity Properties Menu offers the following commands:

Definite	Compute the Definite property for the dissimilarity matrix
Even	Compute the Even property for the dissimilarity matrix
Pseudo Matric	Compute the Pseudo Metric property for the dissimilarity matrix
Ultrametric	Compute the Ultrametric property for the dissimilarity matrix
Tree	Compute the Tree property for the dissimilarity matrix
Robinsonian	Compute the Robinsonian property for the dissimilarity matrix

Means menu commands

The Means menu commands offers the following commands:

Arithmetical Mean	Compute the Arithmetical Mean
Geometrical Mean	Compute the Geometrical Mean
Quadratical Mean	Compute the Quadratical Mean
Harmonical Mean	Compute the Harmonical Mean
Quadratic Average Difference	Compute the Quadratic Average Difference
Common Average Removal	Compute the Common Average Removal

Bi-Dimensional Mapping Dialog Help Index

Commands

Options menu

Edit menu

Help menu

Options menu commands

The Options menu offers the following commands:

Zoom Commands	Shows the Zoom Commands Form.
Exit	Exits Dialog.

Edit menu commands

The Edit menu offers the following commands:

Copy	Copies data from the graph to the clipboard.

Zoom menu commands

The Zoom menu offers the following commands:

Zoom In	Enlarge the central part of graphic.
Zoom Out	Decrease the graphic.
Refresh	Bring back the graphic at the original condition.
2X	Enlarge or decrease the graphic twice.
4X	Enlarge or decrease the graphic four times.
8X	Enlarge or decrease the graphic eight times.
Up	Shift the graphic's content upward.
Down	Shift the graphic's content downward.
Right	Shift the graphic's content on the right.
Left	Shift the graphic's content on the left.
List of Symbolic Objects	Select a symbolic object and shift it in the graphic's centre.
Catesian Co-ordinates	Shows or hides the Cartesian Co-oridinate.
SO's Name	Shows or hides in the Sammon representation the label/name of symbolic objects.
Labels/ Names	Shows in the Sammon representation the name or the label of

	symbolic objects.
Legend	Shows or hides in the Sammon representation the legend.

Graphic Representation Dialog Help Index

Commands

Options menu

Edit menu

Help menu

Options menu commands

The Options menu offers the following commands:

Style Commands	Shows the Styles Menu Form.
Exit	Exits Dialog.

Styles menu commands

The Styles menu offers the following commands:

Partial Lines	Visualize the graphic of distances of dissimilarity in linear form. In this representation considering only the lower triangolar matrix of the dissimilarity matrix.
Total Lines	Visualize the graphic of distances of dissimilarity in linear form. In this representation considering the whole dissimilarity matrix.
Bars	Visualize the graphic of distances of dissimilarity in bar chart form. In this representation considering the whole dissimilarity matrix.
Pies	Visualize the graphic of distances of dissimilarity in pies chart form. In this representation considering the whole dissimilarity matrix.
2D	Visualize the different charts in bi-dimensional form.
3D	Visualize the different charts in tri-dimensional form.
Labels/ Names	Shows in the Graphic Representation the name or the label of symbolic objects.
Legend	Shows or hides in the Graphic Representation the legend.

Edit menu commands

The Edit menu offers the following commands:

Copy	Copies data from the graph to the clipboard.

Bi-Dimensional Mapping command (Tools menu)

Shows the graphic display of the symbolic object's projetion from a L-dimensional space to a bi-dimensional space.

In the bi-dimensional scatterplot, the results of a nonlinear mapping of the Symbolic Objects into a bi-dimensional space are shown. The nonlinear mapping is based on Sammon's algorithm that takes in input a matrix of distances between N objects and returns a collection on point in the bi-dimensional space such that their Euclidean distances preserve the "structure" of the original distance matrix.

Sammon's nonlinear mapping: Let Xi, i=1, …, N, be N vectors in an M-dimensional space, and =dist(Xi,Xj) the distances between two vectors Xi and Xj. Then Sammon's nonlinear mapping determines N vectors Yi in the bi-dimensional space such that the distances dij=dist(Yi,Yj) preserve the "structure" of the distances.

Graphic Representation command (Tools menu)

Shows the graphic display of measure of dissimilarity among different symbolic objects; the measure of dissimilarity expressed to matrix of measures of dissimilarity inside file Sodas/Xml (*.sds/*.xml).

In the line graph dissimilarities are reported along the vertical axis, while Symbolic Objects are listed along the horizontal axis, in the same order in which they are reported in the Sodas/Xml file.

In the Partial Lines each line represents the dissimilarity between a given Symbolic Object and the subsequent Symbolic Objects in the file. The number of line in each graph is equal to the number of Symbolic Objects minus one.

In the Total Lines, Bars and Pies each line, bar or pie represents the dissimilarity between a given Symbolic Object and the others Symbolic Objects in the file. The number of lines, bars or pies, in each graph is equal to the number of Symbolic Objects.

SO's Label command (Options menu)

Shows in the matrix the label of symbolic objects.

SO's Name command (Options menu)

Shows in the matrix the name of symbolic objects.

Property command (Options menu)
Show the Dissimilarity Properties Form..

Exit command (Tools/Options menu)
Use this command to end your work on dialog.

Shortcuts
Keys: ALT+F4

Copy command (Edit menu)
Use this command to copy the graph onto the clipboard.

Zoom In command (Zoom Menu Commands)
Enlarge the central section of graphic.

The extent of this section depends on the typology of enlargement (2X, 4X or 8X): the more great is the value of the typology of enlargement, the more narrow is the section and more great is the enlargement.

Observation:
If the section to enlarge in not in the central area of the graphic, it is possible shift it through the buttons on form.

Zoom Out command (Zoom Menu Commands)
Decrease the graphic.

Decrease depends on the typology of enlargement (2X, 4X or 8X): the more great is the value of the typology of enlargement, the more great is the decrease.

Refresh command (Zoom Menu Commands)
Bring back the graphic at the original condition and annul every operation to zooming or to shifting.

2X, 4X and/or 8X command (Zoom Menu Commands)
Fix:

- the extent to the central section of graphic to enlarge; this section is inversely proportional to the typology of enlargement (2X, 4X or 8X): the more great is the value of the typology of enlargement, the more narrow is the section and more great is the enlargement.
- indicates how many times the area of graphic is decreased.

2X means that the area is decreased/enlarged twice.

4X means that the area is decreased/enlarged four times.

8X means that the area is decreased/enlarged eight times.

Up command (Zoom Menu Commands)
Shift the graphic's content upward.

With the others commands of direction, shift the area to the centre of the graphic to execute the enlargemnt or the decrease.

Down command (Zoom Menu Commands)
Shift the graphic's content downward.

With the others commands of direction, shift the area to the centre of the graphic to execute the enlargemnt or the decrease.

Right command (Zoom Menu Commands)
Shift the graphic's content on the right.

With the others commands of direction, shift the area to the centre of the graphic to execute the enlargemnt or the decrease.

Left command (Zoom Menu Commands)
Shift the graphic's content on the left.

With the others commands of direction, shift the area to the centre of the graphic to execute the enlargemnt or the decrease.

List Of Symbolic Objects (Zoom Menu Commands)
Selecting a symbolic object to the list box, it shifts to the centre of the area inside graphic.

Besides, this list box is also a legend because it show the likeness of the symbolic objects to the order numbers.

Cartesian Co-ordinates command (Zoom Menu Commands)
Use this command to display or hide the Cartesian Co-ordinates of the point.

SO's Name command (Zoom Menu Commands)
Use this command to dispaly or hide the label/name of symbolic objects in the Sammon representation.

Labels/Names command (Zoom Menu Commands)
Shows in the Sammon representation the name or the label of symbolic objects.

Legend command (Dialog commands)
Use this command to display or hide the legend in the representation.

Arithmetical Mean command (Means Menu Commands)
Compute the Arithmetical Mean.created with Help to RTF file format converter

Geometrical Mean command (Means Menu Commands)
Compute the Geometrical Mean.

Quadratical Mean command (Means Menu Commands)
Compute the Quadratical Mean.

Harmonical Mean command (Means Menu Commands)
Compute the Harmonical Mean.

Quadratic Average Difference command (Means Menu Commands)

Compute the Quadratic Average Difference.

Common Average Removal command (Means Menu Commands)
Compute the Common Average Removal.

Execute command (Means Menu Commands)
Compute the value of the selected mean

Partial Lines (Styles Menu Commands)
Shows the matrix of measures of dissimilarity on graphic form (2D and/or 3D) as follows:

- on the axis of abscissae there are the symbolic objects *(objects)*
- on the axis of ordinates there is the range of value of the dissimilarity matrix *(distances).*

As from every symbolic object, on the axis of abscissae, start a line of different color; this line shows the trend of the measures of dissimilarity on relationship with others symbolic objects.
This trend takes into consideration, for every symbolic object, only the symbolic objects with a greatest ordinal numbers.

Observation:

- the legend shows the color of the line for every symbolic object
- through a mouse click either on dissimilarity line or on symbolic object in the legend, it is possible visualize the different value of dissimilarity.

Total Lines (Styles Menu Commands)

Shows the matrix of measures of dissimilarity on graphic form (2D and/or 3D) as follows:

- on the axis of abscissae there are the symbolic objects *(objects)*
- on the axis of ordinates there is the range of value of the dissimilarity matrix *(distances).*

As from every symbolic object, on the axis of abscissae, start a line of different color; this line shows the trend of the measures of dissimilarity on relationship with others symbolic objects.

This trend takes into consideration, for every symbolic object, the totality of symbolic objects.

Observation:

- the legend shows the color of the line for every symbolic object
- through a mouse click either on dissimilarity line or on symbolic object in the legend, it is possible visualize the different value of dissimilarity.

Bars (Styles Menu Commands)

Shows the matrix of measures of dissimilarity on graphic form (2D and/or 3D) as follows:

- on the axis of abscissae there are the symbolic objects *(objects)*
- on the axis of ordinates there is the range of value of the dissimilarity matrix *(distances).*

As from every symbolic object, on the axis of abscissae, start a set of bars of different color; these bars show the trend of the measures of dissimilarity on relationship with others symbolic objects.

This trend takes into consideration, for every symbolic object, the totality of symbolic objects.

Observation:

- the legend shows the color of the bar for every symbolic object
- through a mouse click either on dissimilarity bar or on symbolic object in the legend, it is possible visualize the different value of dissimilarity.

Pies (Styles Menu Commands)

Shows the matrix of measures of dissimilarity on graphic form (2D) as follows:

- every symbolic objects is associeted with a pie graphic.

The width of every graphic slice is directly proportional at the mesure of dissimilarity between the symbolic object linked to graphic pie and the symbolic object linked to color of the slice.

Observation:

- the legend shows the color of the slice for every symbolic object
- through a mouse click either on a slice or on symbolic object in the legend, it is possible visualize the differet value of dissimilarity.

Partial Lines (Styles Menu Commands)

Shows the matrix of measures of dissimilarity on graphic form (2D and/or 3D) as follows:

- on the axis of abscissae there are the symbolic objects *(objects)*
- on the axis of ordinates there is the range of value of the dissimilarity matrix *(distances).*

As from every symbolic object, on the axis of abscissae, start a line of different color; this line shows the trend of the measures of dissimilarity on relationship with others symbolic objects.

This trend takes into consideration, for every symbolic object, only the symbolic objects with a greatest ordinal numbers.

Total Lines (Styles Menu Commands)

Shows the matrix of measures of dissimilarity on graphic form (2D and/or 3D) as follows:

- on the axis of abscissae there are the symbolic objects *(objects)*
- on the axis of ordinates there is the range of value of the dissimilarity matrix *(distances).*

As from every symbolic object, on the axis of abscissae, start a line of different color; this line shows the trend of the measures of dissimilarity on relationship with others symbolic objects.

This trend takes into consideration, for every symbolic object, the totality of symbolic objects.

Bars (Styles Menu Commands)

Shows the matrix of measures of dissimilarity on graphic form (2D and/or 3D) as follows:

- on the axis of abscissae there are the symbolic objects *(objects)*
- on the axis of ordinates there is the range of value of the dissimilarity matrix *(distances).*

As from every symbolic object, on the axis of abscissae start a set of bars of different color; these bars show the trend of the measures of dissimilarity on relationship with others symbolic objects.

This trend takes into consideration, for every symbolic object, the totality of symbolic objects.

Pies (Styles Menu Commands)

Shows the matrix of dissimilarity on graphic form (2D) as follows:

every symbolic objects is associated with a pie graphic.

The width of every graphic slice is directly proportional at the measure of dissimilarity between the symbolic object linked to graphic pie and the symbolic object linked to color of the slice.

2D and/or 3D command (Style Menu Commands)

Change the form of representation from bi-dimensional way (2D) to tri-dimensional way (3D) and vice versa, for every type graph, with the exception of the graphic representation with pies, where is realized only the bi-dimensional form.

2D fix the bi-dimensional form.

3D fix the tri-dimensional form.

Labels/Names command (Options menu)
Shows in the Graphic Representation the name or the label of symbolic objects.

Definite (Dissimilarity Properties Menu Commands)
Compute the Definite property for the dissimilarity matrix:

*The measure d is called a **definite** dissimilarity measure if:*
*for all a, b member of dissimilarity matrix: d(a,b)=0 implies a=b*created with Help to RTF file format converter

Even (Dissimilarity Properties Menu Commands)
Compute the Even property for the dissimilarity matrix:

*The measure d is termed a **even** if for all a, b member of dissimilarity matrix:*
d(a,b)=0 implies d(a,c)=d(b,c) for all c member of dissimilarity matrix

Pseudo Metric (Dissimilarity Properties Menu Commands)
Compute the Pseudo Metric property for the dissimilarity matrix:

A dissimilarity matrix which fulfils the triangle inequality:
d(a,b)<=d(a,c)+d(c,b) for all a,b,c member of dissimilarity matrix

*is called a **pseudo metric** or a **semi-distance**.*
*It is termed **metric** or **distance** if it fulfils, in addition the definite property*

Ultrametric (Dissimilarity Properties Menu Commands)
Compute the Ultrametric property for the dissimilarity matrix:

*A dissimilarity matrix is called an **ultrametric** if it fulfills the ultrametric inequality*
d(a,b)<=max{d(a,c), d(c,b)} for all a, b,c member of dissimilarity matrix

Tree (Dissimilarity Property Dialog Commands)
Compute the Tree property for the dissimilarity matrix:

*The misure d is termed a **tree distance** if it fulfils the **'four-pointt condition'**:*
d(a,b)+d(c,d)<=max{d(a,c)+d(b,d), d(a,d)+d(b,c)}
for all a, b,c,d member of dissimilarity matrix
*which is also called **Buneman's inequality***

Robinsonian (Dissimilarity Properties Menu Commands)
Compute the Robinsonian property for the dissimilarity matrix:

A dissimilarity matrix is called Robinsonian if the dissimilarity dkl increasing when k or l moves
away from diagonal (with k=l). Formally: for all k member of {1,..,n} we have:
dk,k<=dk,k+1<=..<=dk,n-1<=dk,n and dk,k<=dk,k-1<=..<=dk,2<=dk,1
dk,k<=dk+1,k<=..<=dn-1,k<=dn,k and dk,k<=dk-1,k<=..<=d2,k<=d1,k

Execute command (Dissimilarity Properties Menu Commands)
Compute the value of the selected properties

Value of Dissimilarity Property (Dissimilarity Properties Menu Commands)
Show the value of the related property

Summary
Vdiss module gives the possibility to select a Sodas/Xml file (*sds or *.xml) containing a symmetric dissimilarity matrix, to show it in a table format, to visualize a line graph of dissimilarities as well as a scatterplot of Sammon's nonlinear mapping into a bi-dimensional space.

In the line graph dissimilarities are reported along the vertical axis, while Symbolic Objects are listed along the horizontal axis, in the same order in which they are reported in the Sodas/Xml file.

In the Partial Lines each line represents the dissimilarity between a given Symbolic Object and the subsequent Symbolic Objects in the file. The number of line in each graph is equal to the number of Symbolic Objects minus one.

In the Total Lines, Bars and Pies each line, bar or pie represents the dissimilarity between a given Symbolic Object and the others Symbolic Objects in the file. The number of lines, bars or pies, in each graph is equal to the number of Symbolic Objects.

In the bi-dimensional scatterplot, the results of a nonlinear mapping of the Symbolic Objects into a bi-dimensional space are shown. The nonlinear mapping is based on Sammon's algorithm that takes in input a matrix of distances between N objects and returns a collection on point in the bi-dimensional space such that their Euclidean distances preserve the "structure" of the original distance matrix.

Sammon's nonlinear mapping: Let Xi, i=1, …, N, be N vectors in an M-dimensional space, and =dist(Xi,Xj) the distances between two vectors Xi and Xj. Then Sammon's nonlinear mapping determines N vectors Yi in the bi-dimensional space such that the distances dij=dist(Yi,Yj) preserve the "structure" of the distances.

Input
Sodas/Xml file containing a dissimilarity matrix.

Output
A dissimilarity table and/or a graphical representation of this table.

Print command (File menu)
Use this command to print a document. This command presents a Print dialog box, where you may specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.

Shortcuts
Toolbar:
Keys: CTRL+P

Print dialog box
The following options allow you to specify how the document should be printed:

Printer
This is the active printer and printer connection. Choose the Setup option to change the printer and printer connection.

Setup
Displays a Print Setup dialog box, so you can select a printer and printer connection.

Print Range
Specify the pages you want to print:

All	Prints the entire document.
Selection	Prints the currently selected text.
Pages	Prints the range of pages you specify in the From and To boxes.

Copies
Specify the number of copies you want to print for the above page range.

Collate Copies
Prints copies in page number order, instead of separated multiple copies of each page.

Print Quality
Select the quality of the printing. Generally, lower quality printing takes less time to produce.

Print Progress Dialog
The Printing dialog box is shown during the time that VDiss is sending output to the printer. The page number indicates the progress of the printing.

To abort printing, choose Cancel.

Print Preview command (File menu)

Use this command to display the active document as it would appear when printed. When you choose this command, the main window will be replaced with a print preview window in which one or two pages will be displayed in their printed format. The print preview toolbar offers you options to view either one or two pages at a time; move back and forth through the document; zoom in and out of pages; and initiate a print job.

Print Preview toolbar

The print preview toolbar offers you the following options:

Print

Bring up the print dialog box, to start a print job.

Next Page

Preview the next printed page.

Prev Page

Preview the previous printed page.

One Page / Two Page

Preview one or two printed pages at a time.

Zoom In

Take a closer look at the printed page.

Zoom Out

Take a larger look at the printed page.

Close

Return from print preview to the editing window.

Print Setup command (File menu)

Use this command to select a printer and a printer connection. This command presents a Print Setup dialog box, where you specify the printer and its connection.

Print Setup dialog box

The following options allow you to select the destination printer and its connection.

Printer

Select the printer you want to use. Choose the Default Printer; or choose the Specific Printer option and select one of the current installed printers shown in the box. You install printers and configure ports using the Windows Control Panel.

Orientation

Choose Portrait or Landscape.

Paper Size

Select the size of paper that the document is to be printed on.

Paper Source

Some printers offer multiple trays for different paper sources. Specify the tray here.

Options

Displays a dialog box where you can make additional choices about printing, specific to the type of printer you have selected.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

INFORMATION SOCIETY TECHNOLOGIES PROGRAMME

VPLOT Help Guide

Biplot Visualisation

Edited by DAUPHINE

Project acronym: **ASSO**
Project full title: **Analysis System of Symbolic Official data**
Proposal/Contract no.: **IST-2000-25161**

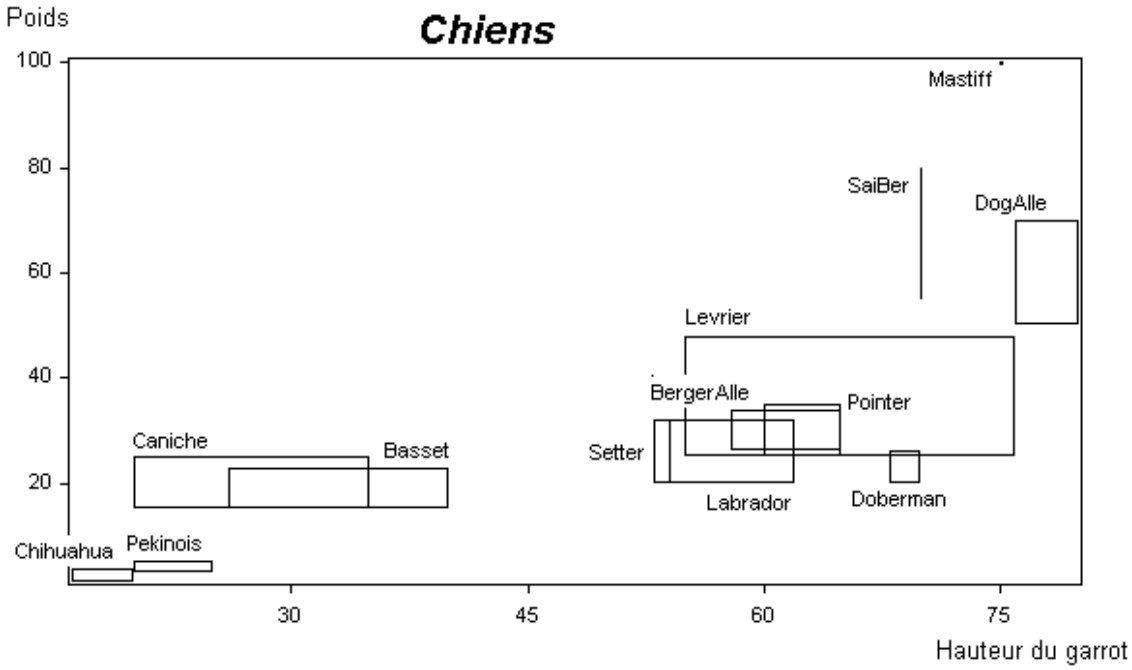
28 VPLOT : Biplot Visualisation

28.1 Introduction

VPLOT (V for Visualisation and PLOT for Biplot) is the module handling the biplot graph display for the following methods: DSTAT, SPCA, SGCA, SFDA, SMLP.

VPLOT works on interval variables from an ASSO symbolic objects file (there are exceptions for the SPCA method – see SPCA documentation).

Basically, it displays one rectangle for each symbolic object, the width and height of it being the intervals of the object for the variables retained as horizontal and vertical axis, as illustrated below:

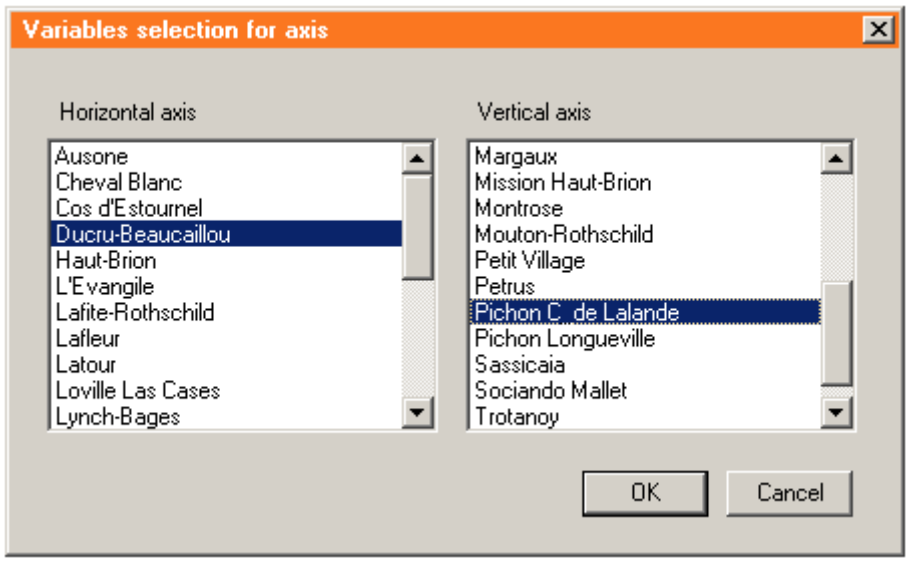


Many interactive features allow the user to tailor the graph to his needs.

28.2 Starting VPLOT

VPLOT is launched by the workbench when the user clicks on the graph icon. A command file is provided, which contains the name of the symbolic objects file to process (.SDS or .XML type file).

If a graph save file exists for this chaining from a previous sesssion, it is directly loaded and displayed. If not, the user is prompted for selecting the variables to be used as axis, via the following dialog:



It is possible to later change the selection at will using the relevant menu command.

Upon clicking on OK, the graph is displayed.

28.3 Saving and recalling the graphs

The VPLOT graph can be saved, to keep the user modifications. This does not mean saving a screen image, but saving the graph data and environment, in order to be able to resume the work when restarting VPLOT, as if it had not been interrupted.

Since a VPLOT graph is linked to a given SO file (.SDS or .XML), the saved graph will stay valid as long as this file is not modified, and will be automatically recalled when VPLOT is restarted for that file, if no change occured meanwhile; if the SO file has been modified, VPLOT will not use the save file and will delete it (to that purpose, the SO file date and time are recorded in the save file).

When a graph is automatically recalled, the user may wish not to use it, in which case he can start a new one using the New command.

The current saved graph is kept; it will be overwritten only if this new graph is saved.

The save files use the same name as the SO file, with the .VGS extension (which stands for VPLOT Graph Save).

28.4 Exporting the graphs

This allows to store the graph so that it can later be imported in other applications (Word, Excel, PowerPoint, Paint, IrfanView (to name a few).

Two types of files save the display image as such:

1. Windows bitmap (uncompressed pixel map).
2. Portable Network Graphic (lossless compressed pixel map).

The third type, Metafile, saves in a kind of vectorial mode, i.e. records the drawing commands (Windows Graphics Device Interface calls), so that the using application may "replay" those commands and draw the graph as if it had created it by itself. This is less widely supported than the two others.

28.5 Menu commands

In addition to the main menu and the toolbar, there are contextual menus that pop up, when applicable, upon clicking with the right button ; they are described in sections 6 to 9.

Hereunder, when a command of the main menu has an associated toolbar button, it is displayed next to the command label; similarly, if the function can also be accessed from a contextual menu, an asterisk (*) is appended to the label.

28.5.1 File

New 

Used for starting a new graph when already processing one.

May be used, e.g., if the saved graph has been automatically recalled and the user does not want to use it.

Calls a dialog box for selecting the variables to retain for the horizontal and vertical axis (same dialog box as when starting VPLOT without recalling a save file).

Save 

Saves the current data and environment, for later resuming the activity on the graph, as explained under "Saving and recalling the graphs" above.

Export as

This function saves the graph under one of the three formats explained under "Exporting the graphs" above.

Print 

Stores the bitmap of the graph and copies it to the printer device after rescaling.

Printer setup

If the default printer options do not match the user needs, this command allows to change them for the whole session instead of changing them at each print request.

This does not change the default printer options, nor the options selected by other applications, and is lost when VSTAT ends.

Exit

Ends VSTAT, and prompts for saving the current graph (same as Save above).

28.5.2 Edit

Copy 

Copies the entire graph bitmap to the clipboard.

Paste

Pastes the contents of the clipboard (bitmap) onto the graph.

The bitmap is initially placed at the top left corner of the graph, but may be dragged with the mouse to adjust its position. And it is possible to move it later at any time in the same way.

28.5.3 View

Tool Bar

Shows or hides the Tool Bar (flip/flop)

Status Bar

Shows or hides the Status Bar (flip/flop)

28.5.4 Variables

Select variables 

Allows to change the variables selection and restart a new graph.

The selection dialog is the same as described under «Starting VPLOT».

The new graph is displayed upon clicking on OK.

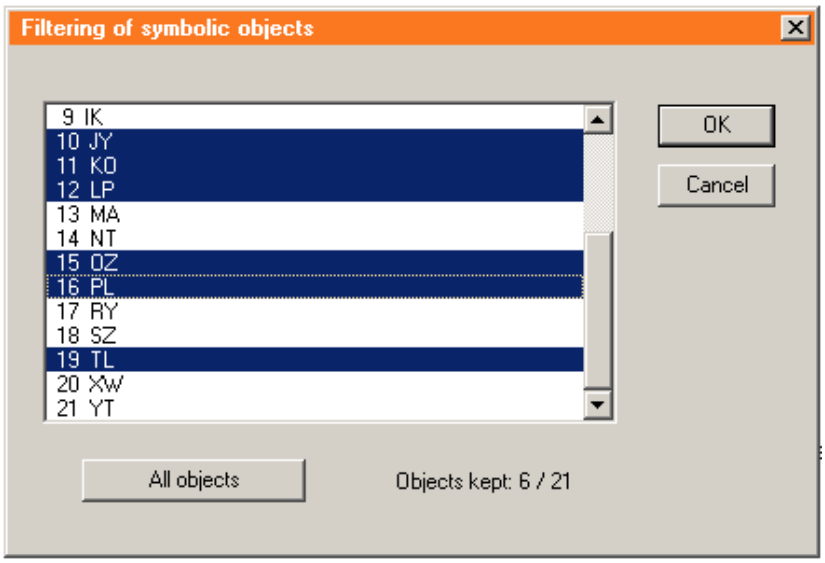
N.B.: contrary to the New command, the new graph there does not use the default customization options; it reuses what was current before the reselection; also, the titles and various texts inserted by the user are kept.

28.5.5 Objects

Filter objects

This allows to retain only part of the symbolic objects for the biplot graph, when there are too many for a readable graph, or when the user wishes to concentrate on a given subset.

The following dialog is used:



28.5.6 Draw

This popup menu deals with the graph layout and display features.

Some commands depend on the type of graph processed; they are disabled when not applicable.

Force origin

Directs VPLOT to include the origin in the graph even if no object interval encompasses it.

Only applies to the «Frequencies for interval variables» type of graph.

This is a flip/flop option.

Hide scales

Removes the scale tags.

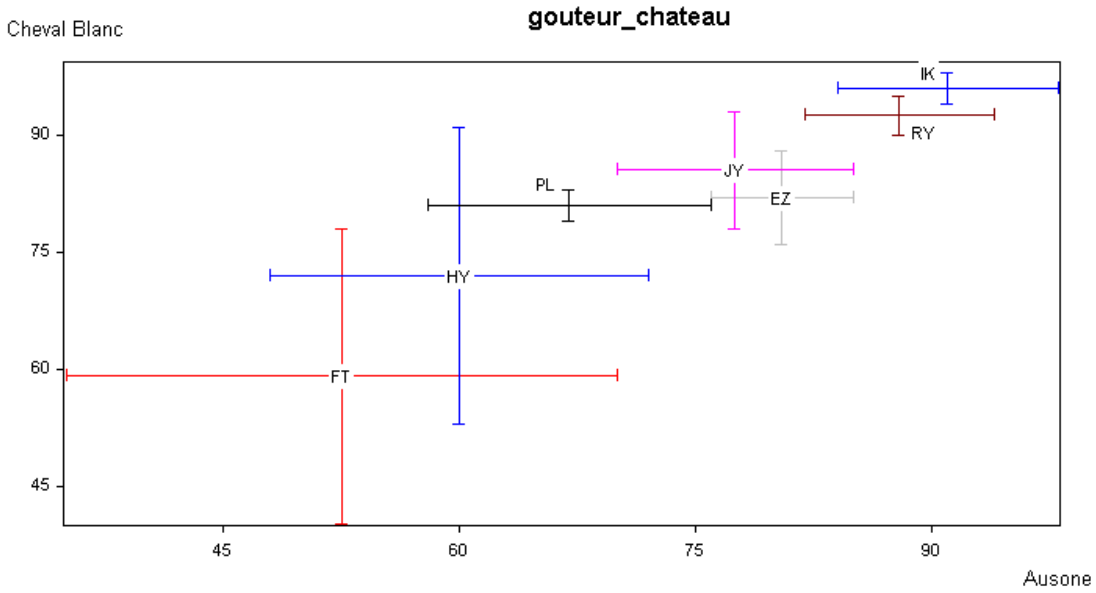
This is a flip/flop option.

Draw objects as crosses / Draw objects as rectangles 

By default, the objects are drawn as rectangles (see Introduction).

In some instances, the graph may be more readable if they are drawn as crosses, the center and the dimensions of the cross being the same. This function allows to do so.

This is a flip/flop option. The command label displayed is the one of the alternate mode, i.e. the cross mode when the current mode is the rectangle, and vice-versa.



Labels

This popup menu deals with the object labels display.

The labels are displayed by default, unless there are so many objects that the graph would likely be unreadable.

Display: displays all the object labels.

The initial position is over the top-middle, but the labels can later be dragged individually using the mouse. See also Center below.

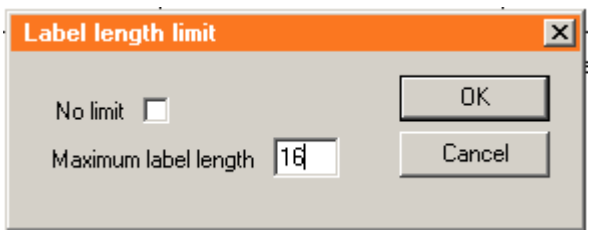
Center: displays the labels at the center instead of over the top-middle.

Reset: since the labels can be moved with the mouse, this command is for resetting all the back to their base position (top-middle or center, depending on the current option).

Erase: removes all the labels.

Set lenght limit: in case of long labels causing some overlaps or jams, allows to enforce a maximum displayable length.

The following dialog is used:



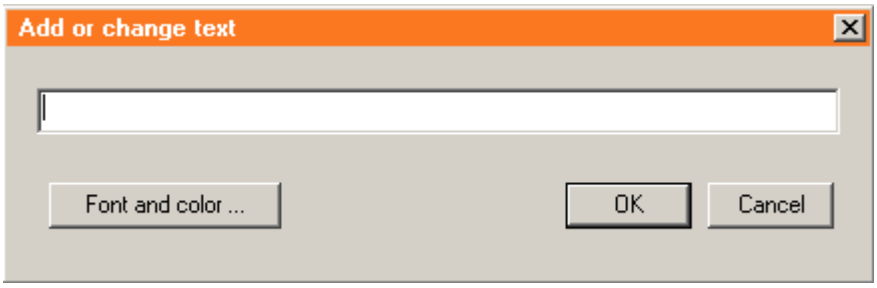
Set transparent mode: the transparent mode allows that the characters only override strictly the pixels necessary to draw them, instead of clearing the enveloppe rectangle of the text («opaque mode», which is the default).
When the transparent mode is selected, the label changes to «Set opaque mode», and vice-versa.

28.5.7 Insert a text

This command allows to insert a text anywhere in the graph area.

When it is selected, the mouse cursor changes to a cross; the user has to move the cursor at the insertion point (top left corner of the text) and click there (or cancel with the Escape key).

The following dialog shows up:



The «Font and color» button calls the standard Windows dialog for selecting another font and/or another text color.

Once displayed, the text can be moved to adjust its position, by clicking on it then moving the mouse while holding down the left button (dragging). It is possible to move it later at any time in the same way.

N.B.: When a text is moveable, the mouse cursor changes to a quadruple arrow when passing over it, so indicating that possibility.

All the texts inserted by the user are moveable, as well as those inserted by VPLOT; the scale tags, linked physically to the graph, are not.

A text can be changed via its contextual menu, which pops up when right clicking on it (see section 8).

The same dialog as above is used.

Redraw 

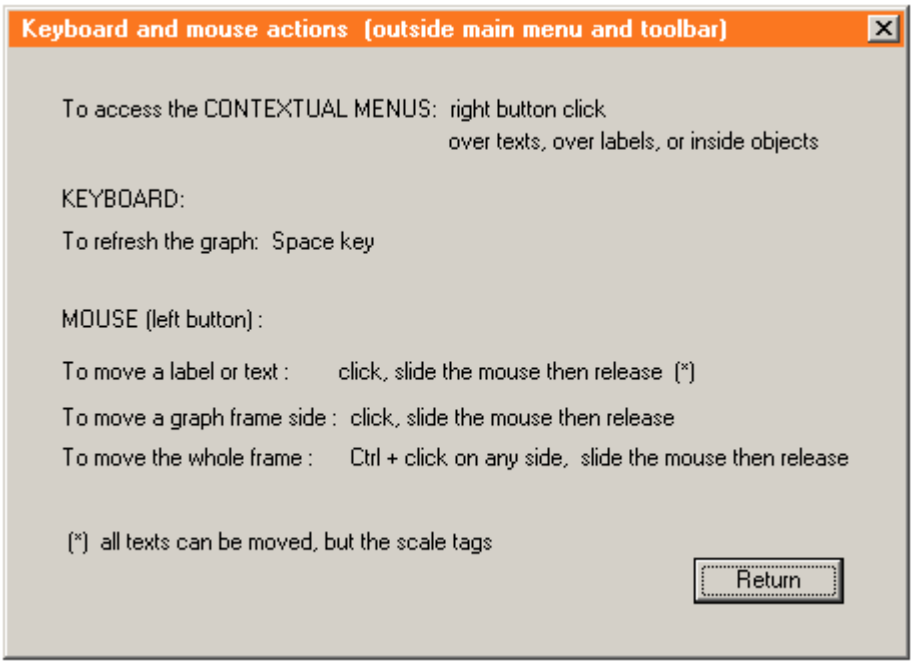
This command clears the display and redraws the graph.

Hitting the space bar does the same.

28.6 Help

Keyboard and mouse actions

A few functions are directly triggered from the keyboard and the mouse.
This command displays the following panel, summarizing those possibilities:



Direct keyboard and mouse commands

See the above main menu command description.

Contextual menu for objects

When clicking inside an object rectangle (or over the cross), one of the following menus will pop up; they only differ from the fact that the object label is displayed or not:

Display label / Delete label

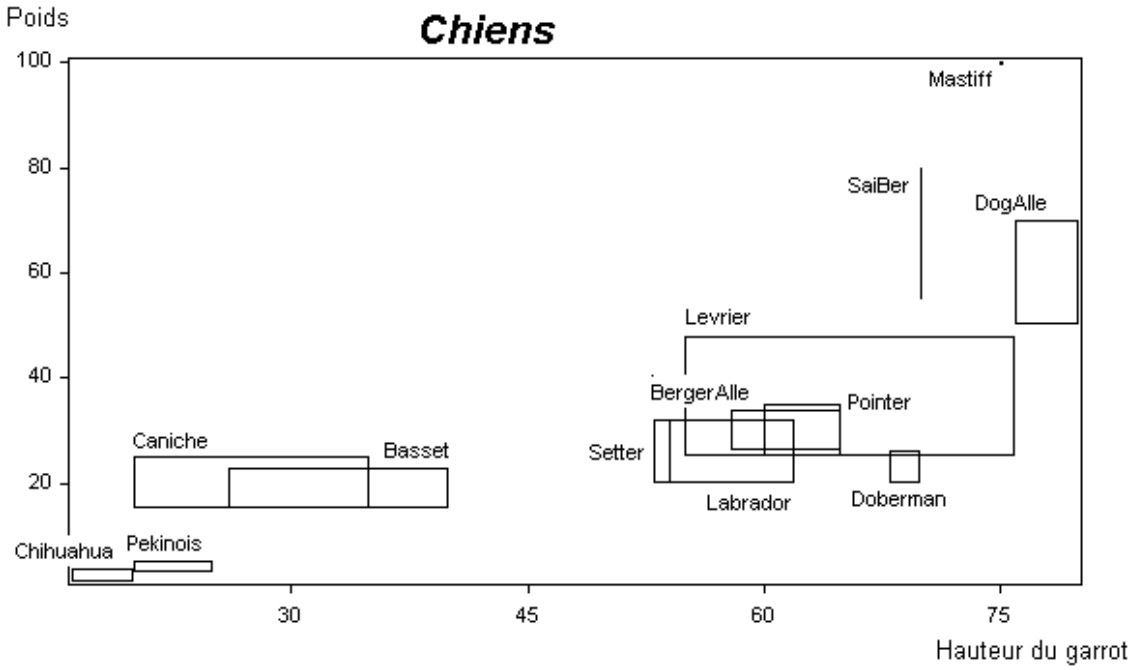
Self-explanatory.

Display item coordinates

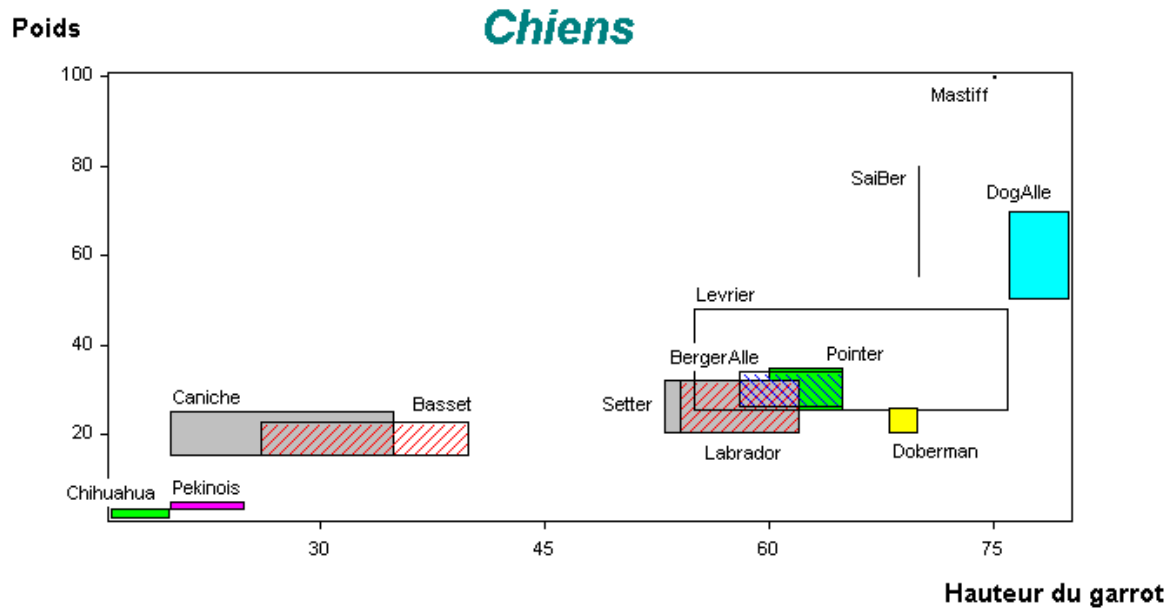
Displays a window containing the interval limits of the object over both axis.

Change color or pattern:

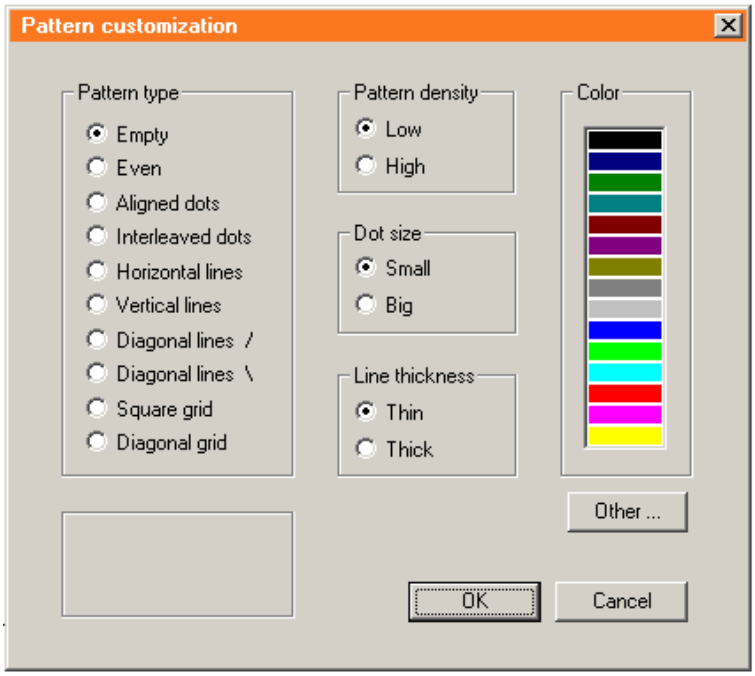
Basically, the objects are drawn with a thin black solid line:



It is possible to fill the rectangles with various patterns and colors, and also to change the rectangle outline color and draw mode:



To do that, the following dialog is called by this contextual menu command:

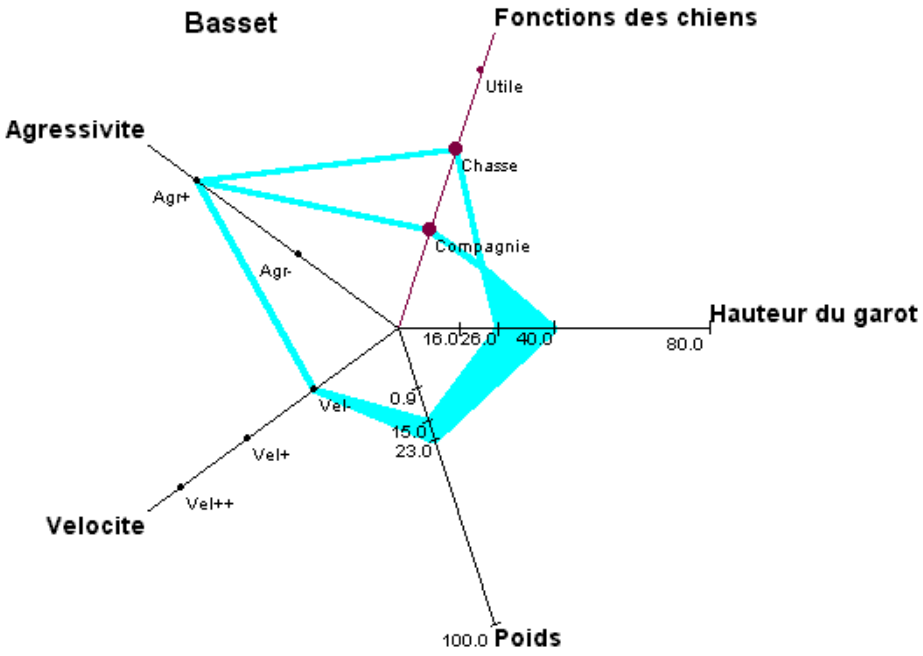


The various options of this dialog are self-explanatory.

The «Other» button under the colors list calls the standard Windows dialog allowing to create custom colors.

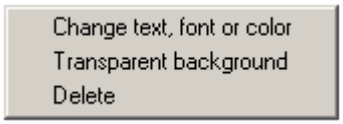
Display STAR graph

Calls the module STAR, which displays a star-shaped graphical representation of the object values for the different variables (see the STAR module Help) :



Contextual menu for texts

This covers all texts but the scale tags, which must not be modified nor deleted, and the objects labels, which are managed differently (see next section).



Change text, font or color

The same dialog as for inserting a text is used (see «Insert a text» in section 3).

Transparent background

In the transparent background mode, the characters only override strictly the pixels necessary to draw them, instead of clearing the enveloppe rectangle of the text («opaque mode», which is the default).

When the transparent background is selected, the label changes to «Opaque background», and vice-versa.

28.7 Delete

Self-explanatory.

Contextual menu for objects labels

Display item coordinates

Displays a window containing the interval limits of the object over both axis.

Display STAR graph

Calls the module STAR, which displays a star-shaped graphical representation of the object values for the different variables (see the STAR graph example in section 7).

Delete label

Self-explanatory.

Delete

Self-explanatory.