

Formalizing Guidelines for Building Meaningful Self-Organizing Maps

Jochen Wendel¹, Barbara. P. Battenfield¹

¹Department of Geography, University of Colorado - Boulder
 Email: jochen.wendel@colorado.edu, babs@colorado.edu

1. Introduction

Many parameters must be accounted for in creating a Self-Organizing Map (SOM). These parameters can dramatically change the content and organization of a SOM, but it is often hard to find instructions. Despite a large body of published literature (Oja et al., 2003 and Kaski et al., 1998) we have not yet discovered any single book or article which outlines a complete set of guidelines on how to build a SOM effectively. Existing literature does reference individual metrics for specific parameters and we draw these together in this paper. This research establishes guidelines for the creation of meaningful SOMs, drawn from empiric testing as well as 'best practice' conventions. We create SOMs using different initialization parameters, training levels, network shape and node size, comparing outputs, uncertainty measures and interpretations, using Kohonen's MatLab toolbox.

2. Pilot Study Data

The pilot study dataset contains binary information listing 108 GIS hydrologic operators categorized on ten distinct dimensions (Table 1).

Table 1. Dimensions characterizing GIS commands, attributes in italics refer to degrees of freedom (discussed in section 2).

Raster Only	1 = task is raster data only	
Raster and Vector	1 = operates on both	<i>Vector Only</i>
Data Management	1 = task is a data mgt. function (copy, delete, etc.)	
Simple	1 = atomic command	<i>Compound</i>
Geometric	1 = task modifies geometry	<i>Attribute</i>
Terrain	1 = task deals with terrain	
Flow	1 = task deals with flow	<i>Terrain and Flow</i>
Regional	1 = task works on neighborhood	
Local	1 = task works on each individual pixel	<i>Global</i>
CSR	1 = task changes spatial relation	

The original contribution of this pilot study was the generation of implicit keywords describing software commands. Most SOMs are built for full text documents from which explicit keywords can be obtained. One challenge specific to software tools and commands is to establish semantics for implicit keywords. Manuscripts such as

articles, news stories or full-text documents can be distinguished by keywords which are explicitly incorporated; but this is not the case with software libraries. While environmental models are typically distributed with documentation, formalized description methods for software commands are generally not well-developed. Other challenges include interpreting the dimensionality of the resulting catalog, and determining the optimal number of keywords (dimensions) which will best distinguish among the catalog entries. Generation of implicit keyword sets is broadly applicable to many problem domains in addition to software libraries, such as intercorrelated census data sets.

For the compilation of this dataset, source materials and online help files from commercially available GIS and statistical analysis products were used to refine the list and eliminate redundancy. Data handling aspects were also captured, such as whether the operators modify spatial relationships, whether they operate on geometry or attributes, what data model is required for input and output, etc. To eliminate dimensional redundancy, degrees of freedom were removed; e.g. binary coding of only two of the three options of global, regional and local operators is required, since knowledge about the first two eliminates the need to code the third. A Boolean matrix of the ten dimensions for the 100 operators was processed by Kohonen's SOM method (Kohonen, 2001) using existing toolsets in MatLab. We will discuss implicit keyword formation in more detail at the conference.

3. Guidelines for building SOMs

We augmented guidelines from existing literature with empiric testing. The established guidelines were grouped into six categories, which form a rough sequence of steps through SOM creation:

1. Initialization
2. SOM size
3. SOM shape
4. Neighborhood size and geometry
5. Training length and matrix-tuning
6. Quantification of uncertainty

3.1 Initialization

Tests were conducted using random and linear initialization. Normalization was not necessary for the binary dataset. Skupin (2008) recommends using a random initialization process as it preserves true self-organization.

3.2 SOM Size

Initial qualitative recommendations for selecting SOM size range from identifying the goal of the SOM to building small, medium and large SOMs depending on the purpose of the data exploration (Ultsch and Simon, 1990). Vesanto (2005) offers a specific quantitative recommendation to compute optimal SOM size (Vesanto, 2005):

$$msize = 5 \times \sqrt{k} \quad (1)$$

where n is the product of observations (rows) and variables (columns) of the dataset. Optimal size will minimize the chance of creating a SOM with too many empty cells (but note that some empty cells are needed to facilitate cluster interpretation). Initial tests included SOMs of sizes ranging from 64 to 1024 cells (Figure 1). The *msize* for

our dataset was computed to be 169 for an input matrix of 108 observations and 10 variables. Figure 1 shows that in this optimal solution clusters are well developed and there is plenty of space between clusters to permit easy distinctions.

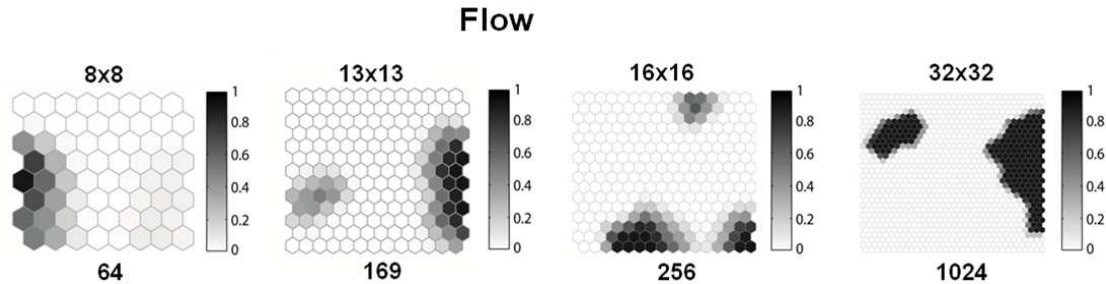


Figure 1. SOM visualization of the "Flow" dimension across different SOM sizes

3.3 SOM Shape

Quantitative recommendations by Kohonen (1990) suggest an asymmetrical rather than a symmetrical SOM shape to avoid edge effects. His advice is that the short side length should be at least half of the longer side of the SOM. We have come to think of this suggestion as the "One-Half Rule". In our tests, a symmetrical shape situated clusters more often than an asymmetrical shape towards the edge of the SOM. Interior cluster locations are easier to interpret, since their distance to other clusters on all sides can be judged. Building on the One-Half Rule, the optimal SOM shape solution was a SOM of 16x11 (176) cells. Notice this is slightly more cells than the *m*size computation advises, but the difference is small.

3.4 Neighborhood Size and Geometry

Neighborhood characteristics affect cluster formation by constraining the number of cell values which are adjusted following each cell assignment. We conducted tests by systematically adjusting neighborhood size and geometry. Best results were archived when using a Gaussian kernel. To establish best size, in the first training step we started with a larger neighborhood size spanning the entire SOM (Skupin, 2008) and then progressively reduced the neighborhood to half as the iterations of the learning algorithm progressed. If the kernel size reaches 0 the SOM algorithm equates with a K-means method (Kohonen, 1990).

3.5 Training length and matrix tuning

Training length and matrix tuning tests ranging from 1 to 100,000 iterations were conducted. As shown in Figure 2, after 1,000 iterations the SOM output begins to stabilize and fine-adjustments can be made. For this data set, we found good results using a training length of 10,000 and a matrix tuning rate of 2,000.

Flow

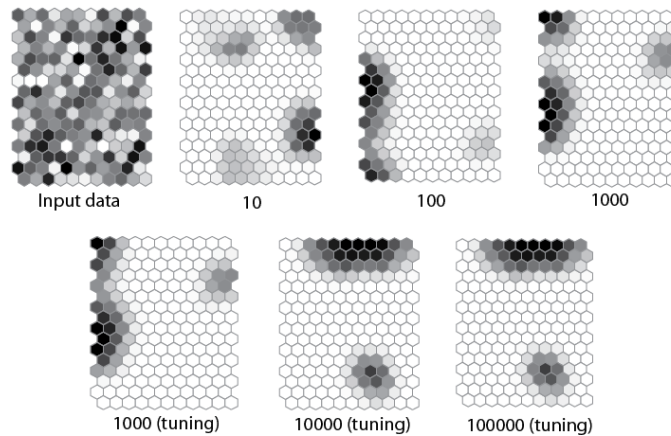


Figure 2. The "Flow" SOM visualization across different training iterations for a single variable (the "Flow" dimension).

3.6 Assessment of Uncertainty

Kohonen commonly uses two metrics to assess the uncertainty of a SOM, the Quantization Error and the Unified Distance Matrix (or U-Matrix). The Quantization error is defined as:

$$E\{\|x - m_c(x)\|\} \quad (2)$$

where c indicates the Best-Matching Unit (BMU) for the input vector x (Honkela, 1999). The Quantization error returns a value between 1.0 and 0.0 which indicates how well the SOM fits the input dataset. The goal is to establish a minimum value, although values very close to 0.0 indicate model over-fitting (Kohonen, 2001). The Quantization error allows comparable quantitative assessments among different SOMs.

Figure 3 shows that Quantization Error drops substantially over initial iterations.

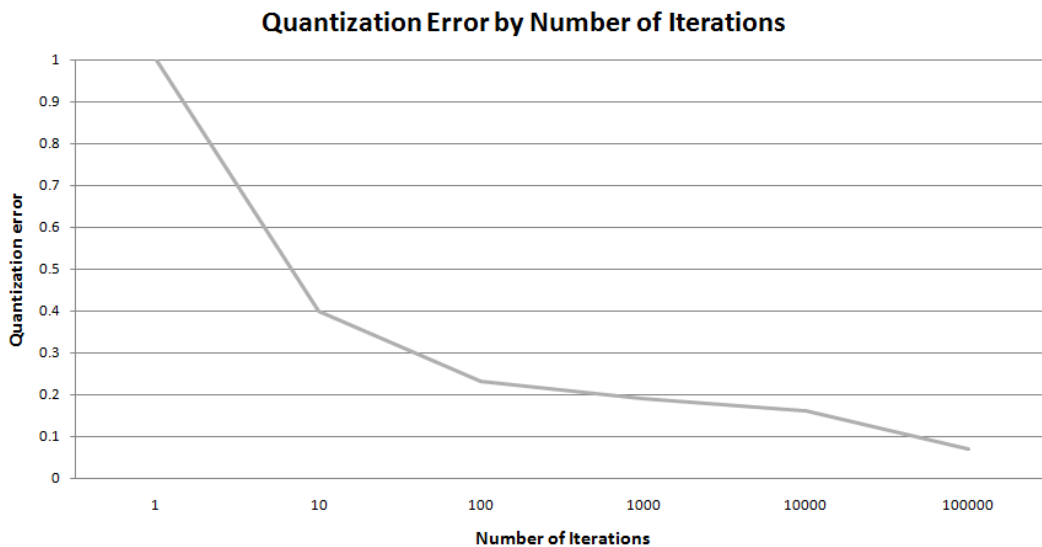


Figure 3. Error measurements by number of iterations for the binary case study dataset on the optimal SOM

Table 2 shows the Quantization Error across different SOM shapes. We also conducted initial tests exploring 3D SOMs, exploring cylindrical and toroidal solids. Further discussion about using 3D SOMs will be presented at the conference.

Table 2. Error comparison of different SOM shapes using 10,000 training iterations and 2,000 fine tuning iterations.

SOM size	Quantization Error
8x8	0.4586
16x16	0.1084
32x32	0.0001
24x8	0.3160
16x11	0.1990
32x6	0.1780
16x11*	0.2208
16x11**	0.2850

* *cylindrical 3D SOM* ** *toroidal 3D SOM*

The Unified Distance Matrix, or U-Matrix indicates quality and clarity of the clusters. The U-Matrix contains twice as many cells as the SOM and is defined by similarity measurements between each cell and neighboring cells. Initial tests on the U-Matrix using different SOM shapes are shown in Figure 4. The 32x32 U-Matrix shows over-fitting because the cluster borders are too pronounced, whereas the 8x8 does not show distinct enough cluster borders. The optimal solution following the recommendation is the 16x11 U-Matrix as U-Matrix.

Kohonen (2001) advises that these two measures (Quantification Error and interpretation of the U-matrix) should not be trusted alone without considering other factors because, both measures can indicate best results even when the SOM has over-fitted the data. Other factors to be considered include number of variables, number of observations, compactness and positioning of the SOM clusters.

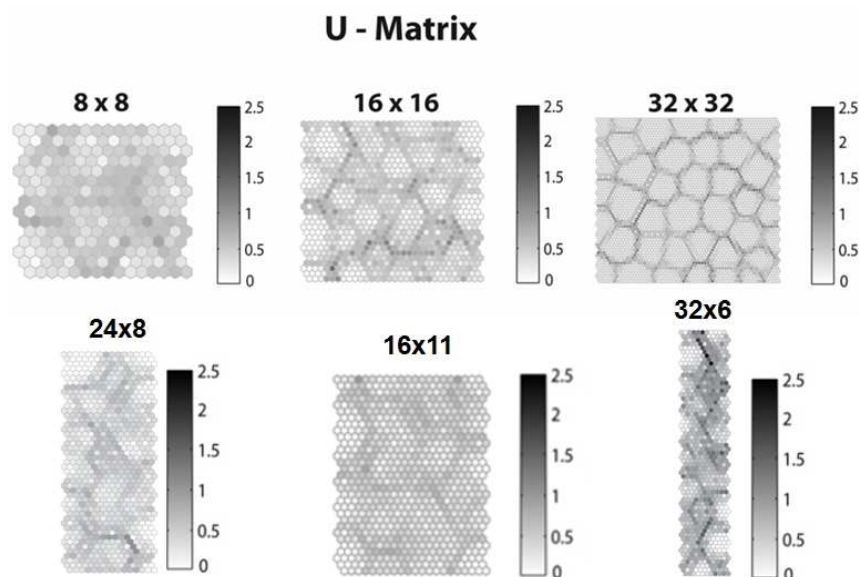


Figure 4 U-Matrix across different SOM shapes and sizes.

4. Discussion

The presentation will cover all six steps for building a SOM. We will offer guidelines for building SOMs for a catalog problem set which includes full text documents and software routines, demonstrating different impacts of size, shape, training iterations on the outcome assessment matrix.

Acknowledgements

This research is supported by USGS grant # 04121HS029 "Generalization and Data Modeling for New Generation Topographic Mapping". We thank Roland Viger and Jeremy Smith for helpful input and critique.

References

- Honkela, T, 1999 Information on Self-Organizing Maps (SOM), <http://www.mlab.uiah.fi/~timo/som/>, Last visited 06/29/2010.
- Skupin, A, 2008 Introduction: What is a Self-Organizing Map. In: Agarwal, P and Skupin A. (Eds.). *Self-Organizing Maps: Applications in Geographic Information Science*. John Wiley & Sons. Ltd, Chichester, UK: 19-44.
- Kaski, S et al., 1998 Bibliography of self-organizing map (SOM) papers: 1981-1997. *Neural Computing Surveys*, 1(3&4): 1-176.
- Kohonen, T, 1990 The self-organizing map. *Proceedings IEEE*, 78(9): 1464-1480.
- Kohonen, T, 2001 Self-organizing maps. Springer, Berlin; New York.
- Oja, M et al., 2003 Bibliography of self-organizing map (SOM) papers: 1998-2001 addendum. *Neural Computing Surveys*, 3(1): 1-156.
- Ultsch, A and Siemon, HP, 1990 Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis, *Neural Networks*. Kluwer Academic Press: 305-308.
- Vesanto, J, 2005 SOM implementation in SOM Toolbox. SOM Toolbox Online Help, <http://www.cis.hut.fi/projects/somtoolbox/documentation/somalg.shtml>