Recognisable Shapes for Self-Organizing Maps

Rudolf Mayer
rudolf_mayer@utanet.at

Institute of Software Technology and Interactive Systems
Vienna University of Technology, Austria

Abstract. The Self-Organizing Map (SOM), and other related architectures, enjoy a growing popularity in the field of Data Mining. These neural network algorithms provide a topology-preserving mapping from high-dimensional data to a lower dimension, which allows for an easier interpretation of complex data. For visualisation of trained maps, a lot of different techniques have been developed. However, convenient and practical methods for *describing* the (normally) rectangular maps have not yet been subject of intensive research, and are still missing. In this paper, new shapes for self organising architectures, which allows for an easier explanation, will be presented. These map layouts are oriented on shapes well-known to readers, as for example country or continent maps, or geometrical shapes.

1 Introduction

The Self-Organizing Map (SOM), as first introduced in [Koh82], and related self-organising architectures, enjoy a growing popularity for Data Mining applications. This is due to their algorithms generating a topology-preserving mapping from a high-dimensional input space to a lower dimensional output space; the thus structured and organised data allows for an easier interpretation of complex inherent structures and correlations in the data.
In many applications, the mapping is towards a two-dimensional, rectangular grid, an easy human read- and interpretable form of representation. There exists a lot of techniques for visualising the thus generated maps, and with additional techniques, like labelling of the nodes, or visualisations of the detected clusters, the user can be further aided.

However, there are still shortcomings when it comes to *explaining* a Self-Organizing Map: when referring to a special region of the map, one often has to refer to some "corners" of the map; for bigger maps, only a part of the map can be referred to like this, and therefore often one has to refer to the respective node(s) by assigning them x/y coordinates in the grid. This method seems to be inappropriate in many cases, as the reader first has to search the specific area on the map. A significant improvement might be when one can refer to a map of which the shape is known by the reader. Examples for this could be maps in the shape of a country or continent, a human body, geometric shapes like a star, or any other, well-known shape. Then, one could refer to provinces, districts or

cities in the first, and to parts of the human body in the second example, and the reader immediately would know what region of the map is talked about.
In this paper, experiments with implementing this approach, using a map layout in the shape of *Austria*, are presented and discussed.

The remainder of this paper is organised as follows: first, we will give a brief overview of the used models by introducing and describing the *Self-Organizing Map* in Section 2, as well as the *Incremental Grid Growing* in Section 3. After describing the data used in our experiments in Section 4.1, we will present and discuss the results of the experiments in Section 4.2. Section 5 will give our conclusions on the proposed approach, as well as a brief outlook on future work.

2 The Self-Organizing Map

The Self-Organizing Map [Koh82] is a well known and widely used neural network model that based on unsupervised learning. The SOM training algorithm generates a mapping from a high-dimensional input space to a lower-dimensional output space. In many cases, the output space is two-dimensional. Though also hexagonal structures are used, in many applications this output space is of the form of a rectangular grid of nodes.

Each node on the grid is associated with a model m_i of some observation, in the form of a so-called *model vector* $m_i = [m_{i1}, m_{i2}, ...m_{in}]^T \in \Re^n$ of the same dimension as the input vectors $x_i = [x_{i1}, x_{i2}, ...x_{in}]^T \in \Re^n$. The models are computed by the SOM algorithm so that they optimally describe the domain of observations, and organised in a way that similar models are closer to each other than more dissimilar ones. To each of the nodes of the SOM, a number of input patterns, in the form of the input vectors x_i, are assigned to during the training process, with similar vectors being mapped to the same node. By that, the SOM provides a kind of clustering of the input data, a desired feature in Information Retrieval and Text Mining applications. An illustration of the mapping generated by the SOM-algorithm is given in Figure 1. Note that input patterns that are spatially close to each other in the input space will be spatially close to each other in the mapping, as well as that the high-density area in V is represented by correspondingly many spatially close elements in A.

The SOM training algorithm will now be described in brief. After initialisation, where a grid of the predefined size is generated, and random values are assigned to the model vectors of the nodes, the SOM training algorithm consists of a number of iterations of two basic steps:

- presenting input patterns and finding the best matching node
- adapting the model vectors of the best matching node and a certain number of neighbouring nodes towards the presented input pattern.

An input pattern x is randomly chosen from the input space and presented to the network. The best matching node, sometimes called the *winner*, is computed as the node of which the model vector is the most similar to the presented input vector x. As a measure for the similarity, the Euclidian distance between the

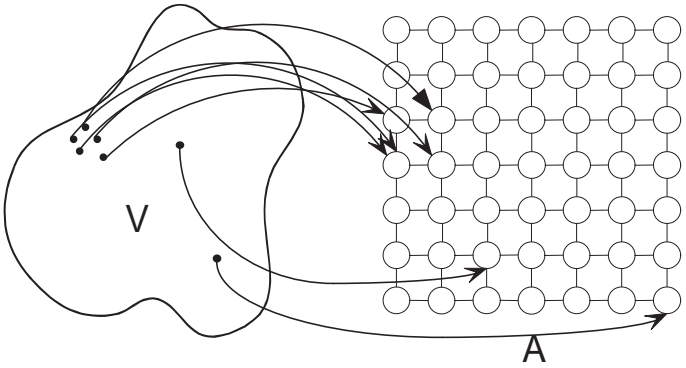

Fig. 1. Topology-preserving mapping from a high-dimensional input-space V to a two-dimensional output-space A.

model vector m_i and the input vector x is widely used. The best matching node c therefore is defined according to

$$c(x,t) = arg\min_i\{\|x(t) - m_i(t)\|\}. \tag{1}$$

To improve the quality of the mapping generated by the SOM, after each vector presentation, some of the model vectors of the SOM are adapted towards the input vector x in the *training process*. The value for each of the new model vectors is determined by the vector's current value and two other factors, the *learning rate* α, as well as the *neighbourhood function* h_{ci}, and can be computed according to

$$m_i(t+1) = m_i(t) + \alpha(t) \cdot h_{ci}(t)[x(t) - m_i(t)]. \tag{2}$$

The learning rate α, $0 < \alpha(t) < 1$, determines how much a vector is adapted. It is a time-decreasing function, i.e. vectors are adapted more in the beginning of the learning process, with this adaptation decreasing towards the end.

The neighbourhood function is typically designed to be symmetric around the winning node. Its task is to impose a spatial structure on the amount of model vector adaptation. The neighbourhood function determines which nodes (and how much) are going to be adapted: only nodes that are within a certain range around the winner are going to be adapted, where nodes that are farther away are adapted less than nodes close to the winner. A gaussian function, centred on the winner, is widely used for determining the adaptation strength, defined as follows:

$$h_{ci}(t) = e^{\frac{\|r_i - r_c\|^2}{2 \cdot \sigma(t)}}, \tag{3}$$

where r_i and r_c denote the coordinates of the node i and the winning node c in the two-dimensional output space $\Re^2$, respectively.

In general, the SOM consists of a rectangular grid which is fully occupied, i.e., every position on the grid represents a node. However, for representing the irregular shapes we are proposing and using in this paper, for example countries, we need an algorithm that can as well work on only sparsely filled grids.

Additionally, the SOM also assumes the grid to be *fully connected*, or, in other words, the SOM does not use the concept of *connections* between nodes; therefore, adaptation of nodes will happen on **all** the neighbouring nodes. With this algorithm, it is not possible to correctly represent concave shapes - an illustration of this problem is given in Figure 2, where the black-coloured node is the winner, and the grey nodes are going to be adapted. Note, however, that the grey-shaded node to the right of the winner is also going to be adapted, while it should actually **not** be adapted, as there should be no connection between the nodes.

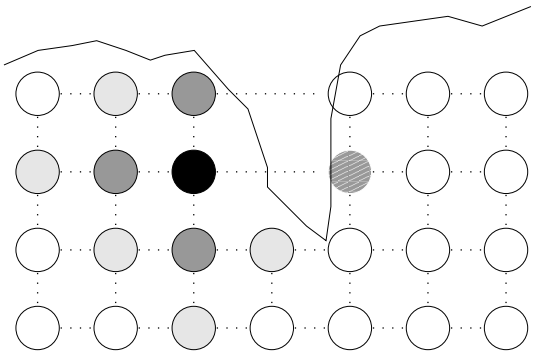

Fig. 2. Modelling concave shapes in a Self-Organizing Map.

For these reasons mentioned above, we need to adapt the SOM algorithm to work according to our needs; however, a model that fulfills our requirements already exists - the Incremental Grid Growing, as presented in the following Section 3.

3 The Incremental Grid Growing

The Incremental Grid Growing [BM93] (IGG) is a neural network model based on the principles of the Self-Organizing Map. The Incremental Grid Growing was originally developed to meet one of the shortcomings of the SOM, namely the need to a-priori define the size of the grid representing the output space; this, in general, requires some knowledge of the input data. This problem is addressed in the IGG by incrementally growing the grid at its perimeter throughout the

training algorithm, to reach a final, not necessarily fully occupied, grid of a size optimally matching the input data.
However, we can as well utilise the IGG algorithm for a static, predefined grid size and shape.

Additionally, the Incremental Grid Growing algorithm presents another interesting concept we require for our purpose: the concept of *connections* between nodes. In the IGG model, nodes can either have a connection to their neighbouring nodes, or not. This concept has impacts to the adaptation of nodes during the learning process - only neighbouring nodes that have a connection with the winner are going to be adapted. Also, the strength of the adaptation is related to the length of the minimal path between the winner and a specific node, rather than their distance in the grid.

The importance of this property becomes apparent when reconsidering Figure 2, pointing out problems with the representation of concave shapes in a standard SOM. The same map layout as above is given in Figure 3. Using a model that supports connectivity, we can also correctly represent concave shapes - the node to the right of the winner is now **not** going to be adapted, as there is no connection between the nodes that lies within the neighbourhood range (in this example, the neighbourhood range has the value of 2).

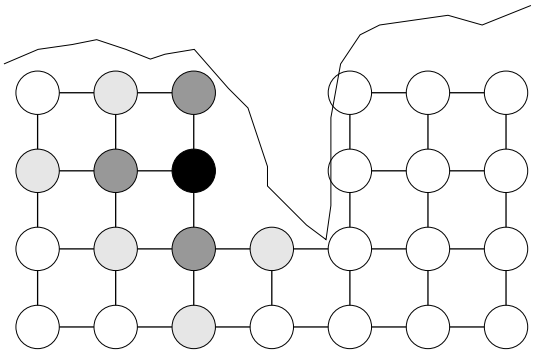

Fig. 3. Modelling concave shapes in an Incremental Grid Growing.

Originally, the concept of connections was introduced to automatically provide a visualisation of cluster boundaries - clusters become apparent as interconnected nodes, which are separated from other clusters of nodes by missing connections to these clusters. Connections are added and removed between adjacent nodes during the training process, depending on the similarity or dissimilarity between their model vectors.
We can, however, as well utilise this algorithm for static connection states.

4 Experiments

The goal of our experiments is to generate a mapping from the high dimensional input data to a two-dimensional output space using a recognisable shape, resembling a map of Austria. In this section, we will first describe the data used for our experiments, and then describe and explain the results of our experiments, taking advantage of the special shape we used for generating the mapping.

4.1 The Data

The data used in our experiments stems from abstracts of papers submitted for the European Congress of Radiology 2003[1]. The subset of the collection we will use in our experiments consists of 1009 different documents submitted for the "Scientific Sessions" at this congress; the language of all the documents is English. Our document collection is in plain-text, therefore no formatting information had to be removed. No stemming, or any other term reduction, was applied.

In order to use any data set as input data set for analysis in the Incremental Grid Growing, or any other related model, it has to be described as a set of *feature vectors*, containing a certain number of components. Therefore, we have to use a technique to describe our document collection in a *vector-space representation*: for each document, we obtain a vector that describes the very same document. As vector components, we extract the list of all words present in the document collection. However, this list would be rather long, and would not necessarily be adequate to unequivocally represent the documents. Therefore, we remove terms that do not significantly contribute to a description of the content of our documents. One approach would be to use so-called *stop word* lists, i.e. lists that contain, among others, grammatical function words such as conjugations, articles, or pronouns, etc. These terms exhibit approximately equal frequency in all the documents of a collection, and therefore do not discriminate between the single documents. Furthermore, terms that are specific for the data set should be discarded. For example, in a document collection of hotel descriptions, the term "hotel" will be present in a very high percentage of the documents, and will therefore not contribute to describing the documents.

However, such lists have to be created manually matching the given language and domain of the documents, and we therefore prefer to automatically discard terms that are not important for distinguishing between documents. For our experiments, we discarded terms that consisted of less than three letters. Moreover, we could discard terms that appear in *more* than a certain percentage of the documents, to discard other stop-words or domain specific terms. As the amount of terms we will loose by this approach is rather small, we, though, did not apply this method.
However, words that occur only in a number of documents below a certain percentage threshold, should be discarded, as these words do not contribute much

[1] http://www.ecr.org

to discriminating the documents; in our experiment, we only accepted terms that appeared in more than 1 % of the documents.

Using these techniques, we reduced the initially 4936 unique words with a length of three or more letters to a final number of 1244 different terms.

For generating our input vectors from our thus created word list, a number of approaches exist. For our experiments, we utilised a $tf \times idf$ weighting scheme [Sal89] to generate weighted values for the components in our input vectors. This weighting scheme asses that words are of importance when they occur more frequently in one document, and less frequently in the rest of the document collection. We therefore extract the term frequency tf_{ij} of a term T_j in document D_i, i.e., how many times the term appears in this document. Additionally, the document frequency df_i, defined as the number of documents in a collection of N documents in which the term T_j occurs, is calculated. An appropriate indication of the term value as a document discriminator can be given by using an *inverse document frequency idf*, like $\log \frac{N}{df_i}$. Then, we can define a measure of the importance of a term for the feature vector x by computing a weighted value for its components k according to

$$x_k = tf_{ij} \cdot \log \frac{N}{df_i}. \tag{4}$$

Our data set contained additional information: regarding their topic, the documents have already been (manually) categorised into 15 different classes, as for example "Radiographers", or "Computer Applications". A complete list of categories can be found in Table 1. Having predefined classes available, we can compare the clustering results from our neural network with the existing one, as can be seen in the following Section.

4.2 Results

For our experiments, we generated two maps that resemble the shape of Austria; the maps contained 296 nodes in a 33 × 17 map, and 552 nodes in a 45 × 23 map, respectively. Illustrations of the shapes of these initial maps are given in Figures 4 and 5, respectively.

For our experiments, we used an Incremental Grid Growing implementation where the parameters for growing the grid and generating hierarchical structures were set such that the initial grid layout remained unchanged. As visualisation of the generated mapping, a convenient solution is to use the HTML format, as we are thereby not limited to a specific software; additionally, *links* to the specific documents in the collection can be easily provided. In Figure 6, the HTML representation of the 33 × 17 map with 296 nodes is depicted to give an overview, while Figure 7 shows a section (the province of Vorarlberg, in western Austria) of the very same map.

Table 1. Categories in the Data

Category	Description	Documents
1	Abdominal and Gastrointestinal Abdominal Viscera (Solid Organs) GI Tract	160
2	Breast	80
3	Cardiac	70
4	Chest	60
5	Computer Applications	30
6	Contrast Media	40
7	Genitourinary	70
8	Head and Neck	40
9	Interventional Radiology	130
10	Musculoskeletal	90
11	Neuro	90
12	Pediatric	30
13	Physics in Radiology	40
14	Radiographers	10
15	Vascular	69
	Total	1009

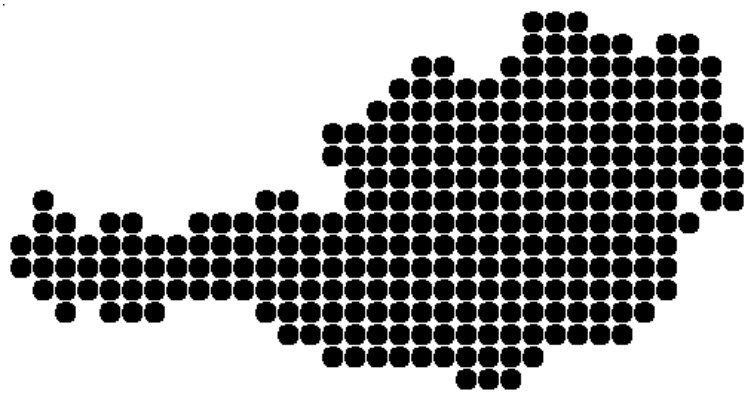

Fig. 4. An IGG in the shape of Austria, 296 nodes.

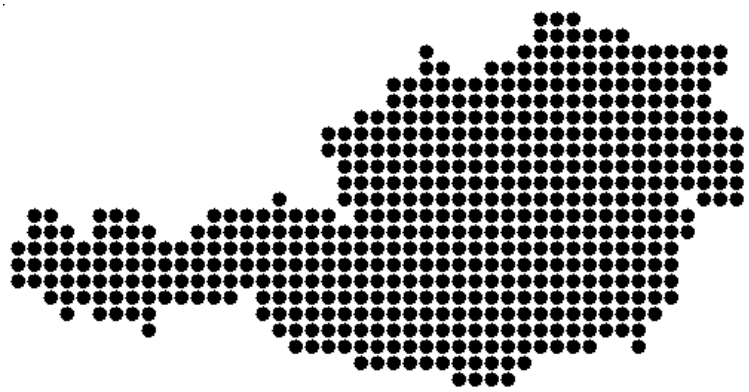

Fig. 5. An IGG in the shape of Austria, 552 nodes.

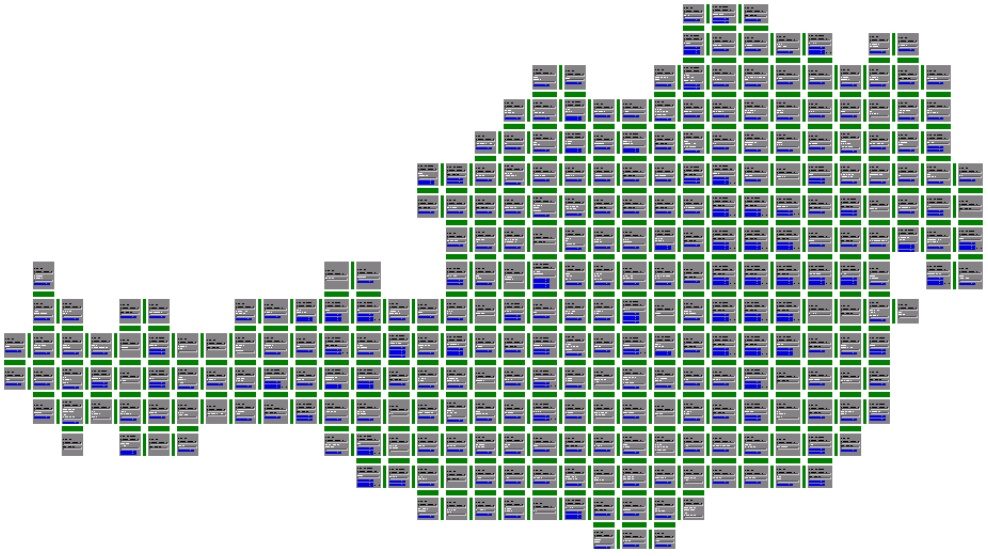

Fig. 6. HTML representation of an IGG in the shape of Austria (296 nodes).

Fig. 7. HTML representation of a section of the IGG showing the province of *Vorarlberg* area of the map (western Austria).

In our HTML representation, nodes are symbolised with gray-coloured boxes, while grid positions which are not occupied by nodes are left white; the map's connectivity is indicated by the existence of green-coloured connections between the individual nodes. The node boxes contain some statistical information on their top (the number of vectors mapped, as well as the so-called *mean quantisation error mqe* of this node as a measure for the mapping quality). Below, the list of *feature labels* describing the mapped data follows. These labels are automatically assigned to the nodes in a map to give some descriptive information about the documents mapped onto the very same node. The method used in our implementation builds on the fact that the most descriptive attributes for a set of input data are the ones shared by all data on a specific node. If a majority of input patterns mapped on a particular node exhibit a highly similar input vector value for a particular feature, the same will apply for the corresponding model vectors; therefore, those model vector elements that show largely the same value for all input patterns mapped on a node may serve as a descriptor for that very node. In text mining applications, we want to describe a cluster of documents by its present features only, in other words, we do not want to describe documents by saying what they are *not* about; therefore, we select only these attributes that have high model vector values.
If there are input patterns mapped on the node, a listing of (some of) the input patterns will appear below the feature labels.

In the section depicted in Figure 7, documents from in total nine different classes have been mapped closely together. Though one might expect a mapping that would more closely resemble the human-generated classification, the result can be explained by the documents mapped in this region - though stemming from different areas - describing experiments and studies that use the same, respectively similar, methods ((like, among others, various different variants of *computed tomography* (CT) and *magnetic resonance*(MR)).

Though the HTML representation has some practical advantages, it can be easily observed that a representation like this is not able to give the user a good overview of the distribution of the documents throughout the map, when the map size becomes to large. Moreover, we did not yet use the additional classification information provided with the data. Therefore, we generated another visualisation, which shows the distribution of the already existing classes over the generated map; examples are depicted in Figures 8 and 9 for the 296 and 552 nodes IGG, respectively. In this representation, the rectangular, coloured shapes represent one node in the IGG which has at least one input pattern mapped onto, while a black "x" stands for an "empty" node, i.e. a node with no input patterns mapped onto. Nodes are coloured according to which classes the documents mapped onto them belong to; below the mapping, a legend shows the association between classes and colours. It is to note that nodes which are coloured with more than one colour represent nodes that contain documents from different classes; then, the width of the coloured areas represent the proportion of documents on this very node from each class.

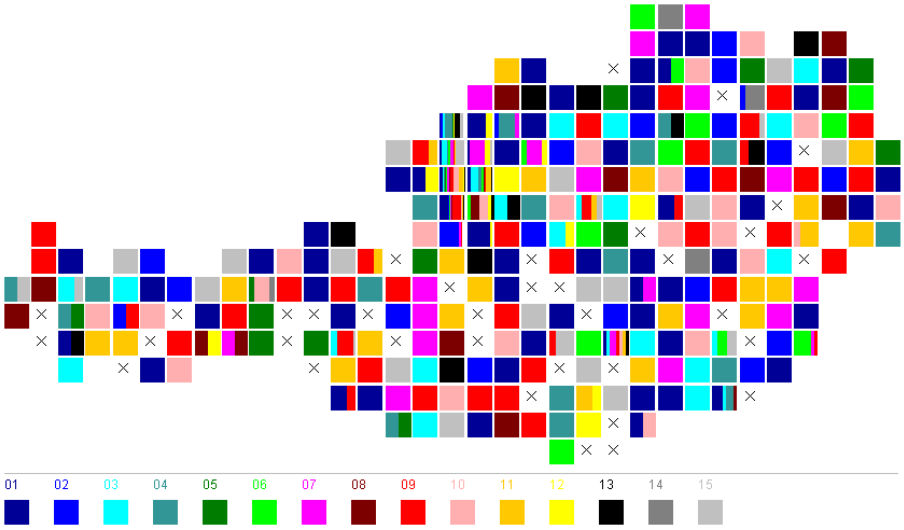

Fig. 8. Graphical representation of the class distribution in a 296 node IGG.

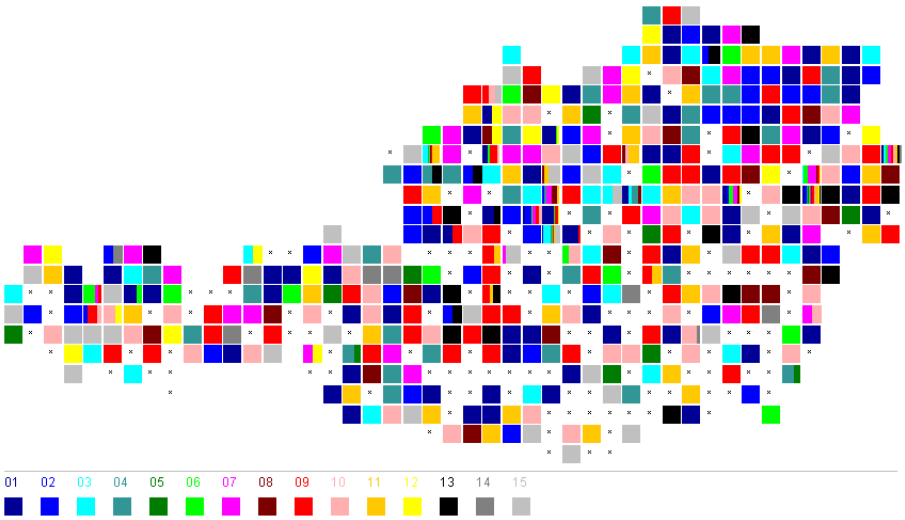

Fig. 9. Graphical representation of the class distribution in a 552 node IGG.

As we could already notice in the Section of the map presented in Figure 7, the distribution of the human-generated classes over the mapping generated by the IGG training algorithm is not what one might expect initially, as documents that belong to the same class are spread almost equally over the whole map. However, the generated mapping is explainable and logical, as documents mapped onto the same or neighbouring nodes have similar content, regardless of the class they belong to, as we have already seen on the example of the nodes in Figure 7.

5 Conclusion

In this paper, we have given a short overview of the Self-Organizing Map (SOM). This neural network algorithm enjoys a growing popularity in the field of Data Mining, as it generates a topology-preserving mapping from high-dimensional data to a lower dimension, which allows for an easier interpretation of complex data. However, convenient and practical methods for *describing* the generated maps have not yet been developed.

Therefore, we applied using shapes for self organising architectures that are easily recognisable by readers, as for example the shapes of countries.
Using these shapes familiar to the reader may allow for an easier explanation of a SOM (or any mapping generated by any other related model) - it allows to refer to regions of the map not by addressing them by the corner of the map they are located in, or by defining the region by its X/Y coordinates in the grid. In the example of our experiments, we can refer to provinces, or parts of provinces, or even cities, and the user - assuming that he or she is familiar with the shape - will immediately know what area of the map is talked about. This technique can become even more useful when we describe sections of a map - the user will easily know where to it belongs in the complete map.

However, though there are some obvious advantages, there are also a couple of points one has to pay attention to when using the here presented method.

Most importantly, care has to be taken not to mix the domain of the document collection with the one of the representation. For example, using a human body as a shape for a map while analysing the collection presented in this paper (i.e., a collection of documents from the domain of medicine) will lead to confusion and discussions, rather than an easier explanation of the map (e.g. when documents describing knee ligament injuries are mapped onto nodes that represent the region of the lungs, etc.).
Similarly, using a collection of documents that somehow contains geographic information will render using a map shaped like a country or continent obsolete (as a reader would just wonder why, for example, some documents originally describing province A get mapped onto nodes in province B).

Additionally, one has to keep in mind that specific shapes might not be well-known to all users. Then, users not familiar with this shape will, instead of getting an easier understandable explanation of the map, have additional diffi-

culties understanding the presented map. Therefore, it is absolutely necessary to have a clear definition of one's target audience.

The main advantage of recognisable shapes, i.e. the better explainable power of a generated mapping, however, can not be verified without testing it in experiments with users.

References

[BM93] Justine Blackmore and Risto Miikkulainen. Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map. In *Proc. ICNN'93, International Conference on Neural Networks*, volume I, pages 450–455, Piscataway, NJ, 1993. IEEE Service Center.

[Koh82] Teuvo Kohonen. Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[Sal89] Gerald Salton. *Automatic text processing – The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley Longman Publishing Co., Inc., 1989.