

ChoroWare: A Software Toolkit for Choropleth Map Classifications

Ningchuan Xiao, Marc P. Armstrong and David A. Bennett

Department of Geography

The University of Iowa

Iowa City, IA 52242

E-mail: {ningchuan-xiao; marc-armstrong; david-bennett}@uiowa.edu

When exploratory spatial data analyses are conducted, choropleth mapping is an important means to display 1) the patterns of the spatial observations for enumeration areas, such as counties or Census Tracts, and 2) the resulting spatial statistical outcomes. Choropleth maps group spatial observations into classes and each class is then assigned a particular color that is used to shade the enumeration areas. The classifications can be performed in a large number of ways, with each method focusing on a specific criterion, such as within-class similarity (Jenks and Caspall 1971), geographical structure (Monmonier 1972; Murray and Shyy 2000), and statistical characteristics (e.g., quantiles) (Slocum 1999: 67).

It is unlikely, however, that a particular classification will satisfy all criteria. In practice, a classification that is optimal for one criterion may turn out to be poor from other perspectives. Therefore, an appropriate design goal for a choropleth map is to consider classification as a multicriteria (or multiobjective) problem. That is, instead of looking for a single classification that is best for all criteria, it is more useful for cartographers to examine the classifications along a Pareto-like front where no classification can be considered to be better than, or dominate, others.

The purpose of this paper is to describe the design and implementation of a software toolkit, called ChoroWare, that can be used to help cartographers find a set of class intervals that is suitable for a specific application. Using ChoroWare, users can explore a variety of alternative classifications and select one that they deem most suitable. To achieve this goal, two critical tasks must be carried out. First, ChoroWare must be able to generate the set of nondominated alternatives that forms the Pareto front. Then, a visualization tool must be designed to allow users to interactively display the trade-offs between criteria and the resulting choropleth map for each non-dominated alternative. The overall architecture of ChoroWare is illustrated in Figure 1. The functionality of each module is discussed below.

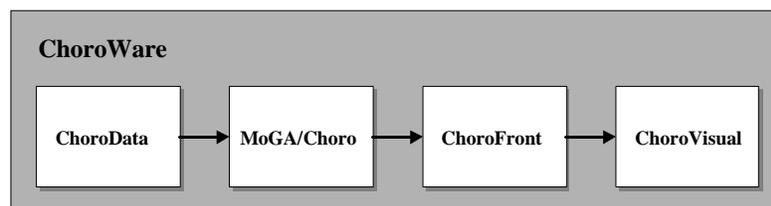


Figure 1. Overall structure of ChoroWare.

Generating nondominated alternatives is difficult for many multicriteria problems (Cohen 1978). Recent developments, however, have shown that genetic algorithms (GAs) are able to generate alternatives for multicriteria problems (Zitzler et al. 2001; Xiao et al. 2002). In a GA, a population of individual solutions (i.e., a classification for a choropleth map) is randomly initialized and then manipulated by a set of iterative operations, including selection, recombination, and mutation. At the end of each iteration, solutions are evaluated according to a set of objectives. To obtain a diverse population of alternatives, which is critical for multicriteria problems, we extended the original GA island model (see Cantú-Paz and Goldberg 2000) and designed a specialized island model. In our approach, the entire population is divided into several sub-populations (“islands”) and each subpopulation is processed using a local set of evolutionary operations. A mechanism, called migration, is used to exchange individuals, at a certain rate and interval, among the islands. Inside each sub-population, an individual (i.e., a specific classification scheme) is evaluated on a (partial) set of objectives.

Based on this principle, we designed a module called MoGA/Choro (multiobjective GA for choroplethic classification) using object-oriented techniques (Figure 2). In this context a *population* contains several *subpopulations*, and each *subpopulation* has many individuals represented using class *genotype*. A *genotype* has a full set of genetic operators and a member datum called *chromosome*, which represents a classification scheme by encoding the break points of the classes into a string of integers. A *genotype* also contains an aggregating object called *models*, which contains the models used to calculate the objective values. We have developed four “built-in” objectives: goodness of variance fit (Robinson et al. 1984: 363), equal area of classes, spatial autocorrelation, and boundary accuracy index (Jenks and Caspall 1971); using this pattern of objects (Figure 2), new objectives can be easily defined and incorporated into the source code of the module. Users can specify which objectives are desired in the configuration file of MoGA/Choro. The object *models* contains an aggregating object called *parameters*, which includes the data needed in the calculations. These data are generated from the raw spatial data (in vector or raster formats) by a module called ChoroData. Data generated by ChoroData include an array of all observations (for vector), a two-dimensional array of observations (for raster), an array of non-repeated observations (for both vector and raster), and an array of linked lists of neighbors for each polygon in a vector data structure.

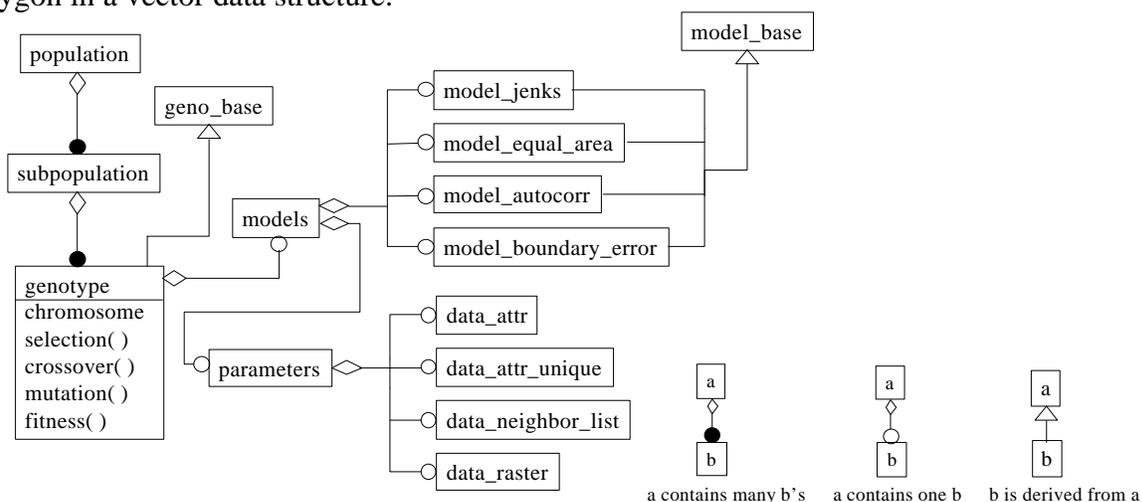


Figure 2. The design pattern of MoGA/Choro.

Within MoGA/Choro, classifications are saved to a file. The nondominated alternatives are then extracted by a module called ChoroFront, and the results are visualized using a module called ChoroVisual. The ChoroVisual tool was developed using GTK+ and GDK (www.gtk.org). It currently consists of six windows to display 1) a map, 2) the attributes of polygons, 3) a value path (also known as parallel coordinate plot), 4) a legend editor (called a classifier, for manual classification), 5) a list of all nondominated alternatives (the window at the upper-left corner marked as “Front List”), and 6) a multivariate plot (Figure 3). In this last window, a user can choose to display a plot formed by any two variables listed in the left panel. Among these variables, the last four are the built-in objectives. Each classification alternative is displayed as a small white square in the solution space drawing area, where a red square is used to indicate the current classification being used to draw the map. The current classification is also highlighted in the front list and in the value path (the red line). Linkages among the map, multivariate plot, and front list enable a user to examine the alternatives on the fly. The purpose of the legend editor is to help users further explore each selected alternative. In the classifier window, a user can examine the histogram of the observations and the break point of the classification, adjust the classification, and change the color of classes. In so doing, a user can start from an acceptable classification and then, based on this classification, try to find a better one.

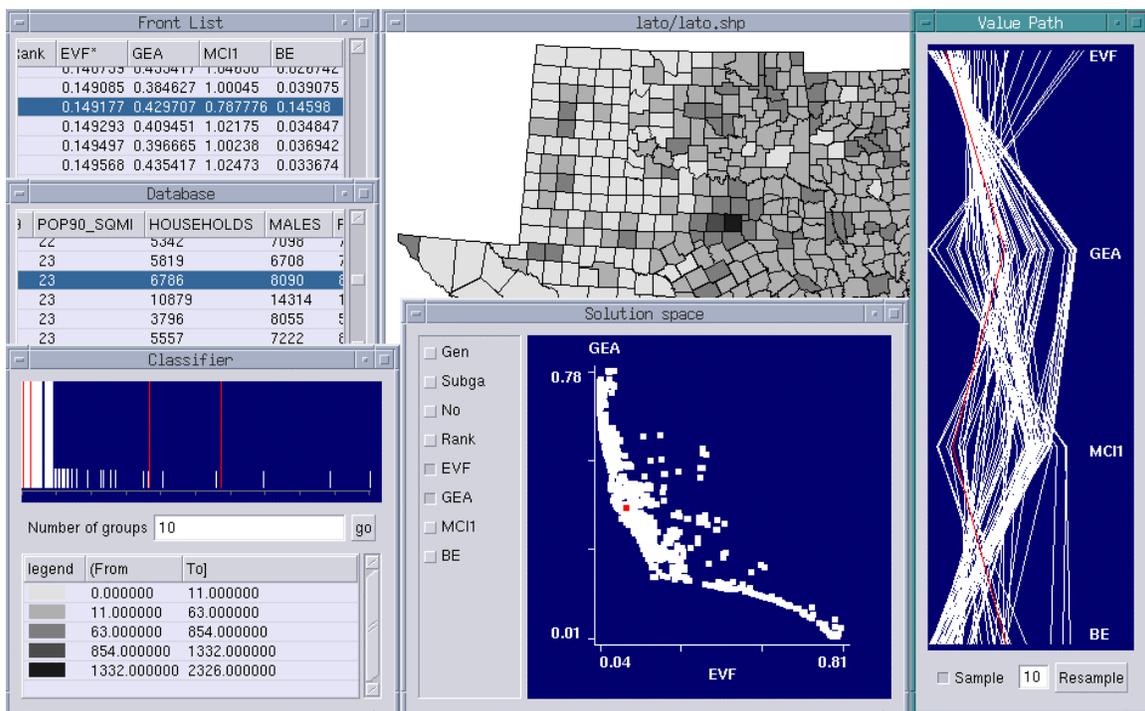


Figure 3. A screenshot of ChoroVisual.

In general, using ChoroWare, cartographers and spatial analysts can have more flexibility in choosing a suitable classification scheme for choropleth maps than current GIS and cartographic software can provide. It is developed using Open Source technology based on Linux. Future developments will focus on integrating ChoroWare with other powerful visualization tools such as GGobi (www.ggobi.org), and developing more modules to help users distinguish alternative classifications.

References

- Cantú-Paz, E., and D. E. Goldberg. 2000. Efficient parallel genetic algorithms: theory and practice. *Computer Methods in Applied Mechanics and Engineering* 186:221-238.
- Cohon, J. L. 1978. *Multiobjective Programming and Planning*. New York: Academic Press.
- Jenks, G. F., and F. C. Caspall. 1971. Error on choroplethic maps: definition, measurement, reduction. *Annals of the Association of American Geographers* 61 (2):217-244.
- Monmonier, M. S. 1972. Contiguity-based class-interval selection: a method for simplifying patterns on statistical maps. *The Geographical Review* 62 (2):203-228.
- Murray, A. T., and T.-K. Shyy. 2000. Integrating attribute and space characteristics in choropleth display and spatial data mining. *International Journal of Geographical Information Science* 14 (7):649-667.
- Robinson, A.H., Sale, R.D., Morrison, J.L., and Muehrcke, P.C. 1984. *Elements of Cartography*. 5th ed. New York, NY: John Wiley & Sons.
- Slocum, T. A. 1999. *Thematic Cartography and Visualization*. Upper Saddle River, NJ: Prentice Hall.
- Xiao, N., D. A. Bennett, and M. P. Armstrong. 2002. Using evolutionary algorithms to generate alternatives for multiobjective site search problems. *Environment and Planning A* in press.
- Zitzler, E., K. Deb, L. Thiele, C. A. C. Coello, and D. Corne, eds. 2001. *Evolutionary Multi-Criterion Optimization: Proceedings of the First International Conference*. Berlin: Springer.