Российская Академия Наук

РОССИЙСКАЯ АКАДЕМИЯ НАУК

ИНСТИТУТ ПРОБЛЕМ БЕЗОПАСНОГО РАЗВИТИЯ АТОМНОЙ ЭНЕРГЕТИКИ

RUSSIAN ACADEMY OF SCIENCES

NUCLEAR SAFETY INSTITUTE

Препринт ИБРАЭ № IBRAE-2002-07

Preprint IBRAE-2002-07

M. Kanevski, S. Chernov, V. Demyanov, E. Savelieva, V. Timonin, A. Trouttse

GEOSOM APPLICATION FOR DATA ANALYSIS USING THE SELF-ORGANIZING MAPS

Москва
2002

Moscow
2002

УДК 502.3

Каневский М.Ф., Чернов С.Ю., Демьянов В.В., Савельева Е.А., Тимонин В.А., Трутце А.А. ПРОГРАММНОЕ ПРИЛОЖЕНИЕ GEOSOM ДЛЯ АНАЛИЗА ДАННЫХ С ИСПОЛЬЗОВАНИЕМ САМООРГАНИЗУЮЩИХСЯ КАРТ КОХОНЕНА. (На англ. яз.). Препринт № IBRAE-2002-07. Москва: Институт проблем безопасного развития атомной энергетики РАН, 2002. 15 с. — Библиогр.: 2 назв.

Аннотация

В работе описывается программное приложение GeoSOM, разработанное и реализованное для анализа данных с помощью Самоорганизующихся Карт Кохонена. Содержится теоретическое описание реализуемого метода. Приводится пример анализа данных при помощи приложения GeoSOM.

Kanevski M., Chernov S., Demianov V., Savelieva E., Timonin V., Trouttse A. GEOSOM APPLICATION FOR DATA ANALYSIS USING THE SELF-ORGANIZING MAPS. Preprint IBRAE-2002-07. Moscow: Nuclear Safety Institute RAS, 2002. 16 p. — Refs.: 2 items.

Abstract

In this work the program application GeoSOM described which was developed and realized to data analyze using Self-Organizing Kohonen's Maps. The paper contains the theory of SOM method which had been realized. There is given the example of data analysis using GeoSOM application.

GeoSOM application for data analysis using the Self-Organizing Maps

M. Kanevski, S. Chernov, V. Demyanov, E. Savelieva, V. Timonin, A. Trouttse

ИНСТИТУТ ПРОБЛЕМ БЕЗОПАСНОГО РАЗВИТИЯ АТОМНОЙ ЭНЕРГЕТИКИ
113191, Москва, ул. Б. Тульская, 52
тел.: (095) 955-26-20, факс: (095) 230-20-29, эл. почта: dargot@ibrae.ac.ru

Content

1 Introduction

1.1 Introduction

Nowadays because of great development in the informational technologies there is an opportunity to collect huge data sets, for instance concerning the environmental pollution. Visualizing and analysis such data sets are difficult and require the special methods development. One of the alternatives is Self-Organizing Kohonen's Maps (SOM)[1]. Basing on data classification on internal relations they allow to decrease the dimension of data space and to simplify data interpretation. In some cases SOM can be applied to spatial data analysis[2].

Special application was developed to provide data analysis using SOM method. Because of its meaning to analyze the geostatistical and time series data this application was called GeoSOM.

1.2 Disclaimer

GeoSOM application is created with the help of Borland C++ 5.0 and does not contain any instructions leading to loss or damage of the data, except the one loaded in the operative memory. Only incorrect set-up or usage can initiate all the conflicts with other software.

Any use of the programs in situations that could result in personal injury or property loss is done at the user's own risk. The authors disclaim all liability for direct or consequential damages resulting from use of the GeoSOM software.

2 Basic theory of self-organizing maps

2.1 Brief idea of the algorithm

SOM is a variant of ANN competitive learning. The basic idea of this method is in arrangement the neurons (nodes) into x - dimensional array. A vector (weights of neuron) in the dimension of the original data space is assigned to each node. During the training of a map not only the weights of the winner neuron are modifying (as in case of simple competitive learning). With the weights of winner neuron the weights of its neighbors in the original array are modifying. This process can be imagined as stretching the elastic net on the training data set.

Hence it is obtaining that the vectors which assigned to the neighbors in our two-dimensional array are neighbors in the data space.

Therefore it is obtaining not only the input data set quantization but regulating the input data set in two- (or one-) dimensional map.

2.2 Initializing and learning procedures

Let R^n be an n - dimensional input vector space. Let R^2 be a two-dimensional array of neurons. $r \in R^2$ is called node of Kohonen's map. The type of the array can be different, but rectagonal or hexagonal are the most useful.

Assign $m_i = [\mu_{i1}, \mu_{i2},..., \mu_{in}]^T$, $m_i \in R^n$, to each $r_i \in R^2$. A m_i is called a reference vector of r_i. The reply of the net on input vector $x \in R^n$ is the winner node $c \in R^2$ such that the distance from c to x is minimal for all $r_i \in R^2$. The distance can be defined by different means, but usually Euclidean $|x-m_c|$.

In order to choose the winner node c vector x compared with all reference vectors m_i. and $c=argmin_i\{|x-m_i|\}$, i.e.

$$|x-m_c|=min_i\{|x-m_i|\}. \quad (1)$$

The training starts with the starting reference vectors' definition. Usually they are defined as random values in the range of corresponding coordinate of the input data set.

During the training the vector m_i which refers to node i changes its meaning corresponding to the input vector $x(t)$ which is given to the net in moment t by the formula:

$$m_i(t+1)= m_i(t)+h_{ci}(t)[x(t)- m_i(t)], \quad (2)$$

where t is a discret time coordinate, and $h_{ci}(t)$ is a neighborhood function. The neighborhood function takes central part in the training process. It is necessary that $h_{ci}(t) \to 0$ under $t \to \infty$. Usually:

$$h_{ci}(t)=h(|r_c-r_i|,t), \quad (3)$$

where $r_c \in R^2$ and $r_i \in R^2$ are vectors of nodes c and i correspondingly. As $|r_c-r_i| \to \infty$, $h_{ci} \to 0$. Mostly two simple variants for $h_{ci}(t)$ are used.

First of them – "bubble neighborhood". Let N_c be the set of indexes of nodes which belongs to neighborhood of a winner node with radius $R(t)$. If $i \in N_c$ then $h_{ci}(t)=\alpha(t)$; else $h_{ci}(t)=0$:

$$m_i(t+1)= m_i(t)+ \alpha(t)[x(t)- m_i(t)], \; i \in N_c$$

$$m_i(t+1)=0, \; i \notin N_c. \quad (4)$$

The $\alpha(t)$ is said to be learning rate ($0<\alpha(t)<1$). $\alpha(t)$. $R(t)$ decreasing monothonically as $t \to \infty$.

The second popular variant of the $h_{ci}(t)$ definition is in the form of Gauss function:

$$h_{ci}(t)=\alpha(t)*exp(-|r_c-r_i|^2/2\sigma^2(t)), \quad (5)$$

where $\alpha(t)$ is a learning rate and $\sigma(t)$–parameter representing the radius of the neighborhood. $\alpha(t)$ and $\sigma(t)$ are both decreasing monothonically as $t \to \infty$.

It is possible to divide the process of training into two steps (ore more). During the first step reference vectors are ordered. During the second step reference vectors are tuned. Usually second step has slower learning rate than first. Both steps go on for a pre-defined number of iterations T. Usually $\alpha(t)$ is defined in order to $\alpha(T)=0$, for example $\alpha(t)= \alpha(1-t/T)$, where α – pre-defined constant.

2.3 Data analysis procedure

During the analysis of data the vector x to be processed is given on the input of trained net. As the reply of the net on this vector the node c is obtained. Comparing the x with m_c and reference vectors of the c neighborhood we can make some conclusions about vector x, classify him, repair missing data and so on. Because of the fact, that analyzed data easily can be shown on two- (or one-) dimensional map, the process of data analysis by SOM sometimes called data visualization.

3 GeoSOM application description

3.1 What is GeoSOM?

GeoSOM is the program application for Windows 95/98/2000 OS to analyze data using Self-Organizing Kohonen's Maps (SOM). It provides:

- working with several maps simultaneously
- creating the map with needed architecture
- initializing the created map using needed parameters
- training the initialized maps with different parameters several times and automatical choosing the best variant
- data analysis using trained map

GeoSOM is designed to function integrally with GeoStat Office software and fully compatible with it. User should use GeoStat Office to data preprocessing and visualization.

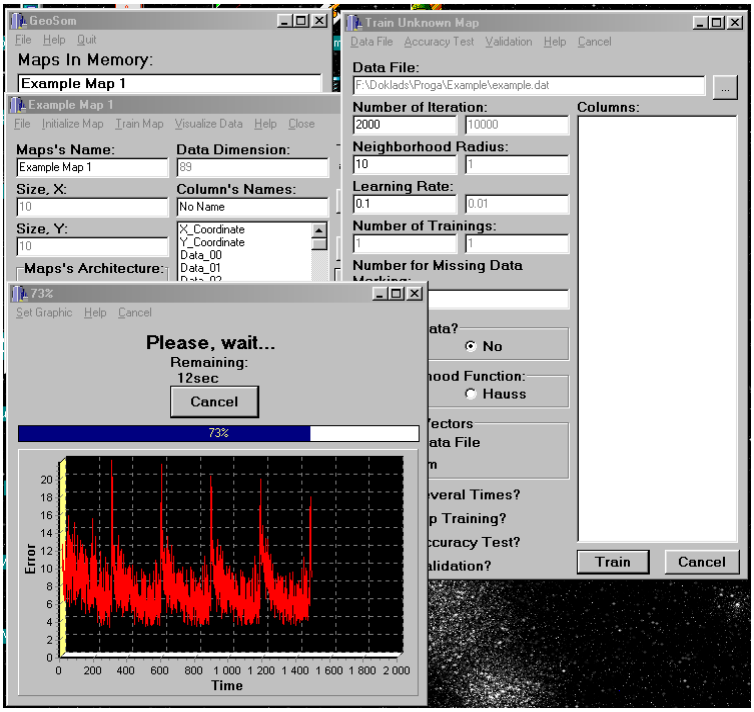


Fig. 1. GeoSOM sample screenshot

Above it can be seen the the screenshot of GeoSOM.

This chapter is organised according to different windows of GeoSOM user interface which correspond to its functions:

- the Main Window which provides creating the new map, loading the saved map and switching between maps are being worked at the moment
- the Map Windows which provide work with created or loaded map

- the Initialise Map Window which provides initialising the map with needed parameters
- the Train Map Window which provides the map's training wit needed parameters
- the Expectation Window which provides the showing the progress of map's training
- the Analyse Data Window which provides data analysis using the trained map

3.2 Main window

The main window is the first window that opens when starting GeoSOM. The function of the main window is helping the user to switch between maps in computer's memory. It contains the list of maps to work (the list with «Maps In Memory» title). Each map in computer's memory (in other words, each map that user can initialize, train or analyze data by it) is in this list. When a new map is loaded from disk or created by user its name appears in this list. When a map is closed its name disappears from the list. Double-clicking on the map's name in this list activates the Map Window corresponding the clicked map.

Besides this window contains the upper menu with the following items:
File:

- New Map – create a new map and show its Map Window.
- Load Map – load saved map and show its Map Window.
- Quit – quit the GeoSOM.

Help:

- About – show some information about the GeoSOM.

Quit: quit the GeoSOM. User is questioned about saving the unsaved maps. The same question is asked when [×] button is pressed.

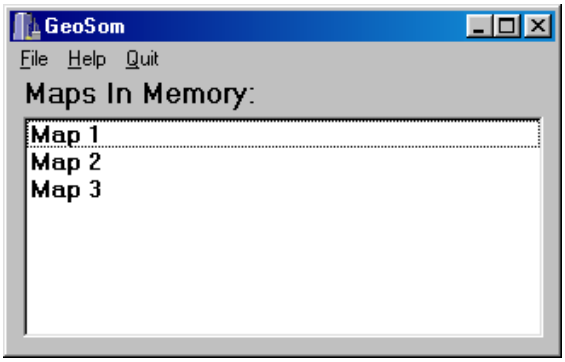


Fig. 2. Main window example

3.3 Map window

Each map in the computers memory is connected with a Map Window. Its function is to create new map or to provide work with created map. New Map Window is created in two cases: when New Map item selected or Load Map item is selected (both in upper menu of Map Window).

Map window contains:

- «Map's Name» editbox - Editbox for changing of the map's name.
- «Size, X» and «Size, Y» editboxes - Editboxes for entering the map's size.
- «Data Dimension» editbox - Editbox for entering the dimension of the processing data.
- «Map's Architecture» group that contains two radiobuttons - Group of radiobuttons for selecting the type of map's nodes' array.
- «Column's Names» editbox and list - List of the map's columns names and editbox for changing the name of selected column.

- «Take» button with comment «Take Data Dimension and Columns' Names from File» - Button for loading the data dimension and columns' names from file with data to processing.
- «Initialize Map» button - Button for map's initializing.
- «Train Map» button - Button for map's training.
- «Visualize Data» button - Button for data analysis using the trained map.
- «Close Map» button -Button for closing this window.
- The upper menu contains the following items:

File:

1. New Map – create new map. New map will be created in this Map Window. If user hasn't saved the current map he will be asked about it.
2. Load Map – loads map from file. Map will be loaded in this Map Window. If user hasn't saved the current map he will be asked about it.
3. Save Map – save current map to file.
4. Close – close this window. If user hasn't saved the current map he will be asked about it.

Close: close this window. If user hasn't saved the current map he will be asked about it.

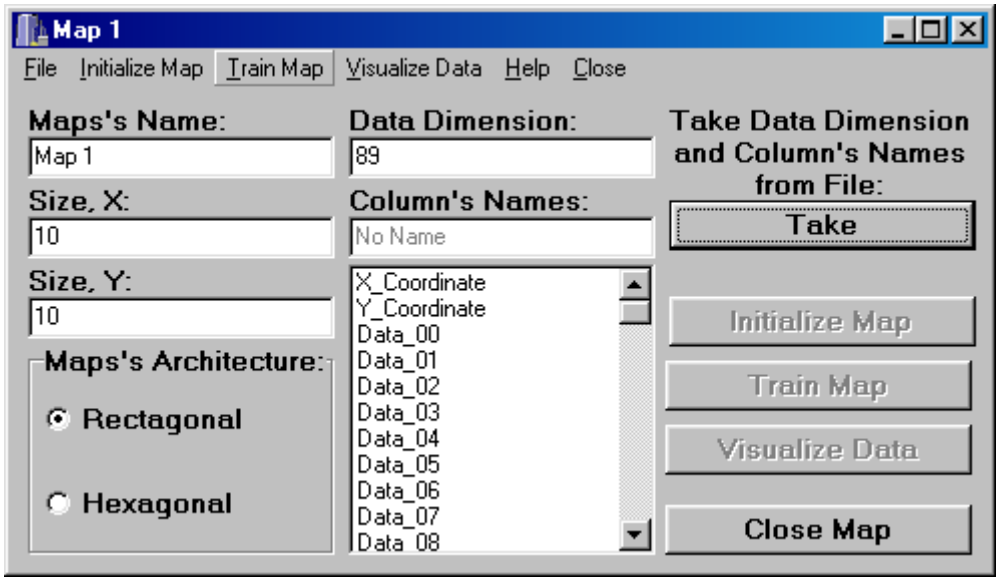


Fig. 3. Map window (Creating New Map mode)

Map window has two modes. First of them showed on Figure 3 and called Creating New Map Mode. Buttons «Initialize Map», «Train Map» and «Visualize Data» are disabled in this mode and this mode is to enter map's architecture, size and data dimension. There is two cases when the Map Window switches in this mode:

- When the Map Window is just opened by choosing on «New Map» menu item in Main Window.
- When «New Map» menu item in this map has been chosen.

The both of them are when new map has not been created and it is necessary to set its parameters. After that it is necessary to press «Enter» and new map will be created. Then Map Window switch to second mode (Figure 4).

In this second mode (called Working with the Map) user is not allowed to change the map parameters but user can initialize map, train map and analyze data using the corresponding buttons.There are two cases when the Map Window switches in this mode:

- When new map has been created by pressing «Enter» in Creating New Map Mode.
- When saved map has been loaded by choosing «Load Map» menu item in Main Window or in Map Window.

When new map has been created all of its reference vectors are initialized zero. In the most cases it is necessary to initialize them according to training data. User can open Initialize Map Window by pressing the «Initialize Map» button. Besides this user can re-initialize any map. In this case user will be questioned to save map (if unsaved).

There is a special window for map's training – Train Map Window which opens by pressing «Train Map» button. If map isn't saved user will be questioned about it.

To visualize data by trained map user must open Visualize Data Window by pressing «Visualize Data» button.

User can close this window by pressing «Close Map» button. If map isn't saved user will be questioned about it.

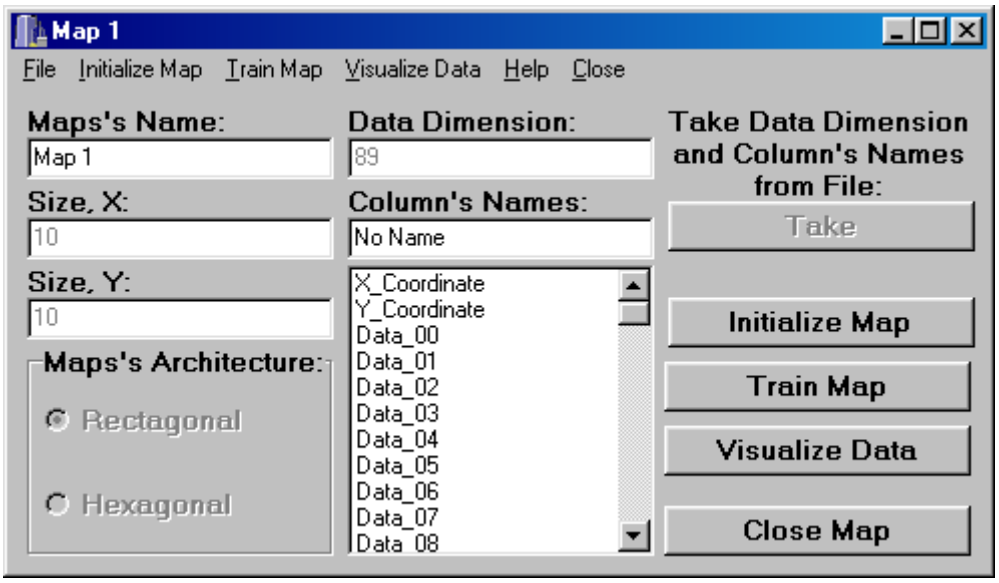


Fig. 4. Map window (Working with the Map mode)

3.4 Initialize map window

The function of Initialize Map Window is to set the parameters of initialization and the file with initializing data.

Initialize Map Window contains:

- «Data File» editbox - Editbox for entering the name of file with initializing data.
- Button «…» for opening dialog for setting of initializing data file.
- «Set Initializing Mode:» group that contains two radiobuttons - Group of radiobuttons for selecting the method of initialization (Random vectors from data set / Random vectors in range of data set). In first mode all the map's reference vectors will be initialized by random vectors from initialized data set. In second mode each component of every reference vector will be initialized by random meaning in range of the minimum and maximum values of this component in vectors of initializing data set.
- «Missing Data?» group that contains two radiobuttons - Group of radiobuttons for selecting the present of missing data (if there is a missing data in data set then some components of some vectors in data set aren't known) in processing data set.
- «Number for Missing data Marking:» editbox - Editbox for entering the value to missing data marking.
- «Initialize» button - Button for map's initialization.
- «Cancel» button - Button for closing this window and returning to Map Window.

User needs to enter the name of file with initializing data for map's initialization. If this data contains missing data values it is necessary to enter the value for their marking. In this case only second method of initialization is allowed.

After setting all parameters it is necessary to press «Initialize» button. Then the map will be initialized. By pressing «Cancel» button user can cancel the map's initialization and return to Map Window.

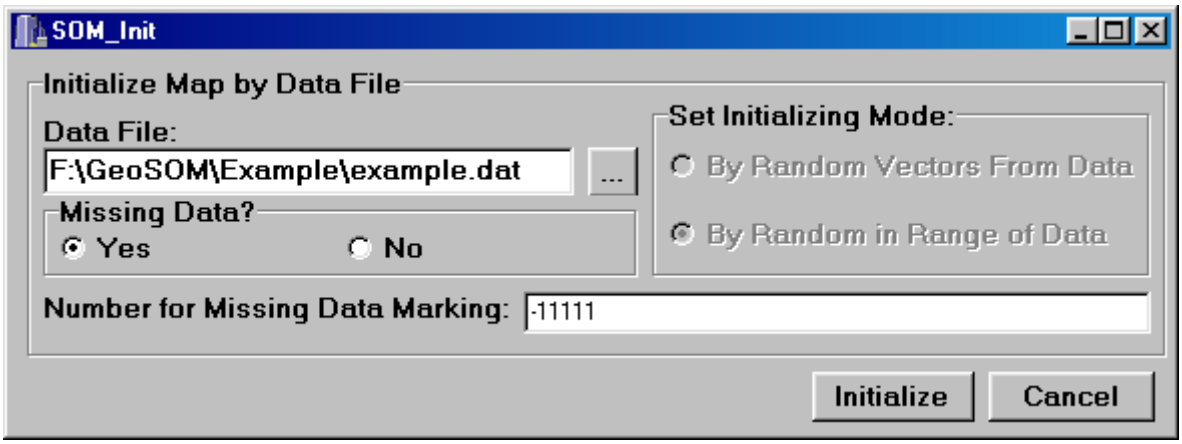


Fig. 5. Initialize map window

3.5 Train map window

The function of Train Map Window is to set the parameters of training and the file with training data. Train Map Window contains:

- «Data File» editbox - Editbox for entering the name of file with training data.
- Button «…» for opening dialog for setting of training data file.
- «Learning Rate:» editbox - Editboxes for entering the learning rate.
- «Neighborhood Radius:» editbox - Editboxes for entering the neighborhood radius.
- «Number of Iterations:» editbox - Editboxes for entering the number of training steps.
- «Number of Trainings:» editboxes - Editboxes for entering the number of times you want to train the map to select the best training variant.
- «Missing Data?» group that contains two radiobuttons - Group of radiobuttons for selecting the present of missing data in processing data set.
- «Neighborhood Function:» group that contains two radiobuttons - Group of radiobuttons for selecting the neighborhood function (Bubble/Gaussian).
- «Order of Vectors» group that contains two radiobuttons - Group of radiobuttons for selecting the order of using vectors from training data set during the map's training (as in data file/random).
- «Number for Missing data Marking:» editbox - Editbox for entering the value to missing data marking.
- «Column's Names» list - List of the training data's columns' names.
- «Make Accuracy Test?» checkbox - Checkbox for marking accuracy test making. If this checkbox is marked then after the training of map training data will be analyzed by trained map to investigate the training's accuracy and results will be saved in file.
- «Make Validation?» checkbox - Checkbox for marking validation making. If this checkbox is marked then after the training of map data from file with validation data set will be analyzed by trained map to investigate the training's accuracy and results will be saved in file.
- «Train Several Times?» checkbox - Checkbox for enabling the several times training mode – in this mode you can enter the number of training times you want and GeoSOM choose the best training map.

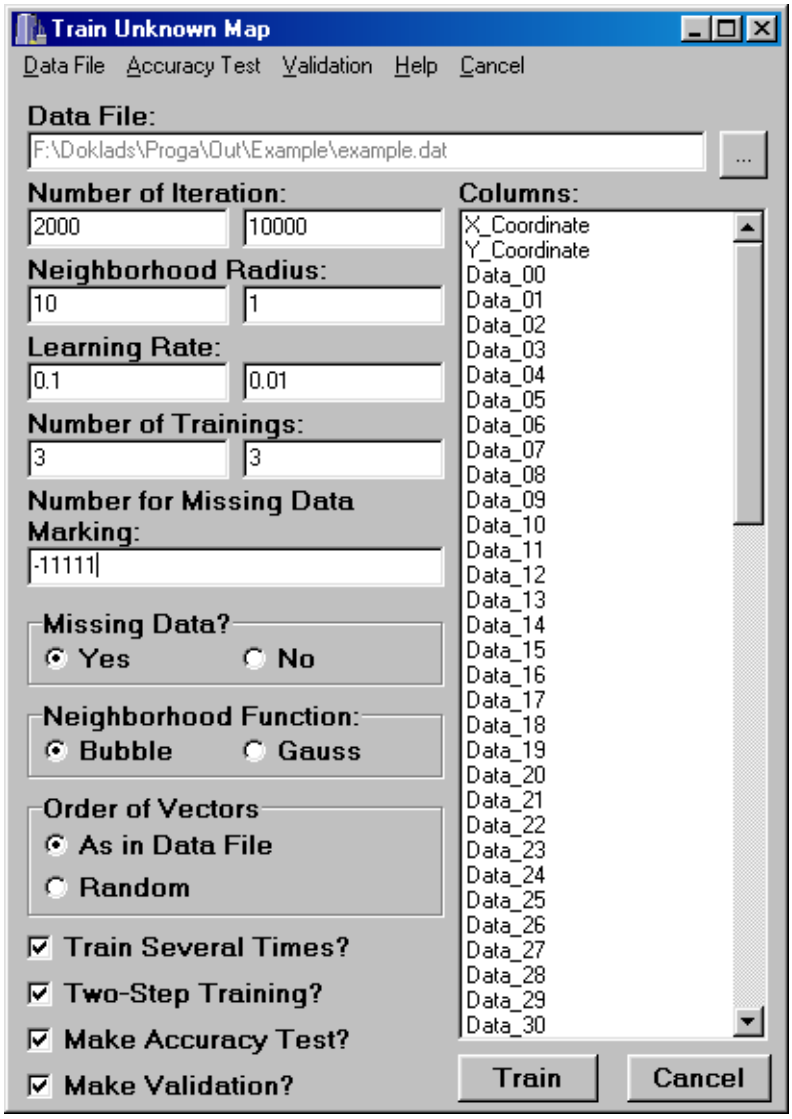


Fig. 6. Train map window

- «Two-Step Training?» checkbox - Checkbox for enabling the two-step training mode – in this mode iech time when the map is training this process contain two step – first – rough training and second – tuning. During the first step number of iterations usually less and radius and learning rate are greater than during the second.
- «Train» button - Button for map's training.
- «Cancel» button - Button for closing this window and returning to Map Window.

For map's training user must enter the desired parameters and press «Train» button. Then this map will be closed and Expectation Window will be opened. User can train several maps simultaneously.

If user checks the «Make Validation Test?» checkbox he will be asked about file with validation data and file to save the validation results. If user checks «Make Accuracy Test?» checkbox he will be asked about file to save the accuracy test results.

By pressing «Cancel» button user can cancel the map's training and return to Map Window.

3.6 Expectation window

Expectation window is to show the training progress. It contains:

- Pie indicator of the remaining part of the training iterations.
- Progress bar of the training.

- Diagram Error/Time.
- Indicator of the remaining time.
- «Cancel» button.
- Upper menu that contains following items:

Set Graphic:

1. Progress Bar – Progress bar on/off.
2. Pie – Pie indicator on/off.
3. Graphic – Diagram on/off.

Cancel – cancel the map's training and close this window.

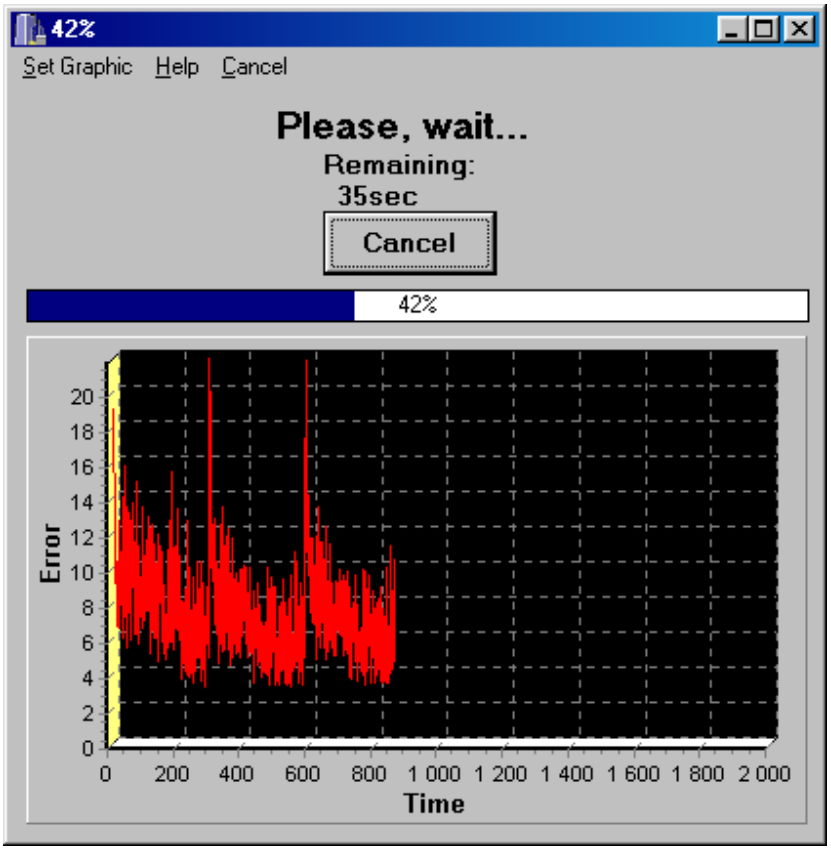


Fig. 7. Expectation window

If user tries to stop the training process he will be asked for confirmation. If you cancel the training process then all map's changes will be cancelled. Then this window will be closed and Map Window will become active. The same happens after the end of training.

3.7 Data analysis window

- Data Analysis Window is to data analysis by trained map. It contains:
- «Visualizing Data File» editbox - Editbox for entering the name of file with analizing data.
- Button «…» for opening dialog for setting of visualizing data file (upper «…» button).
- «Output File» editbox - Editbox for entering the name of output file with results of data analysis.
- Button «…» for opening dialog for setting of output file (lower «…» button).
- «Add Visualizing Vectors?» checkbox - Checkbox for setting whether to add or not the analyzing data to output file.

- «Add Reference Vectors?» checkbox - Checkbox for setting whether to add or not the map's reference vectors to output file.
- «Is There Missing Data?» checkbox - Checkbox for marking the present of missing data in analyzing data.
- «Number for Missing data Marking:» editbox - Editbox for entering the value to missing data marking.
- «Visualize» button - Button for data analyzing.
- «Cancel» button - Button for closing this window and returning to Map Window.

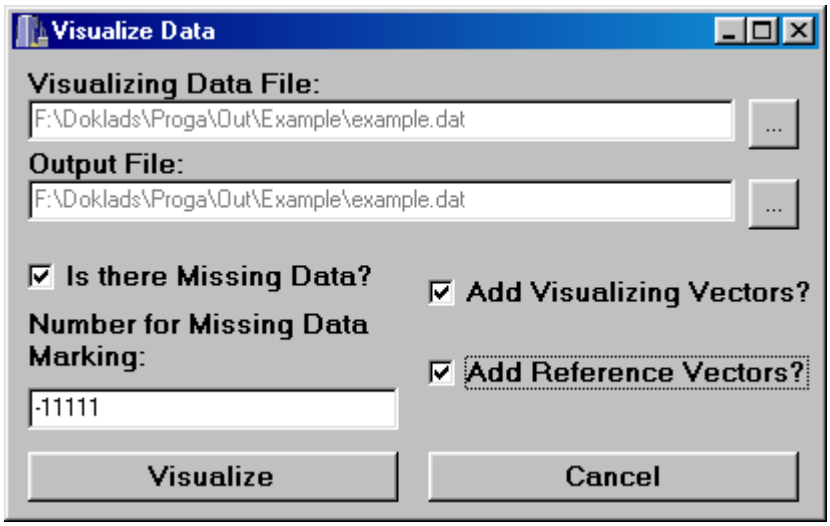


Fig. 8. Data analysis window

Without marking the «Add Visualizing Vectors?» and «Add Reference Vectors?» checkboxes only the map's coordinates X and Y and analysis error will be saved in output file. Each vector in output file corresponds with vector in analyzing data set. User can add analyzing vectors and reference vectors to output file by marking corresponding checkboxes. If both checkboxes are marked difference between analyzing vectors and reference vectors will be added as well.

User must press the «Visualize» button to analyze data and return to Map Window.

By pressing «Cancel» button user can cancel the data analyzing and return to Map Window.

4. Conclusion

As a result of this work the GeoSOM application had been developed and reaized.GeoSOM applicaton may be a helpful tool. Further work on its improvement is planned when there will be enough practical experience of applying it to different problems.

5. Acknowledgments

This work was performed with partially support of INTAS 99-00099 grant and grant for young scientists Russian Academy of Sciences.

6. Literature

1. T. Kohonen. Self-Organizing Maps. Springer-Verlag, Berlin, Heidelberg, 1995.
2. Trutse A.A., Savelieva E.A., Demyanov V.V., Kanevski M.F., Timonin V.A., Chernov S.Yu. Self-Organizing Maps Application to Classification and Analysis of Spatialy Distributed Environmental Non-Full Data (in Russian). Preprint IBRAE-99-10. Moscow: Nuclear Safety Institute, December 1999.

7. Appendix: the example of GeoSOM's practical applicaton

Let's imagine that there is a need to analyze some data. Example spatial data has more than 80 correlated variables.

Before analysis you should initialize and train the map. Following all actions are step-by-step.

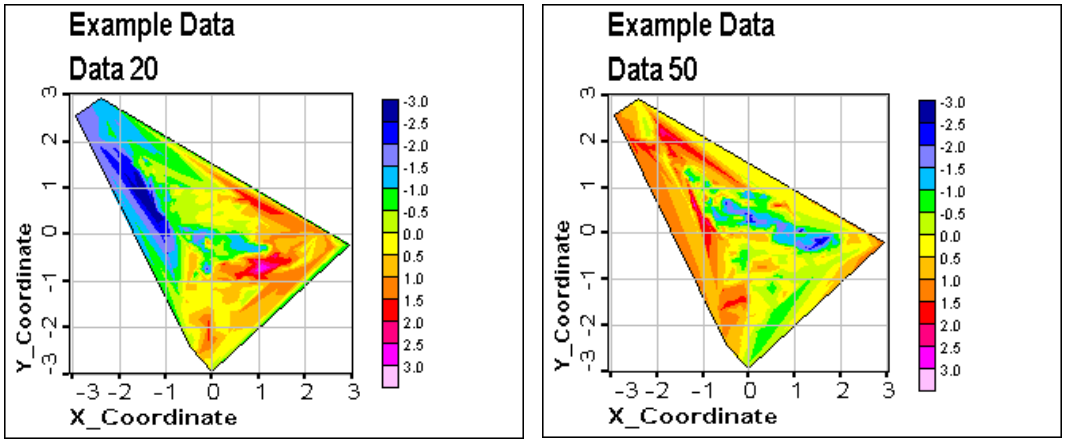
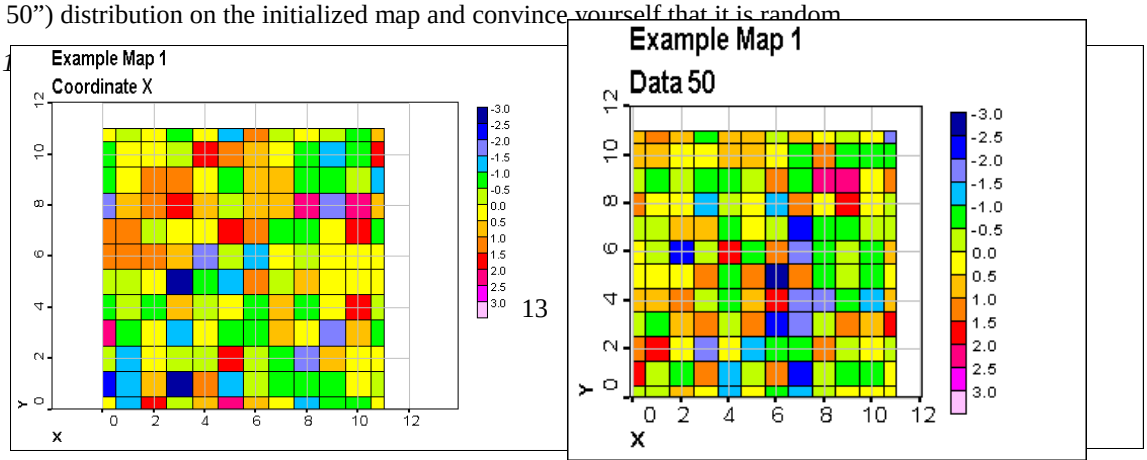


Fig. 9. Distribution of the "Data_20" and "Data_50" data components in processing data set

- Start GeoSOM
- Select the "New Map" menu item then Map Window will be showed (see Chapter 3.2 – Map Window in program description).
- Press «Take» button and choose in appeared dialog file with GEO-Eas data ("Example\example.dat" for example). After dialog's closing «Column's Names» list will be filled by columns' names from this file. In «Data Dimension» editbox will appear dimension of data in this file (89 for "example.dat").
- If you wish to change the map's size or architecture you may enter the desirable values for map's size in editboxes "Size, X" and "Size, Y" (12 and 12 for example) and choose the architecture from "Map's Architecture Group".
- Enter the name ("Example Map 1" for example) of new map in «Map's Name» editbox. Press «Enter». Then Map Window will switch from Creating New Map Mode to Working with Map Mode.
- If you want to change the name of some columns just choose one of them in column's list, enter the new name in editbox above and press "Enter" (for example choose the first column's name ("X_Coordinate") and change it to "Coordinate_X").
- Press the «Initialize Map» button. Initialize Map Window will be opened (see chapter 3.3 in program description).
- Press «…» button and choose in appeared dialog file with GEO-Eas data ("init.dat" for example). After closing the dialog select «Yes» radiobutton on «Missing Data?» panel if this file contains missing data and enter the value by what they are marked in this file (There isn't missing data in example data files and you should just press "Initialize"). Then press «Initialize» button. If there isn't missing data just press «Initialize» button. Map will be initialized, this map will be closed and Map Window will became active. Then you may save the initialized map and see that it has been initialized by random vectors from "init.dat" file. Below you can see the "Coordinate_X", "Y_Coordinate"and two other components ("Data 20" and "Data 50") distribution on the initialized map and convince yourself that it is random.

Fig. 1



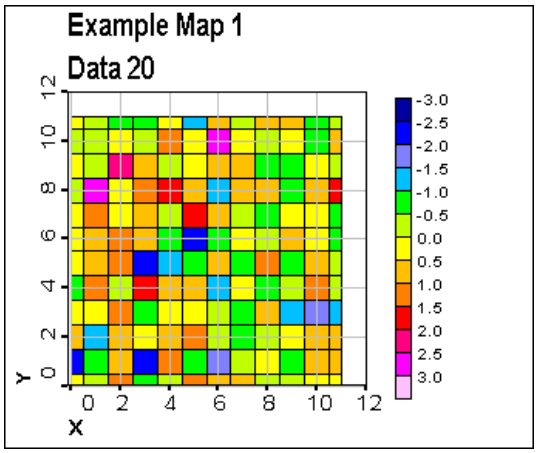


Fig. 11. Distribution of the "Data_20" and "Data_50" initializing data components on initialized map

- Press the «Train Map» button. Train Map Window will be opened (see chapter 3.4 in program description).
- Press «…» button and choose in appeared dialog file with GEO-Eas data ("train.dat" for example). After closing the dialog column's list will be filled by columns' names from this file. Then select «Yes» radiobutton on «Missing Data?» panel if this fail contains missing data and enter the value that they marked in this file ("train.dat" doesn't contain any missing data than don't change this parameter). Note than learning rate Number of iterations and neighborhood radius are already set but you can always change them. For example you can select "Two-Step Training" checkbox and enter the other parameters from Fig.5 in program description. .Then press «Train» button. This map will be closed and Expectation Window will be opened (see chapter 3.6 in program description).
- Below you can see the "Coordinate_X", "Y_Coordinate" and two other components distribution on the trained map and convince yourself that map's nodes was ordered during the training process:
- Press the «Visualize Data» button. Analize Data Window will be opened (see chapter 3.6 in program description).
- Press the upper «…» button and choose in appeared dialog file with GEO-Eas data ("vis.dat" for example). Then press the lower «…» button and choose in appeared dialog file to save the results of data analysis ("vis_out.dat" for example). Then, if input file contains missing data check «Is There Missing Data»? check box and set the value for their marking (if you choose "vis.dat" check this checkbox and set –11111 value for missing data marking). Set the «Add Reference Vectors?» checkbox. Press «Visualize» button. Data will be analyzed, results of it will be saved in output file, this map will be closed and Map Window will become active.

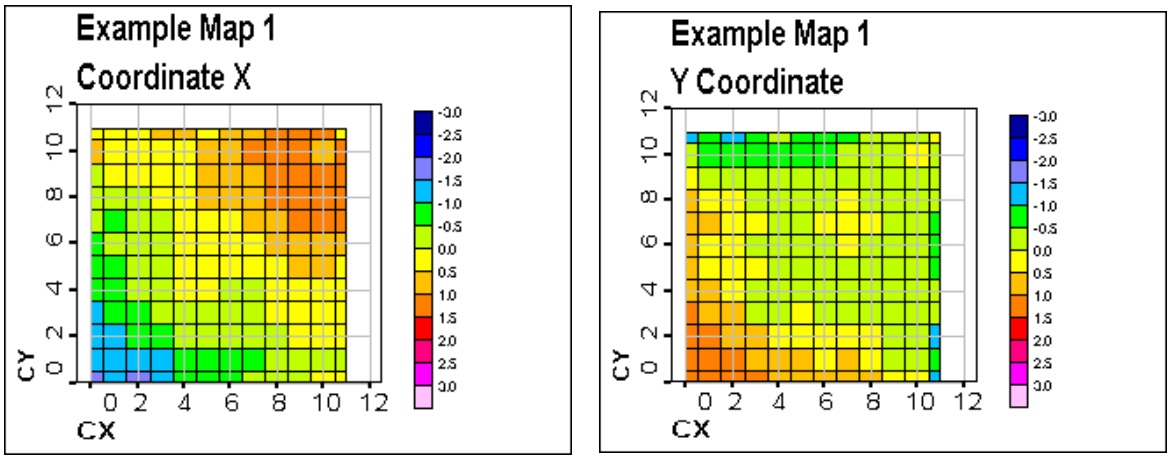


Fig. 12. Distribution of the "Coordinate_X" and "Y_Coordinate" data components on trained map

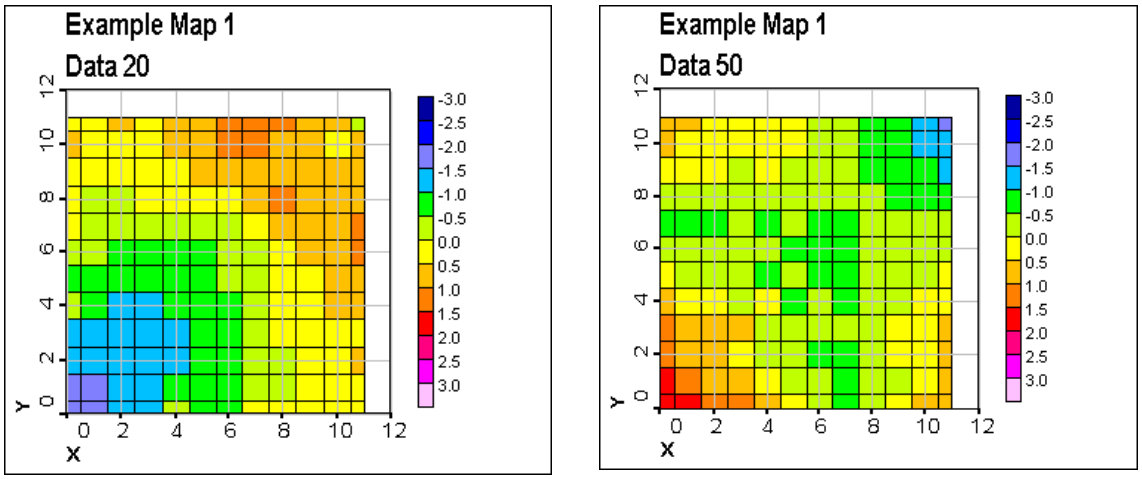


Fig.13. Distribution of the "Data_20" and "Data_50" data components on trained map

- Below you can see the nodes which has been associated with analyzing data:

All illustration in this appendix was made by means of GeoStat Office program Package.

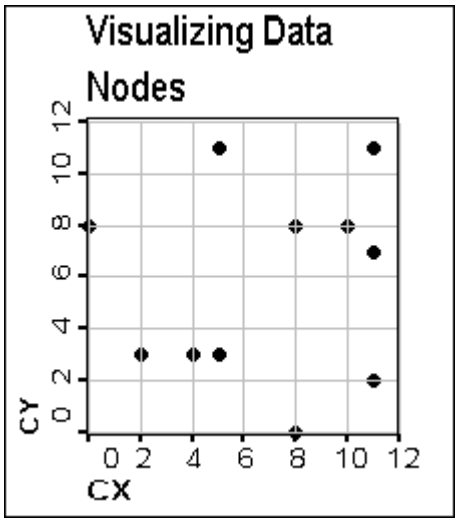


Fig. 14. Map's nodes which has been associated with visualizing data