

RANKED CENTROID PROJECTION: A DATA
VISUALIZATION APPROACH BASED ON SELF-
ORGANIZING MAPS

By

ZHENG WU

Bachelor of Engineer in Electrical Engineering
Beijing Jiaotong University
Beijing, China
1997

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
July, 2006

RANKED CENTROID PROJECTION: A DATA
VISUALIZATION APPROACH BASED ON SELF-
ORGANIZING MAPS

Dissertation Approved:

Dr. Gary G. Yen

Dissertation Adviser

Dr. Keith Teague

Dr. Rao Yarlagadda

Dr. Camille DeYong

Dr. A. Gordon Emslie

Dean of the Graduate College

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to my research advisor, Dr. Gary G. Yen, for his guidance and instruction throughout my Ph.D. studies, and for the opportunity to work with his research group for the past few years. While the knowledge that he has helped me gain through books, papers and research will last me a lifetime, it is watching and learning from him how to achieve a successful professional career that I value the most. I am also thankful Dr. Keith Teague, Dr. Rao Yarlagadda, and Dr. Camille DeYong, for their valuable comments and suggestions about my research and for serving on my dissertation committee.

I would like to thank Dr. Jerzy Krasinski and Dr. Jack Allison for offering me teaching assistantship, which provided financial support to me when it was needed.

I am grateful to all my colleagues, both the past and present members of the Intelligent Systems and Control Laboratory (ISCL) for their friendship and help. It was a great pleasure to work with them, and I shall never forget the experience. I also appreciate the help of Steven A. Morris, whose ideas on the SOM mapping were the inspiration for this study.

Finally, I would like to give my special thanks to my parents, Junjiao Wu and Younan Yi, who have supported me throughout the whole journey. I do not think I would have ended up here if it were not for their unconditional love, encouragement and advice. I hope that some day I will be able to repay them for all they have done for me.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION.....	1
II. THE SELF-ORGAINING MAP (SOM) ALGORITHM.....	6
2.1 An Overview of the SOM.....	6
2.2 Structure.....	7
2.3 Initialization.....	9
2.4 Training.....	10
2.4.1 Two-step Training.....	10
2.4.2 Analysis of the Update Rule.....	12
2.4.3 Neighborhood Function.....	13
2.4.4 Learning Rate.....	14
2.5 Mathematical Treatment.....	16
2.5.1 Convergence.....	16
2.5.2 Energy Function.....	17
2.6 Dynamic SOM Models.....	18
2.6.1 Growing Cell Structure.....	18
2.6.2 Growing Neural Gas.....	19
2.6.3 Incremental Grid Growing.....	19
2.6.4 Other Growing Structure Models.....	20
2.6.5 Hierarchical Models.....	21
III. VISUALIZING SELF-ORGANIZING MAPS.....	23
3.1 Visualizing Map Topology.....	23
3.2 Visualizing Data Density.....	25
3.3 Visualizing Prototype Vectors.....	27
3.4 Visualizing Component Planes.....	29
3.5 Visualizing Best Matching Units.....	30
3.6 Other Visualizations.....	32
IV. VISUALIZATION OF A SCIENTIFIC DOCUMENT COLLECTION.....	34
4.1 Visualizing Documents in General.....	34
4.2 Document Encoding.....	35

4.2.1 Inter-Document Links.....	35
4.2.2 The Vector-Space Model (VSM).....	37
4.2.3 Latent Semantic Indexing (LSI).....	38
4.2.4 Citation-Based Models.....	40
4.2.4.1 Direct and Indirect Citation Links	40
4.2.4.2 Clustering Documents Using Bibliographic Coupling	42
V. RANKED CENTROID PROJECTION.....	44
5.1 The GHSOM Architecture.....	45
5.2 Projection Method.....	49
5.2.1 An Approach Using Weighted Average	49
5.2.2 Improving the Weighting Function by Applying a Ranking Scheme.....	52
5.2.3 Illustrations of the RCP.....	53
5.2.4 Selecting the Ranking Parameter R	56
5.2.4.1 Effect of R on the Projection Result	56
5.2.4.2 Criteria for Selecting R	58
5.3 Incremental Clustering for Dynamic Document Databases.....	61
5.3.1 The Need for Incremental Clustering	61
5.3.2 Formation of the Similarity Matrix.....	61
5.3.3 Dynamic Growth of the Similarity Matrix.....	63
5.3.4 An Incremental Document Clustering Algorithm.....	65
VI. SIMULATION RESULTS	67
6.1 Overview.....	67
6.2 Software Tools Used.....	68
6.3 An Illustrative Data Set.....	68
6.4 Document Data Sources.....	73
6.5 A Collection of Journal Papers on Self-Organizing Maps.....	74
6.6 A Collection of Journal Papers on Anthrax Research	79
6.7 Adding New Documents to an Existing Map	83
VII. DESIGN OF THE SOFTWARE TOOLBOX.....	88
7.1 Overview.....	88
7.2 System Description	89
7.2.1 Document Sources	90
7.2.2 Project Database.....	92
7.2.3 Similarity Matrix.....	92
7.2.4 Mapping of Documents.....	93
7.2.5 Clustering of Documents	93
7.2.6 Visual Exploration	94

7.3 Getting Started with the Menu Bar	94
7.3.1 Working with a Project	95
7.3.2 Building Similarity Matrix.....	96
7.3.3 Training the GHSOM	97
7.4 Displaying and Working with Maps	99
7.4.1 Displaying Maps	99
7.4.2 Exploring Maps.....	101
7.4.2.1 Clustering and Marking Documents	101
7.4.2.2 Visualization of Document Links	104
7.4.2.3 Visualization of Document Dates	105
7.4.2.4 Visualization of Document Citation Counts	107
 VIII. CONCLUSION AND FUTURE WORK	 108
8.1 Summary of the Work.....	108
8.2 Advantages and Limitations	109
8.3 Future Directions	111
 REFERENCES	 114

LIST OF TABLE

Table	Page
V.1 Basic steps of the growth in width	47
V.2 Basic steps of the growth in depth	48
V.3 Basic steps of the incremental clustering algorithm	66
VI.1 Quantitative evaluation results	73
VI.2 List of new papers mapped to the cluster labeled “gene expression data analysis” .	86

LIST OF FIGURES

Figure	Page
II.1 The SOM grid structure: (a) rectangular grid (b) hexagonal grid.....	8
II.2 Two types of SOM topologies (a) planar topology (b) toroidal topology	8
II.3 Two neighborhood functions: (a) bubble neighborhood, (b) Gaussian neighborhood	14
II.4 Simulation results of different models: (a) the SOM, (b) GCS, (c) GNG, and (d) GG. The distribution is uniform in the shaded area. Map units are denoted by circles.	20
III.1 U-matrix presentations of a 10×10 rectangular SOM: (a) a gray-level image and (b) a 3D plot. The Iris data set is used to train the SOM.	24
III.2 Different visualizations of the SOM: (a) The original data set of a mixture of two Gaussians, (b) U-matrix presentation, (c) P-matrix presentation, (d) U*-matrix presentation.	26
III.3 Different ways to visualize the prototype vectors: (a) MDS projection of a SOM, (b) Sammon’s mapping of a SOM. Neighboring map units, depicted as black dots, are connected to each other. The Iris data set is used to train the SOM.	28
III.4 Component planes representation of a SOM trained with the Iris data set. The color bars beside each component planes show the maximum, mean, and minimum values and the corresponding colors.	30
III.5 Different presentations of the hit histogram: (a) a gray level image, and (b) a 3D plot. The Iris data set is used to train the SOM.	31
IV.1 The implicit links between documents based on the document-term relations. Documents D_i (circles) are connected by common terms t_i (black dots).....	36
IV.2 Illustration of citation links.....	41
IV.3 Illustration of four types of citation links between a pair of documents. A pair of documents D_i and D_j are connected by a direct citation (DC) and three forms of	

indirect citation links: bibliographic coupling (BC), co-citation (CC), and longitudinal coupling (LC). The shaded circles represent the corresponding third documents in the indirect links.	42
V.1 The schematic diagram of the proposed SOM-based approach.....	45
V.2 Graphic representation of a trained GHSOM	46
V.3 Illustration of the growing process: (a) A row or (b) a column of units is inserted in a SOM [Rauber et al. 02]. Unit e is the error unit and d is its most dissimilar neighbor. The shaded circles denote the newly inserted units.....	48
V.4 Illustration of mapping an input vector by finding the centroid of the spatial histogram of the output values.....	51
V.5 Illustration of mapping an input vector x_i (marked by a cross) to its BMU. i is the index of each map unit. w_c denotes the BMU. w_2 and w_3 are the next two closest units.....	54
V.6 Illustration of mapping an input vector when two units are considered. d_1 and d_2 are the distances in the input space. d_1' and d_2' are the distances in the output space.....	55
V.7 Illustration of mapping an input vector when three winners are considered. d_1 , d_2 and d_3 are the distances in the input space. d_1' , d_2' and d_3' are the distances in the output space.	55
V.8 Data set I: Samples in a three-dimensional data space marked as small circles and prototype vectors as plus signs.	57
V.9 Projection results with different R values	58
V.10 (a) Sammon's stress and (b) DB index for $R = 1, 2, 3, 4$	59
V.11 The optimal R is found at the minimal point of the signal cost function.	60
VI.1 The resulting 3-layer GHSOM for the zoo data set.....	69
VI.2 Two-dimensional projections of the zoo data. (a) The first-layer map using the proposed approach ($R=3$). (b) A 9×9 SOM with data projected to the corresponding BMUs. (c) PCA. (d) Sammon's mapping.	70
VI.3 Submap of invertebrates generated with $R=3$	72
VI.4 The resulting 3-layer GHSOM for the collection of SOM papers.....	75

VI.5 The projection result of the journal papers on the SOM, where documents are marked as circles in the rectangular area.	76
VI.6 An enhanced visualization of the SOM papers, where the size of the document marker is proportional to the number of times a document has been cited.	77
VI.7 A submap of the resulting GHSOM for the SOM paper collection	78
VI.8 The resulting 3-layer GHSOM for the collection of anthrax papers	79
VI.9 First-layer projection of the anthrax journal papers, where documents are marked as circles in the rectangular area. The cluster labels are added manually showing the major subject of each cluster of papers.	80
VI.10 Labeled map of the anthrax paper set with paper marker sizes proportional to the number of times the corresponding papers have been cited.	81
VI.11 One submap expanded from the upper left neuron in the first-layer map.	83
VI.12 The second-layer document projection of the upper-left cluster in the preceding layer.	83
VI.13 The first-layer map of the initial 600 papers for training	84
VI.14 The first-layer map with the additional 38 papers added as new data. The red circles represent newly added papers.	84
VI.15 3D timeline of the 638 SOM papers	86
VII.1 Main GUI of the Document Visualization and Analysis Toolbox	89
VII.2 Flow of work of the document visualization and analysis toolbox	90
VII.3 A typical journal paper printed in a text file	91
VII.4 Documents are saved in tabular format in the project database.	92
VII.5 Click on <i>New Project</i> to create a new project	95
VII.6 Selecting a database from the list	96
VII.7 Load paper-reference matrix	97
VII.8 Create similarity matrix	97

VII.9 Load similarity matrix to current project	97
VII.10 Select similarity matrix	98
VII.11 Start GHSOM training.....	98
VII.12 Error message	98
VII.13 Set training parameters	98
VII.14 The two-dimensional representation of the GHSOM structure	99
VII.15 Enter the name of the map.....	99
VII.16 The Map control group.....	99
VII.17 The dot map for the first-layer map.....	100
VII.18 (a) Landscape map (b) Contour map.....	100
VII.19 The CLUSTERS control group	101
VII.20 The selected cluster is highlighted.	102
VII.21 Titles of the highlighted documents	102
VII.22 Enter the cluster name	103
VII.23 The cluster name appears in the window	103
VII.24 Labeled map	103
VII.25 The CONNECTIONS window.....	104
VII.26 Select the desired link.....	104
VII.27 Different types of connections: (a) All, (b) Group, (c) Dependent, (d) Precedent.	105
VII.28 The TIME control window	106
VII.29 Documents published within a specific time period are highlighted	106
VII.30 A dot map with document citation counts visualized.....	107

CHAPTER I

INTRODUCTION

Text has been used as a major repository of knowledge. The published scientific documents constitute in many cases the primary source of information for a wide range of application domains. With greater accessibility of scientific documents in electronic form, we now have an opportunity to better identify and extract useful information from this vast and rich data source.

For analyzing textual data, it is necessary to pre-process the documents and encode their textual contents into some kinds of metric feature vectors. Like any other type of real-world data, textual data is often of large volume and represented in high-dimensional spaces. Its inherent patterns and hidden relationships are hence hard to recognize and illustrate. Data visualization has established itself as one powerful tool to explore and interpret raw data. “A picture is worth 1,000 words,” the old adage goes, which points out the value of displaying quantitative information as images. Data visualization helps to reveal the hidden information within data that cannot be easily detected in any other way. The most obvious and popular means of data visualization is to use maps to facilitate a spatial understanding of data. It is usually achieved by mapping the data into geometric attributes so that the similarity measures between data objects are associated with the geometric proximity of objects in some spatial configuration. Maps

present an ideal information space for displaying various types of geographically based data, using concrete or abstract symbols [Olsen 05]. In the context of textual data, such representations are called document maps. In a document map, documents are represented as points on a two-dimensional plane and the geometric relations of the points depict the similarity relations between different documents. Document maps help us gain insight into the information hidden in a large collection of documents, such as the cluster structure, conceptual relationships and emerging developments. We may visually spot documents not directly related to each other but potentially indirectly related. The topics of such documents may be good candidates for knowledge integration. Such maps may also aid us in identifying seminal papers representing the evolution of a subject domain.

However, due to the limitation in human visual and cognitive perception, the complexity of transformed numeric data needs to be reduced in order for a comprehensive map to be realized in case of large data volumes and data with a dimensionality higher than three. There exist several data analysis methods that are able to reduce the data complexity. They can be broadly divided into two categories, clustering methods [Hartigan 75, Jain and Dubes 88] and projection methods [Davison 83, Sammon 69]. The first category aims at reducing the amount of data by grouping similar data patterns together. A similarity measure such as Euclidean distance is used on pairs of data patterns to characterize how they are distributed in a multidimensional space. Intuitively, patterns within a cluster are more similar to each other than they are to patterns belonging to different clusters. Clustering is useful when dealing with a large complex data set with many variables and unknown structure. However, some additional

illustration methods are usually needed to facilitate our understanding of the clusters. The second category of data analysis methods emphasizes reducing the dimensionality of the data. The goal of the projection is to represent the data items in a lower-dimensional space in such a way that certain properties of the structure of the data set are preserved as faithfully as possible. The projection can be used to visualize the data set if a sufficiently small output dimensionality is chosen [Kaski 97]. The Self-Organizing Map (SOM) [Kohonen 82, Kohonen 95] is a special case in that it can be used at the same time for both clustering and projection. As a result, considerable interest has been devoted to the SOM for the purpose of visualizing textual data.

The SOM, originated by Kohonen [Kohonen 82], is a neural network paradigm based on unsupervised competitive learning. It creates prototype vectors representing the data and projects high-dimensional data onto a low-dimensional grid, usually two-dimensional. The similarity inherent in the input data is reflected by the geometric relationships between the grid units. Similar input vectors are mapped close to each other, while dissimilar ones are mapped far apart. Therefore the cluster structure and other patterns of the data can be identified visually from the map created. It is a very intuitive and straightforward way to visualize the data structure. Because of its implicit ability in dimensionality reduction, the SOM has been popularly used as a data clustering and visualization tool.

As noted above, the SOM is a useful tool for organizing and visualizing large complex data sets. Document visualizations may be regarded as an interesting application domain for the SOM, since the documents can be encoded statistically into a set of very high-dimensional feature vectors. In this research, we focus on how the clustering and

visualization ability of the SOM can be enhanced and employed to produce meaningful document maps. As the major subject of our study is a collection of scientific publications, which have the distinct property of containing bibliographic citations, the inter-document similarities are based on the citation patterns. These citation patterns provide explicit linkages between publications having particular points in common, and hence are considered as reliable indicators of intellectual connections among documents. Inter-document similarities calculated from citations generally produce meaningful document maps whose patterns expose clusters of documents and relations among those clusters [Morris et al. 02]. Citation based similarity measures are used extensively when working with journal articles and patents from sources like the Science Citation Index (SCI). The SCI provides access to current and retrospective citation information for scientific literature published in the physical, biological and medical fields. It presents a great source of information for citation based document analysis.

SOM's capability to deal faithfully with very high-dimensional data, especially in the analysis of document collections, is explored and discussed in the work presented here. A novel visualization method in conjunction with a dynamic SOM model, namely the Growing Hierarchical Self-Organizing Map, is proposed for reflecting the cluster structure of the underlying structure presented in the data with variable resolutions. The SOM-based approach is used both as a tool for analyzing the data, as well as serving directly as the basis for a corresponding interface to the data collections.

The remainder of this report consists of seven chapters. In Chapter II, a review of the previous work on the SOM and its variations is introduced. Chapter III provides a review of various approaches to visualize the SOM clustering results. Chapter IV

overviews various existing document encoding models. The details of the proposed SOM-based approach, the Ranked Centroid Projection, are then presented in Chapter V. Chapter VI provides illustrative examples demonstrating the principles of the algorithm and comparisons with some state-of-the-art methods. Based on the approach developed in this research, a user-friendly software toolbox for analyzing and visualizing document collections is designed and presented in Chapter VII. Finally, Chapter VIII concludes this report with a few pertinent observations. Directions for continued research to extend the research are also discussed.

CHAPTER II

THE SELF-ORGANIZING MAP (SOM) ALGORITHM

2.1 An Overview of the SOM

The Self-Organizing Map (SOM) is a neural network paradigm for exploratory data analysis. The idea of the SOM was originally motivated by the localized regions of activities in the human cortex, where similar regions react to similar stimuli. This model stems from Kohonen's work [Kohonen 82] and builds upon earlier work of Willshaw and von der Malsburg [Willshaw and von der Malsburg 76]. As a data analysis tool, the SOM can be used at the same time both to reduce the amount of data by clustering, and to project the data nonlinearly onto a lower-dimensional display [Kohonen 95]. Because of its benefits, the SOM has been used in a wide variety of scientific and industrial applications such as image recognition, signal processing, and natural language processing. In the research community, it has received significant attention in the contexts of clustering, data mining, topology preserving vector projection for high-dimensional input spaces, and visualization.

The SOM is equipped with an unsupervised and competitive learning algorithm. It consists of an array of neurons placed in a regular, usually two-dimensional grid. Each neuron is associated with a weight vector (or prototype vector). Similar to other

competitive networks, the learning rule is based on weight adaptations. In the original design of the SOM, only one neuron (winner) at a time is activated corresponding to each input. The presentation of each input pattern results in a localized region of activity in the SOM network. During the learning process, a sufficient number of different realizations of the input patterns are fed to the neurons so that the neurons become tuned to various input patterns in an orderly fashion. The principal goal of the SOM is to adaptively transform an incoming pattern of arbitrary dimension into the low-dimensional SOM grid. The locations of the responses in the map grid tend to become ordered in the learning process as if some meaningful nonlinear coordinate system for the different input features were being created over the network. This projection can be visualized in numerous ways in order to reveal the characteristics of the underlying input data or to analyze the quality of the obtained mapping [Pözlbauer et al. 05].

2.2 Structure

The neurons, or map units, in a SOM are usually placed in a regularly spaced one-, two- or higher dimensional grid. The two-dimensional grid is most commonly used because it provides more information than the one-dimensional and is less problematic than the higher dimensional ones. The positions of the neurons in the grid are fixed, so they won't move during the training phase of the SOM.

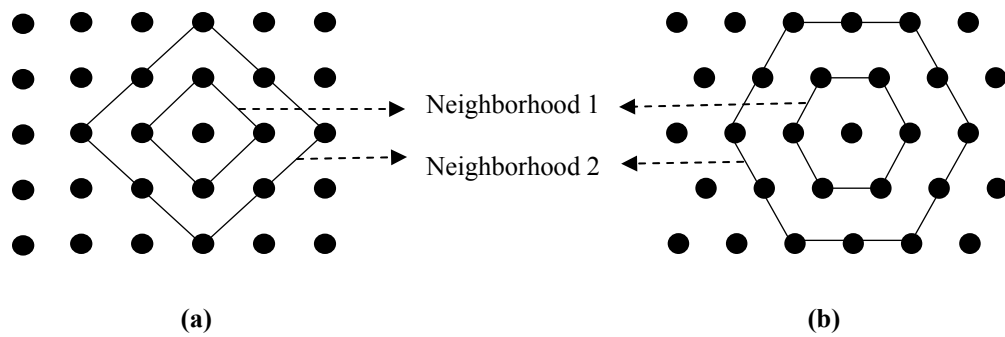


Figure II.1 The SOM grid structure: (a) rectangular grid (b) hexagonal grid

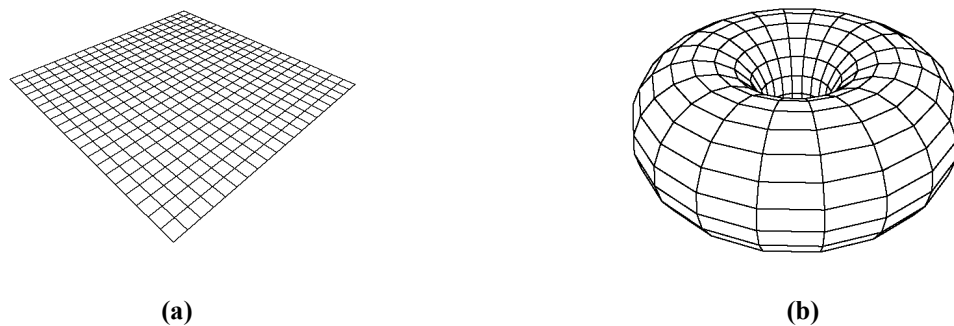


Figure II.2 Two types of SOM topologies (a) planar topology (b) toroidal topology

The neurons are connected to adjacent neurons by a neighborhood relation, which dictates the structure or topology of the map. The neurons most often are connected to each other via a rectangular or hexagonal grid structure. The grid structures are illustrated in Figure II.1, where neurons are marked with black dots. Each neuron has neighborhoods of increasing diameter surrounding it. The neighborhood size controls the smoothness and generalization of the mapping. Neighborhoods of different sizes in both topologies are also illustrated in Figure II.1. Neighborhood 1, the neighborhood of diameter 1, includes the center neuron itself and its immediate neighbors. The neighborhood of diameter 2 includes the neighborhood 1 neurons and their immediate neighbors. The map topology is usually planar but toroidal topologies [Ultsch and

Mörchen 05] have also been suggested. Figure II.2 illustrates these two types of topologies.

2.3 Initialization

In the basic SOM algorithm, the layout and number of neurons are determined before training. They are fixed from the beginning. The number of neurons determines the resolution of the resulting map. A sufficiently high number of neurons should be chosen to obtain a map with a decent resolution. Yet, this number should not be too high, as the computational complexity grows quadratically with the number of neurons [Vesanto and Alhoniemi 00].

Each neuron in the SOM is associated with an n -dimensional weight vector $w_i = [w_{i1}, w_{i2}, \dots, w_{in}]^T$ in the space of the data, where n is the dimension of the input vectors and T denotes the matrix transpose. The weight vector is often referred to as the prototype vector. In this study, the terms *weight vector* and *prototype vector* are used interchangeably. Before the training phase, initial values are assigned to the weight vectors. Three types of network initializations are proposed by Kohonen [Kohonen 95]:

- random initialization, where random values are assigned to weight vectors. This is the case if little is known about the input data at the time of the initialization.

- initialization using initial samples, which has the advantage that the initial locations of the weight vectors lie in the same part of the input space as the data points.
- linear initialization, where the weight vectors are initialized to lie in the linear subspace spanned by two largest eigenvectors of the input data. This helps to stretch the SOM to the orientation in which the input data set has the most significant amount of information.

2.4 Training

The SOM is an unsupervised neural network, which means the training of a SOM is completely data-driven. No external supervisor is available to provide target outputs. The SOM learns only from the input vectors through repetitive adaptations of the weight vectors of the neurons.

2.4.1 Two-step Training

The training process of the SOM uses the repeated application of a two-step learning rule. At each time step, one input vector x is drawn randomly from the input data set and presented to the network. The training consists of two essential steps:

1) Winner selection

This step is often called *competition*. For each input pattern, a similarity measure is calculated between it and all the weight vectors of the map. The neuron with the greatest similarity with the input vector will be chosen as the winning neuron, also called

the best-match unit (BMU). Usually the similarity is defined by a distance measure, typically Euclidian distance. Therefore the winner, denoted as c , is the neuron whose weight vector is the closest to the data sample in the input space. This can be defined mathematically as the neuron for which

$$c = \arg \min_i \{ \|x - w_i\| \}. \quad (\text{II.1})$$

2) Updating weight vectors

After the winner is determined, the winning unit and its neighbors are adjusted by modifying their weight vectors towards the current input according to the update rule formulated as

$$w_i(t+1) = w_i(t) + \alpha(t)h_{ci}(t)[x(t) - w_i(t)], \quad (\text{II.2})$$

where $w_i(t)$ is the weight vector associated with unit i at time t and $x(t)$ is the input vector randomly drawn from input set at time t . $\alpha(t)$ is the learning rate function and a scalar parameter monotonically decreasing with t . $h_{ci}(t)$ is a non-increasing neighborhood function centered on the winner unit at time t .

This adaptation rule of the weights is closely related to the k -means clustering. The weight vector of each neuron represents a cluster center. Like the k -means, the weight of the best matching neuron (cluster center) is updated in a small step in the direction of the input vector x . However, unlike k -means, the winner, and also the neurons surrounding it, are updated instead of the winner alone. The size of the surrounding region is specified by $h_{ci}(t)$, which is a non-increasing function of time and of the distance of neuron i from the winner c .

As a result of the update rule, the neuron whose weight vector is the closest to the input vector is updated to be even closer. Consequently the winning unit is more likely to

win the competition the next time a similar input sample is presented, while less likely to win when a very different input sample is presented. As more input samples are presented to the network, the SOM gradually learns to recognize groups of similar input patterns in such a way that neurons physically close together on the map respond to similar input vectors.

2.4.2 Analysis of the Update Rule

The update rule in Equation II.2 can be rewritten as

$$w_i(t+1) = [1 - \alpha(t)h_{ci}(t)]w_i(t) + \alpha(t)h_{ci}(t)x(t). \quad (\text{II.3})$$

This equation characterizes the influence of data samples during training and directly shows how the parameters, $\alpha(t)$ and $h_{ci}(t)$, affect the motion of w_i . Every time a data sample $x(t)$ is presented to the network, the value of $x(t)$, scaled down by $\alpha(t)h_{ci}(t)$, is superimposed on w_i and all previous values $x(t')$, $t' = 0, 1, \dots, t-1$, are scaled down by the factor $[1 - \alpha(t)h_{ci}(t)]$, which we assume < 1 . The contribution of the data samples can be shown more clearly by rewriting Equation II.3 into a non-iterative form.

Given $w_i(0)$ as the initial condition, Equation II.3 can be transformed into the following form by iteratively substituting $w_i(t')$ with $w_i(t'-1)$, $t' = t, t-1, \dots, 1$,

$$w_i(t+1) = A(t+1)w_i(0) + \sum_{n=1}^t B(t+1, n)x(n). \quad (\text{II.4})$$

The coefficient $A(t)$ describes the effect of the initial weight value on $w_i(t)$ and $B(t, n)$ describes the effect of the data point presented at time n on $w_i(t)$. Both $A(t)$ and $B(t, n)$ are functions of $\alpha(t)h_{ci}(t)$ and decrease with t [Mulier and Cherkassky 94].

Equation II.4 shows that $w_i(t+1)$, the weight vector at time $t + 1$, depends on a weighted sum of the initial condition and every data points presented to the network. $w_i(t+1)$ can be therefore considered as a “memory” of all the values of $x(t')$, $t' = 0, 1, \dots, t$. As the weight function $B(t,n)$ is a function of $\alpha(t)$ and $h_{ci}(t)$, the influence of a training sample on the final weight vector depends on the specific learning rate and neighborhood function used during the self-organizing process.

2.4.3 Neighborhood Function

The neighborhood function is a non-increasing function of time and of the distance of unit i from the winner neuron c . The form of the neighborhood function determines the rate of change around the winner neuron. The simplest neighborhood function is the bubble function as shown in Figure II.3a, which is constant over the defined neighborhood of the winner unit and zero elsewhere. Using the bubble neighborhood function, every neuron in the neighborhood is updated the same proportion of the difference between the unit and the presented sample vector.

Another widely applied, smooth neighborhood function is the Gaussian neighborhood function

$$h_{ci}(t) = \exp\left[\frac{-\|r_c - r_i\|^2}{2\sigma(t)^2}\right], \quad (\text{II.3})$$

where $\sigma(t)$ is the width of the Gaussian kernel and $\|r_c - r_i\|^2$ is the distance between the winner c and the neuron i with r_c and r_i representing the two-dimensional positions of neurons c and i on the SOM grid respectively. The Gaussian neighborhood function is illustrated in Figure II.3b. Usually the radius of the neighborhood is large at first and

decreases during the training. One commonly used form [Ritter et al. 92] of $\sigma(t)$ is given by

$$\sigma(t) = \sigma(0) \left(\frac{\sigma(f)}{\sigma(0)} \right)^{\frac{t}{t_{max}}}, \quad (\text{II.4})$$

where $\sigma(0)$ is the initial neighborhood radius, $\sigma(f)$ is the final neighborhood radius, and t_{max} is the number of training iterations. Therefore $\sigma(t)$ is a monotonically decreasing function of time. The decreasing neighborhood radius ensures that the global order is obtained at the beginning, whereas towards the end the local corrections of the weight vectors of the map will be more specific [Honkela 97, Vesanto 97].

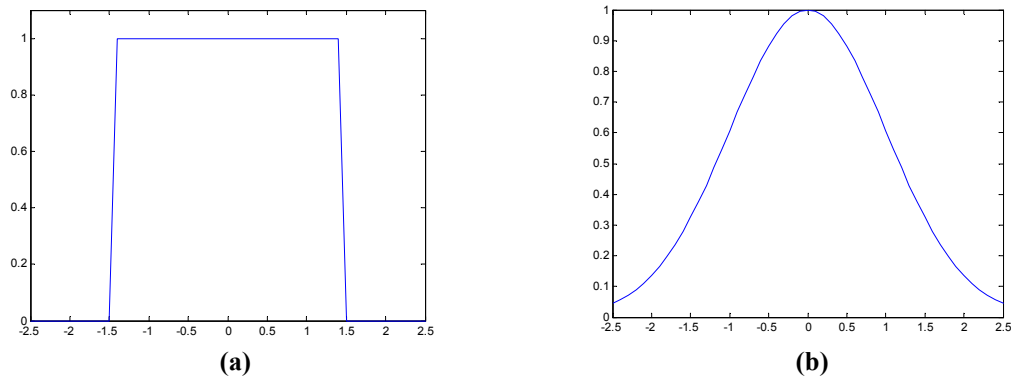


Figure II.3 Two neighborhood functions: (a) bubble neighborhood, (b) Gaussian neighborhood

2.4.4 Learning Rate

The learning rate $\alpha(t)$ is a function decreasing with time, which can be linear, exponential or inversely proportional to time. The linear learning rate function [Hollmén 96] can be defined as

$$\alpha(t) = \alpha(0)(1-t/t_{max}), \quad (\text{II.5})$$

where $\alpha(0)$ is the initial learning rate. A commonly used exponentially decreasing function [Cherkassky and Lari-Najafi 91] is given by

$$\alpha(t) = \alpha(0) \left(\frac{\alpha(f)}{\alpha(0)} \right)^{\frac{t}{t_{\max}}}, \quad (\text{II.6})$$

where $\alpha(f)$ is the final learning rate. A function inversely proportional to time is given in [Mulier and Cherkassky 94] with the form

$$\alpha(t) = \frac{1}{m(t-1) + 1}, \quad m = \frac{1 - N/t_{\max}}{N - N/t_{\max}}, \quad (\text{II.7})$$

where N is the total number of neurons. Using the learning rate function in Equation II.7 ensures that earlier and later input samples have approximately equal effects on the training result.

The learning rate and the neighborhood function together determine which neurons and how much these neurons are allowed to learn. These two parameters are usually altered during training through two phases. In the first phase, namely the ordering phase, relatively large initial learning rate and neighborhood radius are used. The parameters keep decreasing with time. During this phase, a comparatively large number of weight vectors are to be updated and they move in big steps towards the input samples. In the second phase, the fine tuning phase, both parameters start with small values from the beginning. They continue to decrease but very slowly. The number of iterations for the second phase should be much larger than that in the first phase, as the tuning usually takes much longer [Demuth 98].

2.5 Mathematical Treatment

The mathematical proofs of the SOM properties have turned out to be very difficult. A thorough analysis was performed only for the one-dimensional case, while for

higher dimensions the analysis has not been successfully completed to date. Some of the mathematical properties of the SOM are listed in this section.

2.5.1 Convergence

The SOM algorithm starts with total disorder and gradually self-organizes over t_{max} iterations. In practice under a wide variety of conditions for different input distributions, $\alpha(t)$ and $h_{ci}(t)$, it has been found that the weight vectors will converge to the ordered configuration, which can be defined [Erwin et al. 91] as

$$|r - s| < |r - q| \Leftrightarrow |w_r - w_s| < |w_r - w_q|, \forall r, s, q \quad (\text{II.8})$$

where r , s , and q are neurons while w_r , w_s , and w_q are the associated weights.

No general analysis of the convergence exists and little is known about the essential conditions required for self-organization. The first complete proof of the convergence of the SOM learning process in the one-dimensional case was given in [Cottrell and Fort 87]. The authors have proven that, for uniform distribution of the inputs and a step-neighborhood function, the weight values will almost surely converge to the ordered configuration. This result was further generalized to a very large class of input distribution [Bouton and Pagés 93]. Erwin et al. extended the proof of self-organization [Erwin et al. 92a, Erwin et al. 92b] to include all monotonically decreasing neighborhood functions and investigated the effect of various types of neighborhood functions on the convergence rates of the SOM.

2.5.2 Energy Function

As opposed to many other neural algorithms, the SOM can not be associated to a global decreasing energy function [Erwin et al. 92a]. However, in the case of a discrete input set and a fixed neighborhood function, the SOM has been shown to have an energy function [Ritter et al. 92]:

$$E = \sum_k \sum_i h_{ci} \|x_k - w_i\|^2, \quad (\text{II.9})$$

where the index c depends on the input x_k and the weight vector w_i . The SOM learning rule, Equation II.2, corresponds to a stochastic gradient descent on this energy function. Equation II.9 closely resembles that of the k -means clustering algorithm, which is given as

$$E_K = \sum_k \|x_k - w_{c(k)}\|^2, \quad (\text{II.10})$$

where $w_{c(k)}$ is the centroid closest to x_k . The difference is that the SOM takes into account the distance of x_k from all the weight vectors, instead of just the closet one, weighted by the neighborhood function.

The energy function of the SOM, Equation II.9, can be decomposed into two terms as follows [Lampinen and Oja 92, Kaski 97, Vesanto 97]:

$$E = \sum_k \|x_k - n_c\|^2 + \sum_i \sum_j h_{ij} N_i \|n_i - w_j\|^2, \quad (\text{II.11})$$

where N_i is the number of data samples closest to the weight vector w_i , and n_i is their centroid. The first term in the equation corresponds to the energy function of the k -means clustering, i.e., the average distance from the data points to the nearest cluster centers. The second term corresponds to the ordering of the map, which is minimized when nearby map units have weight vectors close to each other in the input space.

2.6 Dynamic SOM Models

In spite of the wide-spread use of the SOM, some shortcomings have been noted, which are related to the static architecture of the basic SOM model. First of all, the topology of the model, in terms of the number and the layout of the neurons, has to be determined before training. The need for predetermining a fixed network structure is a significant limitation on the final mapping [Fritzke 94, Fritzke 95a, Fritzke 95b, Alahakoon et al. 00]. To address the issue of static SOM architecture, several variations based on the basic SOM have been developed recently. The resulting dynamic SOM models usually employ an incremental growing architecture to cope with the lack of prior knowledge about the number of map units. Some of the models are summarized in this section.

2.6.1 Growing Cell Structure

One of the first models of such kind is the Growing Cell Structures (GCS) [Fritzke 94]. In the GCS, the basic two-dimensional grid of the SOM is replaced by a network of nodes whose basic building blocks are triangles. Starting with a triangle structure of 3 nodes, the algorithm both adds new nodes to and removes existing nodes from the network during the training process. The connections between nodes are adjusted in order to maintain the triangular connectivity. A local error measure is used to decide the position to insert a new node, which is usually between the node with the

highest accumulated error and its most distant neighbor. The algorithm results in a network graph structure consisting of a set of nodes and the connections between them.

2.6.2 Growing Neural Gas

In addition to the GCS, Fritzke has also proposed the Growing Neural Gas (GNG) [Fritzke 95a] and the Growing Grid (GG) [Fritzke 95b].

The GNG algorithm combines the GCS and the Neural Gas algorithm [Martinetz and Schulten 91]. It starts with two nodes at random positions, and as in GCS, new nodes are inserted successively to support the node with high accumulated errors. Unlike the GCS, the GNG structure is not constrained. The nodes are connected by edges with a certain age. Once the age of an edge exceeds a threshold, it will be deleted. After a fixed number of iterations, a new node is added between the node with the highest accumulated error and the one with maximum accumulated error among all its neighbors.

The GG, as an alternative form of growing network, starts with 2×2 nodes, taking advantages of a rectangular structured map. The model adds rows and columns of neurons during the training process, and therefore is able to automatically determine the height/width ratio suitable for the data structure. The heuristics used to add and remove nodes and connections are the same as those used in GCS.

2.6.3 Incremental Grid Growing

Another approach is the Incremental Grid Growing (IGG) [Blackmore and Miikkulainen 93]. Starting from a small number of initial nodes, the IGG generates new nodes only at the boundary of the map. This guarantees that the IGG network will always maintain a two-dimensional structure, which results in easy visualization. Another feature

of IGG is that connections between neighboring map units may be added and removed according to a threshold value of the inter-unit weight differences. This may result in several disconnected subnetworks, which represent different clusters of input patterns. The Growing Self-Organizing Maps (GSOM) [Alahakoon et al. 00], in similar spirit as IGG, introduces a spread factor to control the growing process of the map.

2.6.4 Other Growing Structure Models

Other modified models have also been proposed, including the Plastic Self Organizing Maps (PSOM) [Lang and Warwick 02], the Grow When Required (GWR) [Marsland et al. 02], and etc. Figure II.4 shows the simulation results of the original SOM and some of the dynamic models discussed above, which are given in [Fritzke 06]. The simulation results are achieved using 40,000 input signals from a probability distribution, which is uniform in the shaded area. The growing versions of the SOM aim at achieving an equal distribution of the input patterns across the map by adding new nodes near the nodes that represent an unproportionally high number of input data.

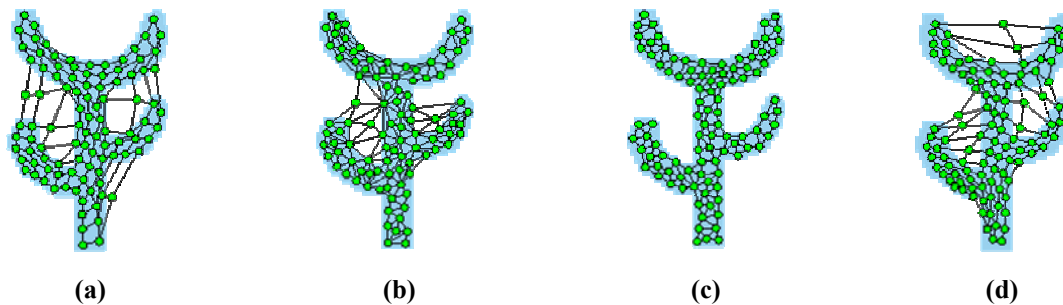


Figure II.4 Simulation results of different models: (a) the SOM, (b) GCS, (c) GNG, and (d) GG. The distribution is uniform in the shaded area. Map units are denoted by circles.

2.6.5 Hierarchical Models

Beside the limitation of the fixed structure, another deficiency of the classic SOM is the inability of capturing the hierarchical structure commonly present in real-world data. The structural complexity of such data sets is usually lost during the mapping process by means of a single, low-dimensional map. In order to handle a data set with hierarchical relationships, hierarchical models should be used. These models try to organize data in a hierarchy by displaying a representation of the entire dataset at a top level in a coarse granularity and allowing the lower levels to reveal the internal structure of each cluster found in a higher-level representation, where such information might not be so apparent [Vicente and Vellido 04].

The hierarchical feature map [Miikkulainen 90] uses a hierarchical setup of multiple layers, where each layer is composed of a number of independent SOMs. Starting with one initial SOM at the top layer, a separate SOM is added to the next layer of the hierarchy for every unit in the current layer. Each map is trained with only a portion of the input data that is mapped onto the respective unit in the higher layer map. The amount of training data for a particular SOM is reduced as the hierarchy is traversed downwards. As a result, the hierarchical feature map requires a substantially shorter training time than the basic SOM for the same data set. Moreover, it may be used to produce fairly isolated, or disjoint, clusters of the input data, while the basic SOM is incapable of performing the same [Merkl and Rauber 98].

Another hierarchical model, the hierarchical self-organizing map (HSOM) [Lampinen and Oja 92], focuses on speeding up the computation during winner selection by using a pyramidal organization of maps. However, like the hierarchical feature map,

while representing the data in a hierarchical way this model does not provide a hierarchical decomposition of the input space.

In addition, an extension to the hierarchical SOM models and the Growing Grid, the Growing Hierarchical Self-Organizing Map (GHSOM) was introduced in [Rauber et al. 02]. This model builds a hierarchy of multiple layers, where each layer consists of several independent growing SOMs. Starting from a top-level SOM, each map grows incrementally to represent data at a certain level of detail in a manner similar to the GG. In GHSOM, the level of detail is measured in terms of the overall quantization error. For every map unit in a level, a new SOM might be added to a subsequent layer if this unit represents input data that are too diverse and thus more details are desirable for the respective data.

Once the training process is over, visual display of the map must be carried out in order for the underlying structure of data to be perceived. A variety of visualization techniques based on the SOM have been developed, which will be reviewed in the next chapter.

CHAPTER III

VISUALIZING SELF-ORGANIZING MAPS

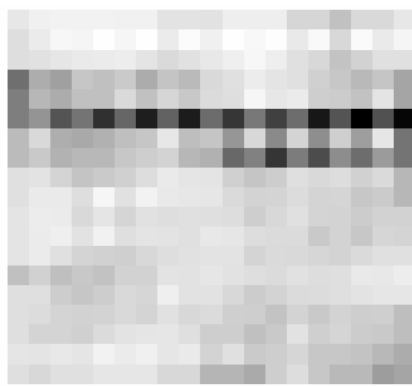
The visualization potentiality is a key reason to apply the SOM for data analysis. Once the learning phase is over, visual display of the map can be carried out in order for the underlying structure of the data to be observed. Extracting the visual information provided by the SOM is one of the primary motivations for this study.

The visualization of SOMs is motivated by the fact that a SOM achieves a nonlinear projection of the input distribution through a commonly two-dimensional grid. This projection can be visualized in different ways by a variety of techniques. Some of them visualize the input vectors directly, whereas others take only the prototype vectors (or weight vectors) into account. Based on the object to be visualized, these techniques can be divided into several categories, which are reviewed in the remainder of this chapter.

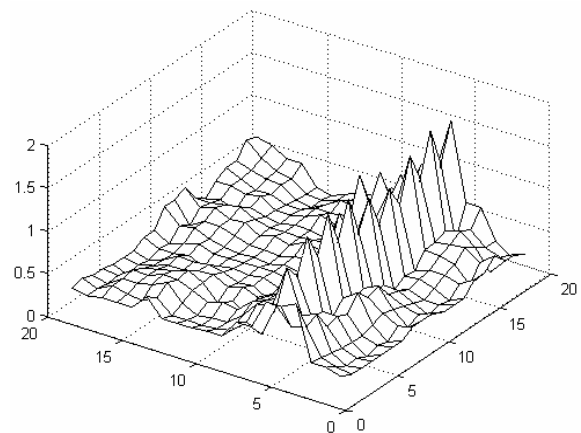
3.1 Visualizing Map Topology

One category of the visualization techniques is to visualize the SOM topology through distance matrices. The most widely used method in this category is the unified

distance matrix (U-matrix) [Ultsch and Siemon 90], which enables visualization of the topological relations between the neurons in a trained SOM. The idea is to show the underlying data structure by graphically displaying the inter-neuron distances between neighboring units in the network. The distances of the prototype vector of each map unit to its immediate neighbors are calculated and form a matrix. The same metric is used to compute the distances between map units, as is used during the SOM training to find the BMU. By displaying the values in the matrix as a 3D landscape or a gray-level image, the relative distances between adjacent units on the whole map become visible. The U-matrix is calculated in the prototype space and displayed in the map space.



(a)



(b)

Figure III.1 U-matrix presentations of a 10×10 rectangular SOM: (a) a gray-level image and (b) a 3D plot. The Iris data set is used to train the SOM.

A simplified approach is to calculate a single value for each map unit, such as the maximum or the sum of the distances to all immediate neighbors, and use it to control the height or color in the U-matrix representation [Kraaijveld et al. 95]. High values in the U-matrix encode dissimilarity between neighboring units. Consequently they correspond to cluster boundaries and are marked by mountains in a 3D landscape or dark shades of gray

in a coloring scheme. Low values correspond to similarity between neighboring units, represented by valleys or light shades of gray.

A demonstration of the U-matrix is presented in Figure III.1, which is based on a 10 by 10 rectangular SOM. The Iris data set [Fisher 36] is used to train the SOM in this example. The Iris data set contains 150 data points from 3 classes, the *setosa* class and two linearly inseparable classes, *versicolor* and *virginica*. In Figure III.1, two essential clusters can be observed from both the gray-level presentation and the 3D landscape presentation. Apparently, the two linearly inseparable classes are not separated in the projection space.

3.2 Visualizing Data Density

Recently a density-based visualization technique, the P-matrix [Ultsch 03a, Ultsch 03b], has been introduced, which estimates the data density in the input space sampled at the prototype vectors. The P-matrix is defined analogously to a U-matrix. Instead of local distances, this technique uses density values in data space measured at the position of each prototype vector as height values, called P-heights. The estimate of the data density is constructed using Pareto Density Estimate (PDE) [Ultsch 03a], which calculates the density as the number of input data points inside a hypersphere (Pareto sphere) within a certain radius (Pareto radius) around each prototype vector. In contrast to the U-matrix, neurons with large P-heights are located in dense regions of the data space, while those with small P-heights are in sparse regions. Illustrations of different visualizations of a SOM are given in Figure III.2, taken from [Ultsch and Mörchen 05]. Figure III.2(c) shows the P-matrix of the Gaussian mixture data set in Figure III.2(a), where darker gray

shades correspond to larger densities. Compared to the U-matrix presentation shown in Figure III.2(b), the P-matrix gives a complementary view of the same data set.

A combination of the U-matrix and the P-matrix has also been proposed by Ultsch, namely the U*-matrix [Ultsch 03b]. Commonly viewed as an extension to the U-matrix, it takes both the prototype vectors and the data vectors into account. The values of the U-matrix are dampened in highly dense regions, unchanged in regions of average density, and emphasized in sparsely populated regions. It is designed for use with Emergent SOMs [Ultsch and Mörchen 05], which are SOMs trained with a high number of map units compared to the number of data samples. U*-matrix is advantageous over the U-matrix in data sets with clusters that are not clearly separated. The U*-matrix presentation of the Gaussian mixture data set in Figure III.2(d) shows clearly two Gaussian distributions, which U-matrix fails to reveal.

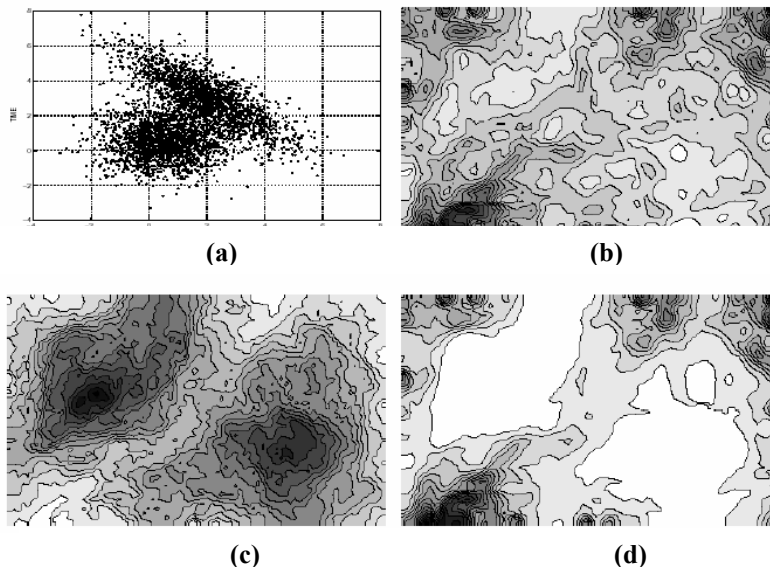


Figure III.2 Different visualizations of the SOM: (a) The original data set of a mixture of two Gaussians, (b) U-matrix presentation, (c) P-matrix presentation, (d) U*-matrix presentation.

3.3 Visualizing Prototype Vectors

An alternative way to visualize the SOM is to project the prototype vectors onto a two-dimension output space using a generic projection method. Such methods include multidimensional scaling (MDS) [Davison 83] and Sammon's mapping [Sammon 69]. MDS is a traditional technique for transforming a dataset from a high-dimensional space to a space with lower dimensionality. It creates a mapping to a usually two-dimensional coordinate space, where object can be represented as points. The inter-point distances in the original data space are approximated by the inter-point distances of the projected points in the projected space. Accordingly, more similar objects have representative points that are spatially nearer to each other. The error function to be minimized can be written as

$$E = \frac{\sum_{i \neq j} (d_{ij} - d_{ij}^*)^2}{\sum_{i \neq j} (d_{ij}^*)^2}, \quad (\text{III.1})$$

where d_{ij} denotes the distance between vectors i and j in the original space, and d_{ij}^* denotes the distance between i and j in the projected space. A gradient method is commonly used to optimize the above objective function. MDS methods are often computationally expensive.

Closely related to MDS, Sammon's mapping also aims at minimizing an error measure that describes how well the pairwise distances in a data set are preserved [Kaski 97]. The error function of Sammon's mapping is

$$E = \frac{1}{\sum_{i \neq j} d_{ij}} \sum_{i \neq j} \frac{(d_{ij} - d_{ij}^*)^2}{d_{ij}}. \quad (\text{III.2})$$

Compared to MDS, the local distances in the original space are emphasized in Sammon's mapping. Since the mapping employs steepest descent procedure to minimize the error, it requires both first- and second-order derivatives of the objective function at each iteration [Ridder and Duin 97]. The computational complexity, as a result, is even higher than MDS.

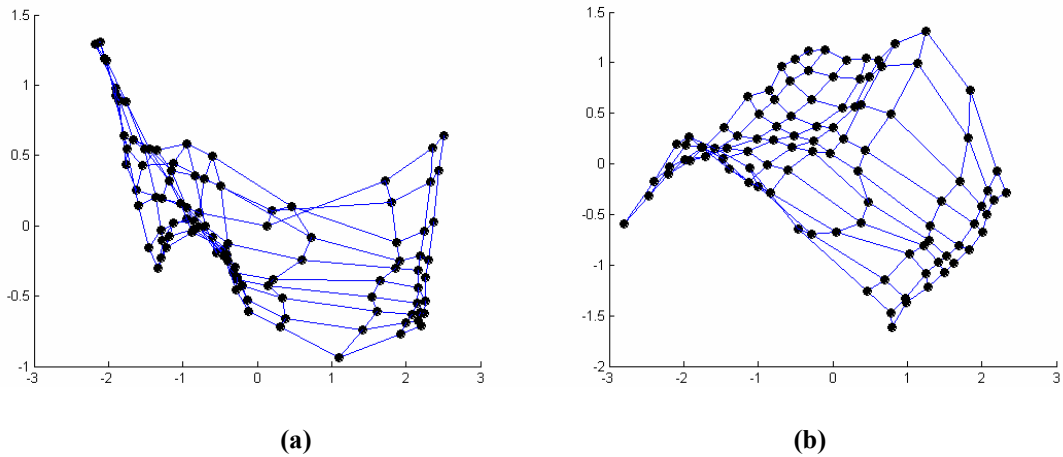


Figure III.3 Different ways to visualize the prototype vectors: (a) MDS projection of a SOM, (b) Sammon's mapping of a SOM. Neighboring map units, depicted as black dots, are connected to each other. The Iris data set is used to train the SOM.

Since the SOM provides a topology-preserving mapping of the input data, the MDS or Sammon's projection of the SOM can be used as a rough approximation of the shape of the input data. Both of these nonlinear projection approaches are iterative and computationally expensive. However, the computation load can be alleviated to an acceptable level when applied to the prototype vectors of a SOM instead of the original data set, provided a much smaller number of map units are used compared to the input vector number. The MDS projection and Sammon's mapping of a SOM are illustrated in Figure III.3, where the map units are visualized as black dots and connected to their neighbors by lines. In this example, the Iris data set is used to train a 10×10 rectangular

SOM. Roughly two clusters can be seen from both projections. Apparently, the two linearly inseparable classes, *versicolor* and *virginica*, are still joined in the projection space.

In addition to the high computational cost, another drawback of MDS and Sammon's mapping is that they do not yield a mathematical or algorithmic mapping procedure for previously unseen data points [Ridder and Duin 97]. That is, for any new input data point to be accounted for, the whole mapping procedure has to be repeated based on all available data. Mao and Jain have proposed a feed-forward neural network [Mao and Jain 95] to solve this problem, which employs a specialized, unsupervised learning rule to learn Sammon's mapping.

3.4 Visualizing Component Planes

The prototype vectors can also be visualized using the component plane representation. Instead of a single plot, this technique provides a "sliced" version of the SOM, which shows the projection of each individual dimension of the prototype vectors on a separate plane [Simula et al. 98]. The values of each component are taken from all prototype vectors and depicted by color coding. Each component plane shows the distribution of one prototype vector component. Similar patterns in different component planes indicate correlations between the corresponding vector components. This technique is hence useful when the correlation between different data features is of interest. However, one drawback of component planes is that cluster borders cannot be easily perceived. In addition, data with high dimensionality results in lots of plots.

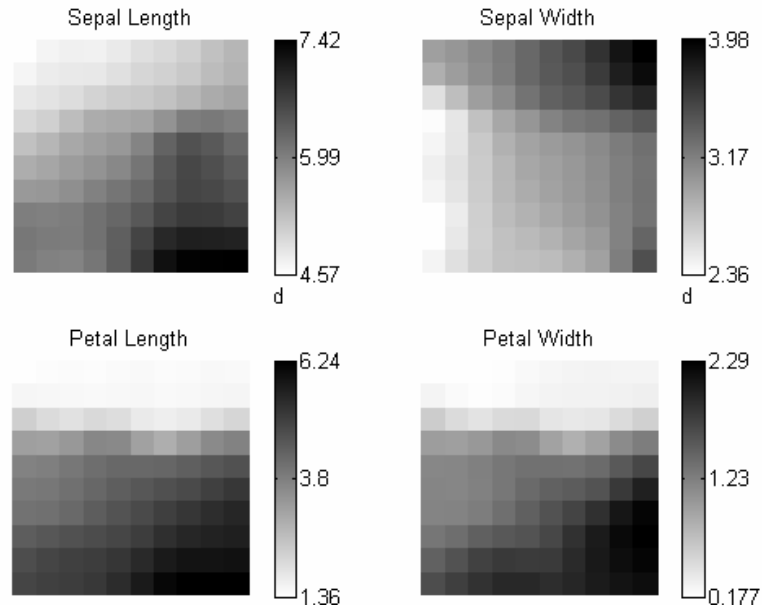


Figure III.4 Component planes representation of a SOM trained with the Iris data set. The color bars beside each component planes show the maximum, mean, and minimum values and the corresponding colors.

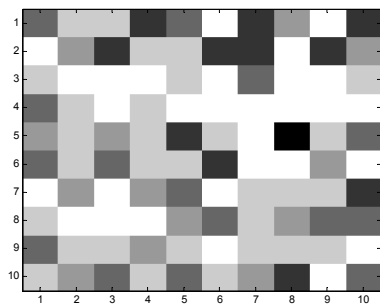
The component planes of the 10×10 rectangular SOM trained with the Iris data set is presented in Figure III.4. The color scheme of the map units has been set so that the lighter the color is, the smaller the component value of the corresponding prototype vector is. It can be seen, for instance, that the two components, *petal length* and *petal width*, are highly related.

3.5 Visualizing Best Matching Units

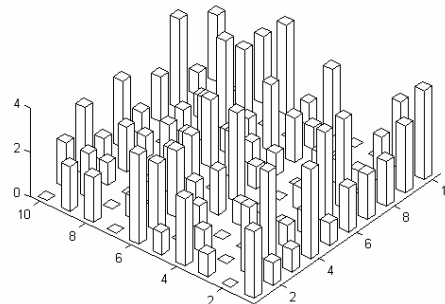
Another category of visualization is to display the BMUs of the input data set. Data vectors can be projected on the map by locating their BMUs. Because the prototype vectors are ordered on the map grid, nearby map units will have similar data projected to them. Projecting multiple input vectors will result in a histogram of the BMUs. For each data vector, the BMU is determined and the number of hits for that map unit is increased

by one. The hit histogram shows the distribution of the data set on the map. Map units on cluster borders often have very few data samples, which implies very few hits in the histogram. Therefore low-hit units can be used to indicate cluster borders. The values of a histogram can be depicted in different ways. Figure III.5 illustrates the gray level presentation and the 3D presentation of the same hit histogram. In Figure III.5(a), the darker the gray shade is, the higher the hit value of that unit is. In Figure III.5(b), the height directly corresponds to the value of the histogram.

However, hit histograms consider only the BMU for each data sample while real world data is usually represented by more than one unit. This inevitably causes distortions in the final map. A variation of the standard hit histogram, namely the Smoothed Data Histogram (SDH) [Pampalk et al. 02], has been developed counting the data sample's relativeness to more than one map unit. The SDH allows a data sample to "vote" not only for the BMU but also for the next few good matches based on the ranks of distances between the data sample and the corresponding prototype vectors.



(a)



(b)

Figure III.5 Different presentations of the hit histogram: (a) a gray level image, and (b) a 3D plot. The Iris data set is used to train the SOM.

3.6 Other Visualizations

Aside from the above categories, other visualization techniques are also available for the SOM. A rather different way to project the prototype vectors, the so-called Adaptive Coordinates [Merkl and Rauber 97], was proposed with a focus on cluster boundary detection. This approach mirrors the movements of prototype vectors during the SOM training within a two-dimensional “virtual” space, which is used for subsequent visualization of the clustering result. The initial positions of the prototype vectors are defined by the network structure, which are on top of the junctions of the map grid. The coordinates of the prototype vectors are adapted during the training. After convergence of the training process, the prototype vectors can be plotted in arbitrary positions in the projected space according to their coordinates. The algorithm offers an extension to both the basic training process and the fixed grid representation.

Another extended SOM model, called the visualization-induced SOM (ViSOM) [Yin 02], has been developed to directly preserve the distance information along with the topology on the map. The ViSOM updates the weight of the winning neuron using the same learning rule as the SOM. For the neighboring neurons, the weight adaptation is decomposed into two parts: a lateral movement toward the winner and an updating movement from the winner to the input vector. ViSOM places a constraint on the lateral contraction force between the neurons and hence regularizes the inter-neuron distances. As a result, the inter-neuron distances in the data space are in proportion to those in the map space. A scalable parameter λ is introduced in the constraint that controls the

resolution of the map. If a high resolution is desirable, a small λ should be used, which will result in a large map.

CHAPTER IV

VISUALIZATION OF A SCIENTIFIC DOCUMENT COLLECTION

4.1 Visualizing Documents in General

Technological innovation has led to a rapid growth in the quantity of textual information. Such massive information can be overwhelming to the users. Powerful tools for exploring and organizing this wealth of information are critically needed. A natural and useful heuristic to assist the user in document exploration is to arrange textual items, or documents, in space, as spatial relations play an important role in human recognition and communication. Such spatial representations are called maps. The document map provides a mapping from the document space to a 2D or 3D display space. Documents and their relations are represented by various visual cues such as points, links, clusters and areas as well as their sizes, colors and geometric arrangements. Therefore, the map display visualizes the documents and the relationships that might exist among them, which may be invisible to the users. The ability to visualize large text data sets enables users to quickly gain insight into a collection of documents.

It is an effective way to cluster documents using the SOM, which generates maps that visualize the similarity between documents in term of distances within the two-

dimensional display space. Hence, similar documents may be found in neighboring regions of the map and clusters with similar concepts may be located nearby on the map. This provides the SOM clustering a main advantage over the traditional statistical clustering tools, which only assign objects to clusters but are unable to reveal the relations between clusters. This chapter presents a novel visualization method based on the SOM algorithm. This method organizes a document collection on a map display, which provides an overview of the collection and sheds light on the associations that might exist among the documents.

4.2 Document Encoding

4.2.1 Inter-Document Links

A pre-process is needed to encode documents into a set of feature vectors before any automatic analysis tools can be applied. There are two types of inter-document relations existing in any document corpus, the implicit links and the explicit links [Olsen et al. 93], based on which the documents can be converted into the vector representation.

The implicit inter-document links are relations between document contents, which exist on the basis of the term-document relations. All documents consist of collections of terms and individual terms can appear in multiple documents. A set of terms, or keywords, can be selected and used as descriptors of the represented document. Links between documents can be established if different documents are described by the same terms. In this case, terms correspond to the features of the documents. The implicit links

are illustrated in Figure IV.1. The network in the figure describes the importance of a term with respect to a document.

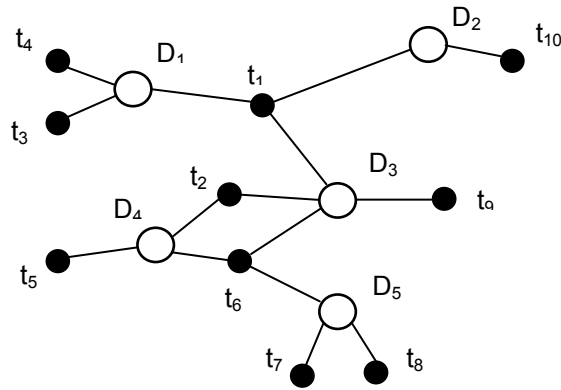


Figure IV.1 The implicit links between documents based on the document-term relations. Documents D_i (circles) are connected by common terms t_i (black dots).

The explicit links within a document collection are pointers into other documents, usually created by the author. The types of explicit links in a document may vary depending on different document categories. In the scientific publications, for instance, authors, organizations, and citations can be used as the explicit links. A scientific document usually contains a list of authors and their affiliated organizations, while individual authors and organizations can appear in multiple documents. Inter-document links can be defined by finding papers with the same authors and organizations. Besides, explicit links are built between an author's current work and prior work of others when the author creates the reference list for his or her publication.

Different sets of features will be selected to describe the documents if different types of links are used. One popular choice is to use the term-document representation so that the features correspond to terms or keywords. One well-know example is the WEBSOM system [Kohonen et al. 00]. However, the term-document representation is only one aspect of the possible semantic associations between documents. Technology

advances that make bibliographic databases easily available also enable us to represent documents in different ways, e.g. using title lists, author lists, and citations. In the following sections, several models that attempt to describe the documents using different sets of features are reviewed.

4.2.2 The Vector-Space Model (VSM)

The VSM [Salton et al. 75] is one class of techniques that are still popularly used in currently available information retrieval systems. It relies on the premise that the meaning of a document can be derived from the document's constituent terms. The VSM represents documents as vectors of terms in a t -dimensional space so that each unique term in the document collection corresponds to a dimension in the space. Each document i is described by a t -dimensional vector $D_i = (d_{i1}, \dots, d_{ij}, \dots, d_{it})^T$, where d_{ij} represents the weight of the j th term in the document i . A 2-dimensional matrix of documents and terms can therefore represent a collection of documents. Document similarities can be computed by simple vector operations.

The simplest way of document encoding using the VSM is to represent document as binary vectors, where the value of a vector element is set to 1 if the respected term is found in the document and to 0 otherwise. More complex types of vector space models involve different methods to compute the elements in D_i in an attempt to provide a better measure of each term's importance in the document. Term weighting schemes are typically used to serve for this purpose. A straightforward approach known as term-frequency weighting is to assign the weight to each term k in accordance with its frequency of occurrence, f_i^k , in a document i . A more sophisticated way is to use the well-

know inverse document frequency (IDF) based schemes, which assumes the less frequent terms are higher in value than those of the more frequent terms. Thus the weight of a term k is computed by multiplying the standard term frequency f_i^k with the inverse document frequency IDF_k [Salton et al., 75], given as

$$W_k = f_i^k \times IDF_k. \quad (IV.1)$$

IDF_k is the inverse of the number of documents in which the term occurs, which can be defined as:

$$IDF_k = \log \frac{N}{d_k}, \quad (IV.2)$$

where N is the total number of documents in the collection and d_k is the number of documents containing term k .

The VSM, basing its rankings on the Euclidean distance or the angle measure between document vectors, is able to automatically find documents that might be conceptually similar. The main problem of the VSM is the large vocabulary in any sizable document collection, which results in a vast dimensionality of the document vectors [Kohonen et al. 00].

4.2.3 Latent Semantic Indexing (LSI)

Latent Semantic Indexing (LSI) [Deerwester et al. 90] is a well-established approach for the investigation of conceptual relations in documents. It is a parallel, yet similar, approach to the VSM. Because of the way it represents terms and documents in a space, it is often considered a vector-space model.

The VSM and some other information retrieval systems suffer from two language related problems due to the facts that terms often have more than one specific meaning

(synonymy) and multiple terms may also describe the same concept (polysemy) [Deerwester et al. 90]. Because of the first problem, it is often difficult to discriminate between two documents that share a given word, but use it differently, without understanding the context in which the word was used. As a result of the second problem, related documents may not use the same terminology to describe their shared concepts.

By using the singular value decomposition (SVD) [Golub and Van Loan 89] on the term-document matrix, LSI attempts to solve the synonymy and polysemy problems. The approach is based on the concept that the principal components of the term-document space expose an underlying latent semantic structure in the data. The SVD is used to reduce the dimensionality of the term-document space by selecting the highest singular values, where the most of the variance of the original space is located. The principal components of the space, the major associative patterns, are extracted from the data and the smaller, less important patterns are ignored. As a result, semantically-related documents may still be placed near each other in the space even though they may not share terms, if these documents have similar term-usage patterns.

In this reduced-space model, no attempt is made to interpret the meaning of each dimension. Each dimension is merely assumed to represent one or more semantic relationships in the term-document space. One advantageous effect of LSI is that the dimensionality of the document vector becomes much smaller, which makes it possible for LSI to represent large data sets.

4.2.4 Citation Based Models

In the previous sections of this chapter, we have discussed document visualization and encoding in general. For scientific documents, a distinct property, which makes them different from other textual items, is that they contain bibliographic references, or citations. Citations serve the purpose of pointing to source documents the author referred to or consulted in the production of a document, thus establishing links between documents from the author's point of view. Our approach to encode documents is based on the analysis of their citations. Citation analysis, which has been used extensively in library and information science, is the study that uses citations in scholarly works to establish links between authors, between scholarly works, between journals, or between author institutions [Osareh 96]. Citations both from and to a certain document may be the objects of the study.

Citation-based document analysis has been limited by the need of manually extracting citation information from documents. The recent availability of Internet-based citation services and databases, allowing easy access to document citations in electronic form, has made the application of citation-based studies quite appealing.

4.2.4.1 Direct and Indirect Citation Links

A useful model of representing a collection of scientific documents as a network of nodes connected by citations was introduced in [Small 99]. The document-citation network is illustrated in Figure IV.2, where the documents are displayed as circles and the citation links as lines connecting them. Both direct and indirect citation links exist in the document-citation network.

A direct citation link occurs when one document cites another document. The following similarity function can be defined to depict the direct citation links between documents:

$$dc_{ij} = \begin{cases} 1 & \text{if } D_i \text{ cites } D_j \text{ or } D_j \text{ cites } D_i \\ 0 & \text{otherwise} \end{cases} \quad (\text{IV.3})$$

The similarity values produced by this function form the elements of a symmetric binary similarity matrix.

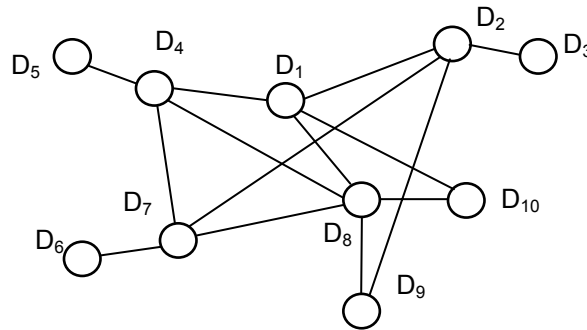


Figure IV.2 Illustration of citation links

Beside the direct citation link, two documents can be connected indirectly by taking two steps in the network in the following three forms [Small 97]:

- Bibliographic coupling, which occurs when a pair of documents cite a common third document;
- Co-citation, which occurs if a pair of documents are cited by a common third document;
- Longitudinal coupling, which occurs between a pair of documents when one document cites a third document that cites the other document of the pair.

The four types of citation links are shown in Figure IV.3.

Citation based similarity measures are used extensively when dealing with journal articles and patents from sources that provide citation data such as the Science Citation Index. Inter-document similarities can be calculated using a single citation link or the combination of multiple citation links, which form a similarity matrix. Each dimension of the similarity matrix corresponds to one document in the collection and the value of each element represents the relative strength of the citation relationship between the corresponding document pair.

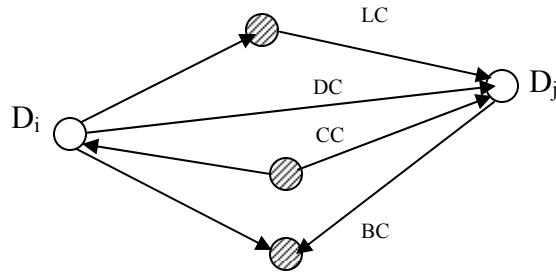


Figure IV.3 Illustration of four types of citation links between a pair of documents. A pair of documents D_i and D_j are connected by a direct citation (DC) and three forms of indirect citation links: bibliographic coupling (BC), co-citation (CC), and longitudinal coupling (LC). The shaded circles represent the corresponding third documents in the indirect links.

4.2.4.2 Clustering Documents Using Bibliographic Coupling

Based on the different formations of the similarity matrix, various citation analysis techniques have been developed with a focus on different aspects of the intellectual structure of the document data set [Small 97, Morris et al. 03]. Clustering scientific documents using bibliographic coupling was introduced in [Kessler 63], where bibliographic coupling was defined as the number of references cited by both documents. It is assumed that if two documents cite many common references, they probably cover similar research topics. Based on this assumption, bibliographic coupling is often used to

cluster documents into research fronts, that is, groups of papers that cover the same research topic [Morris et al. 03].

In our approach, bibliographic coupling counts are used to describe the inter-document relationships, based on which a similarity matrix is built to store pair-wise similarity values between documents. The rows, or columns, of the similarity matrix are vectors corresponding to individual documents. Documents are subsequently clustered using the similarity matrix. The details of the derivation of the similarity matrix are given in Chapter V.

CHAPTER V

RANKED CENTROID PROJECTION

Several challenges remain when using the SOM for visualizing document databases. First, the shape of the grid and the number of nodes have to be predetermined. This requires prior knowledge of the input data characteristics, which is usually unavailable before the analysis. Second, the underlying hierarchical relations can hardly be detected by a single map. Such relations are commonly observed in document collections and thus their proper identification is highly desirable. A further limitation, which occurs when using the SOM projection, is that the map resolution depends solely on the size of the map. To have a high-resolution document map, which is desirable in most cases, it requires a considerably large number of neurons. To achieve a better visualization, a high-resolution SOM may even call for a higher number of neurons than that of input vectors [Bienfait 94]. As a result, the size of the SOM will become impractically huge when dealing with large data sets. The computational complexity grows quadratically with the number of neurons [Vesanto and Alhoniemi 00]. As a result, training huge maps may be exceedingly time-consuming.

To resolve the above limitations, a SOM-based visualization approach has been developed. Figure V.1 shows the schematic diagram of the proposed approach. In the first step, a similarity matrix is derived from the collection of documents of interest. This

process will be further discussed in this chapter. The similarity matrix is then used to train a Growing Hierarchical Self-Organizing Map (GHSOM) [Rauber et al. 02], which clusters document items in a hierarchical manner and at the mean time allows for adaptation of the network architecture during training. Following the training of the GHSOM, a novel SOM projection technique, namely the Ranked Centroid Projection, is used to project the input vectors to a hierarchy of two-dimensional output maps. Using the proposed approach, a high-resolution map can be achieved with comparatively low computational cost.

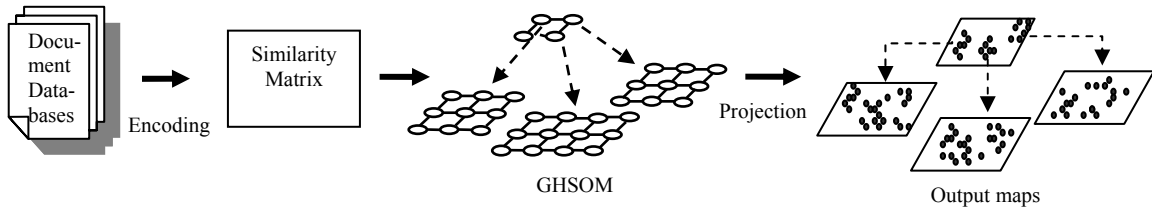


Figure V.1 The schematic diagram of the proposed SOM-based approach

5.1 The GHSOM Architecture

The typical goal of document clustering is to discover subsets of large document collections that correspond to individual topics. Additionally, it can be applied hierarchically, yielding more refined groups within clusters. This leads to a large-to-small-scale presentation of the conceptual structure of the document collection, in which large scale clusters correspond to more general topics and smaller scale ones correspond to more specific topics within the general topics. Cluster hierarchies thus serve as topic hierarchies [Noel et al. 02]. In order to detect this hierarchical structure, the GHSOM is employed in the proposed approach.

The GHSOM combines the advantages of two principal extensions of the Self-Organizing Map, the dynamic growth and the hierarchical structure. It uses an adaptive architecture which grows during its unsupervised training process to uncover the hierarchical structure of the data set under analysis.

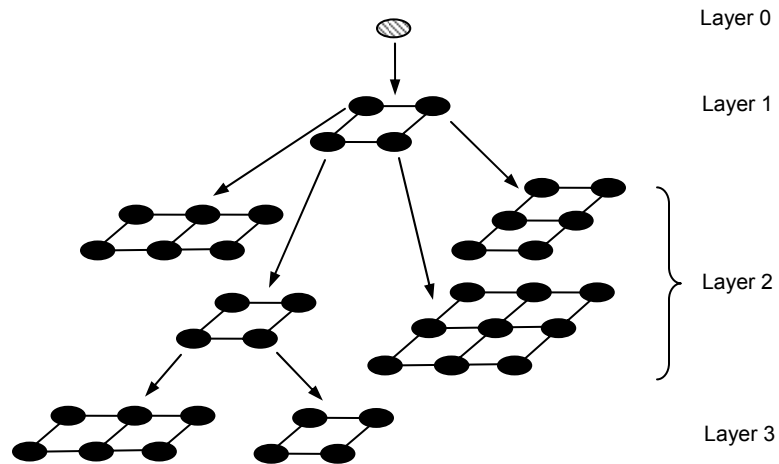


Figure V.2 Graphic representation of a trained GHSOM

As depicted in Figure V.2, the GHSOM evolves to a multi-layered architecture composed of independent growing self-organizing maps. At layer 0, a single-unit SOM serves as a representation of the complete data set. Only one map is used at the first layer of the hierarchy, which initially consists of a small number of units, usually a grid of 2×2 units. For every unit in this map, a separate SOM can be added to the second layer. This process is repeated with any subsequent layers in the hierarchy. In the GHSOM shown in Figure V.2, two units from one of the second-layer maps have been further expanded into third-layer maps. The maps in the upper layers show a coarse representation of the data, revealing the major clusters, whereas those in the lower layers offer a more detailed view of the data.

This model grows in two dimensions: in width (by increasing the size of each SOM) and in depth (by increasing the number of levels in the hierarchy). For growing in width, each SOM attempts to modify its layout and increase its size in a systematical way similar to the Growing Grid model, so as to represent the data at a specific level of granularity. The basic steps of the growth in width are summarized in Table V.1.

Table V.1 Basic steps of the growth in width

1.	Initialize the weight of each unit with random values. Reset error variables E_i for every unit i .
2.	The standard SOM training algorithm is applied.
3.	For every input vector, the quantization error (qe) of the corresponding winner is measured in terms of the deviation between its weight vector and the input vector. Update the winner's error variable by adding the qe to E_i .
4.	After a fixed number λ of training iterations, identify the error unit e with the highest E_i .
5.	Insert a row or a column between the error unit e and its most dissimilar neighboring unit d in terms of the distance between respective weight vectors.
6.	Repeat steps 2-5 until the whole map's mean quantization error (MQE_m) reaches a given threshold so that $MQE_m < \tau_1 \cdot qe_u$ is satisfied, where qe_u is the quantization error of the corresponding unit u in the proceeding layer of the hierarchy and τ_1 is a fixed percentage.

Figure V.3 is a graphic representation of the growing process, which illustrates the insertion of a row (see Figure V.3(a)) or a column (see Figure V.3(b)).

As for growing in depth, the general idea is to form a new map in the subsequent layer for the units representing a set of input vectors that are too diverse. The basic steps for the growth in depth are summarized in Table V.2.

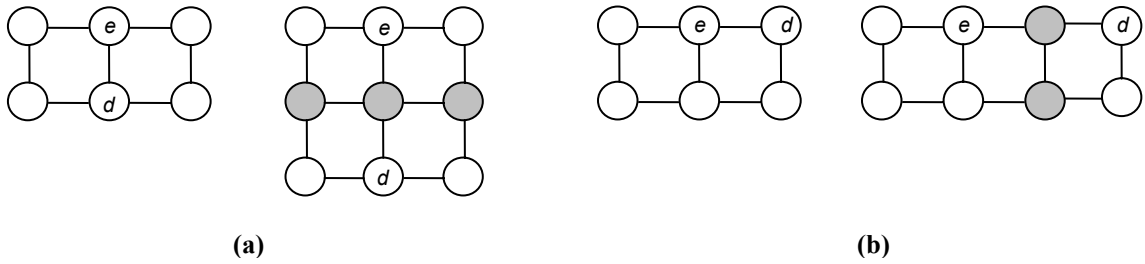


Figure V.3 Illustration of the growing process: (a) A row or (b) a column of units is inserted in a SOM [Rauber et al. 02]. Unit e is the error unit and d is its most dissimilar neighbor. The shaded circles denote the newly inserted units.

Table V.2 Basic steps of the growth in depth

1.	When the training of a map is finished, every unit is examined and those units fulfilling the criterion given as $qe_i > \tau_2 \cdot qe_0$ will be subject to a hierarchical expansion. qe_0 is the quantization error of the single unit in the layer 0.
2.	Train the newly added SOM with input vectors mapped to the unit map that has just been expanded.

GHSOM automatically determines the architecture of the SOMs at different levels. The sizes of the SOMs and the depth of the hierarchy are determined during the learning process according to the structure of the input data. Moreover, it enables users to choose the granularity of the hierarchical representation of the data by setting different parameters. The growing process of the GHSOM is guided by two parameters τ_1 and τ_2 . τ_1 specifies the desired quality of input data representation at the end of the training process while τ_2 specifies the desired level of detail that is to be shown in a particular SOM. The smaller τ_1 is, the larger the emerging maps will be. Conversely, the larger τ_2 is, the deeper the hierarchy will be.

5.2 Projection Method

The proposed Ranked Centroid Projection (RCP) approach is based on the standard SOM architecture and learning procedure [Morris et al. 01, Wu and Yen 03]. It can also be applied to individual maps in a GHSOM. This approach projects the input vectors onto the two-dimensional SOM grid and the resulting topographic map indicates the similarities between input vectors and various prototype vectors in terms of the distance between the respective units.

As discussed in Chapter I, the SOM is a special data mining tool that can be used at the same time for both clustering and projecting data. During the SOM training process, a set of prototype vectors, which is much lesser than the number of data vectors, becomes ordered along a two-dimensional “elastic network” that follows the distribution of the data. By adapting the prototype vectors of the winner unit and a number of neighboring units to the input patterns iteratively, the similarities between the input patterns originally present in the n -dimensional data space are mirrored within the two-dimensional output space of the SOM. Therefore the SOM realizes a topology-preserving projection from the high-dimensional input space onto a low-dimensional grid of neurons. This ordered grid can be used as a convenient visualization surface for showing the cluster structure and other features of the data.

5.2.1 An Approach Using Weighted Averages

The prototype vectors of a trained SOM can be interpreted as cluster centers, while the coordinates (cx_i, cy_i) of each map unit i indicate the position of the

corresponding cluster center within the map grid. After convergence of the training process, for any data vector x_i in the input space, one or several of the prototype vectors are close to it. The similarities between the input vector and all the prototype vectors can be calculated in terms of the Euclidean distances between them. A similarity measure can be defined as the inverse of the Euclidean distance:

$$s_{ij} = d_{ij}^{-1} = \|x_i - w_j\|^{-1}, \quad (\text{V.1})$$

where s_{ij} is the similarity value and d_{ij} is the distance between input vector x_i and prototype vector w_j .

The objective of the proposed projection approach is to map input vectors onto the output space based on their similarities to the prototypes, which are inversely proportional to the Euclidean distances between the respective vectors as given in Equation (V.1). The map unit with the smallest distance to the input vector x_i is the BMU, which satisfies:

$$c = \arg \min_j \{ \|x_i - w_j\| \}, \quad (\text{V.2})$$

where c denotes the BMU. The BMU has the greatest similarity value with the input vector x_i and corresponds to a cluster that x_i is the most closely related to. Hence x_i should be projected to a position closer to the BMU than to the other units. However, in most cases, there are usually several units that have almost as good matches as the BMU. As a result, pointing out only the BMU does not provide sufficient information of the cluster membership of x_i , which is the problem with hit histograms. Intuitively, the data item should be projected to a position so that it is amidst a set of units that have the smallest distances to it, which correspond to its nearest neighbors in the input space.

In essence, the task of projecting input vectors can be treated as a problem of vector interpolation into a two-dimensional regular grid. Assume that the coordinates of each map unit, $Cw_i = (x_{w_i}, y_{w_i})$, are given. An interpolation function f is sought to assign coordinates to any data item in the input space, which can be represented as:

$$Cx_i = f(Cw_1, Cw_2, \dots, Cw_N), \quad (V.3)$$

where Cx_i represents the coordinates of the data sample x_i . This function should reflect the similarity relations between x_i and the map units.

A function that satisfies the above criteria is a weighted average of the positions of the map units, where the weighting is based on the distances between the data sample and those units. The inverse distance is used as a measure of the similarity. The coordinates of the input vector x_i can be calculated using the following function:

$$Cx_i = \begin{cases} \sum_{j=1}^N (d_{ij})^{-1} Cw_j / \sum_{j=1}^N (d_{ij})^{-1} & \text{if } d_{ij} \neq 0 \text{ for all } j \\ Cw_j & \text{if } d_{ij} = 0 \end{cases} \quad (V.4)$$

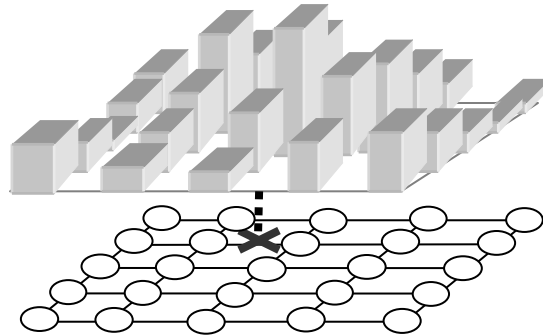


Figure V.4 Illustration of mapping an input vector by finding the centroid of the spatial histogram of the output values.

As given in Equation (V.1), d_{ij} is inversely proportional to the similarity between the data sample x_i and prototype vectors w_j . Therefore the weighting factors indicate the

normalized responses of x_i to various prototypes. Since the map units are arranged in a rectangular grid, the set of weights may be characterized as a two-dimensional histogram plotted across the map units as illustrated in Figure V.4. The SOM projection procedure continues with finding the centroid of this spatial histogram, where the data sample is then mapped.

5.2.2 Improving the Weighting Function by Applying a Ranking Scheme

To enhance the performance of the projection method, the basic weighting function discussed in the previous section is subject to modifications and correction terms. Instead of mapping the data sample directly onto the centroid of the spatial responses of all map units, a ranking scheme is applied to the weighting function. First a constant R is set to select only a number of prototypes that are nearby the input vector in the input space. Only the positions of the associated R units will affect the calculation of mapping position. R is in the range of one to the total number of the neurons in the SOM. A membership degree of a data sample to a specific cluster is then defined based on the rank of closeness between the data vector and the unit associated with that cluster, which is given by:

$$m_i = \begin{cases} \frac{R}{S} & \text{for the closest unit} \\ \frac{R-1}{S} & \text{for the 2nd closest unit} \\ \dots & \\ \frac{1}{S} & \text{for the } R\text{th closest unit} \\ 0 & \text{for all other units} \end{cases} \quad (\text{V.5})$$

where $S = \sum_{i=0}^{R-1} (R - i)$, ensuring a normalized membership.

For a data sample x_i , the new weighting function is defined by applying the membership degree m_i to function (V.4):

$$\mathbf{C}x_i = \begin{cases} \sum_{j=1}^N m_j (d_{ij})^{-1} \mathbf{C}w_j / \sum_{j=1}^N m_j (d_{ij})^{-1} & \text{if } d_{ij} \neq 0 \text{ for all } j \\ \mathbf{C}w_j & \text{if } d_{ij} = 0 \end{cases} \quad (\text{V.6})$$

With the new weighting function, the projection procedure continues with finding the centroid of the spatial response ranked by the corresponding membership degrees.

The effect of the ranking scheme is two-fold. Firstly, since only nearby units are significant in determining the projection of the data sample, eliminating calculations with distant prototypes can lessen the tendency that all data samples are biased toward the center of the map without impairing the projection quality [Morris et al. 01]. Secondly, the ranking scheme introduces a membership degree factor into the new weighting function in addition to the distance factor, which enables the proposed projection technique not only to reveal the clustering tendency in the data but also to visualize information on cluster memberships. A positive side effect of the ranking scheme is a considerable saving in computation as the result of selecting only R closest units. This saving becomes more significant when the map size is large.

5.2.3 Illustrations of the RCP

Unlike the hit histogram [Simula et al. 98], which simply maps a data sample onto its BMU, the RCP takes into account of several map units ranked by their closenesses to the data sample. Analogously, each map unit exerts an attractive force on the data item proportional to its similarity to that data item. The greater the force is, the closer the data

item will be drawn toward the map unit. The data item will end up being placed in a position where these forces reach an equilibrium state.

The number of the nearest units to be included in the calculation is determined by setting the parameter R . Different R will result in different projection results. In the following, a simple projection is illustrated with $R = 1, 2$ and 3 .

A set of SOM units in the input space is shown on the left side of Figure V.5. An input vector x_i is presented and marked by an \mathbf{X} . If R is set to 1, which is a winner-takes-all case, only the BMU is taken into account in computing the coordinates of x_i in the output space. The membership degree of x_i to each map unit is:

$$m_i = \begin{cases} 1 & i = 3 \\ 0 & i \neq 3 \end{cases} \quad (\text{V.7})$$

By applying Equation (V.5), the coordinates of x_i are:

$$\mathbf{C}x_i = \mathbf{C}w_c, \quad (\text{V.8})$$

where $\mathbf{C}w_c$ represents the coordinates of the BMU. The resulting mapping is illustrated on the right side of Figure V.5, where x_i is mapped directly onto its BMU. Mapping multiple data samples results in a hit histogram.

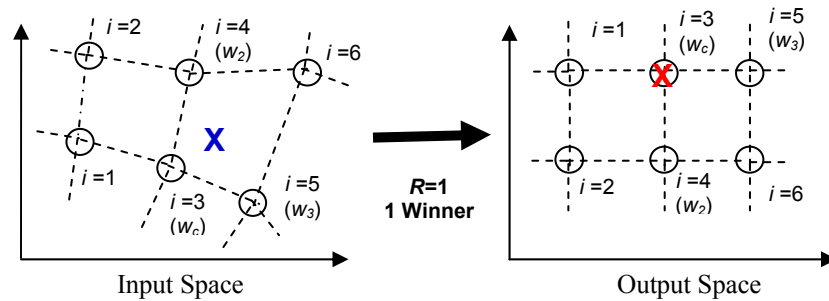


Figure V.5 Illustration of mapping an input vector x_i (marked by an \mathbf{X}) to its BMU. i is the index of each map unit. w_c denotes the BMU. w_2 and w_3 are the next two closest units.

If $R = 2$, the two closest units are taken into account in the calculation. As shown in Figure V.6, d_1 and d_2 are the Euclidean distances between the data sample and two winners w_c and w_2 . The membership degree of x_i to each map unit is:

$$m_i = \begin{cases} \frac{2}{3} & i = 3 \\ \frac{1}{3} & i = 4 \\ 0 & \text{otherwise} \end{cases} \quad (\text{V.9})$$

The coordinates of x_i are calculated as:

$$\mathbf{C}x_i = \frac{2}{3}(d_1)^{-1}\mathbf{C}w_c + \frac{1}{3}(d_2)^{-1}\mathbf{C}w_2, \quad (\text{V.10})$$

where $\mathbf{C}w_2$ are the coordinates of the second winner. The projection result is illustrated on the right side of Figure V.6.

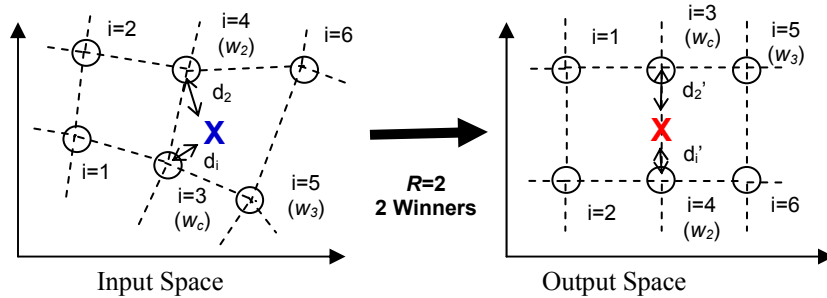


Figure V.6 Illustration of mapping an input vector when two units are considered. d_1 and d_2 are the distances in the input space. d_1' and d_2' are the distances in the output space.

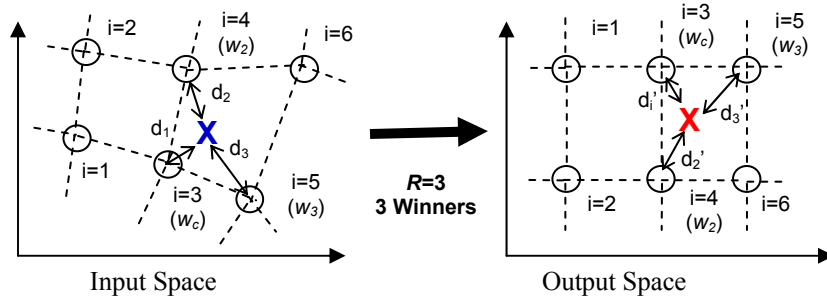


Figure V.7 Illustration of mapping an input vector when three winners are considered. d_1 , d_2 and d_3 are the distances in the input space. d_1' , d_2' and d_3' are the distances in the output space.

In the case of $R=3$, the coordinates of x_i are calculated as:

$$\mathbf{C}x_i = \frac{1}{2}(d_1)^{-1}\mathbf{C}w_c + \frac{1}{3}(d_2)^{-1}\mathbf{C}w_2 + \frac{1}{6}(d_3)^{-1}\mathbf{C}w_3, \quad (\text{V.11})$$

where $\mathbf{C}w_3$ are the coordinates of the third winner. This projection is illustrated in Figure V.7.

The computational complexity of the SOM increases linearly with the number of data samples, while the complexity scales quadratically with the number of map units. Because the RCP algorithm allows the data points to be projected to any locations across the SOM network, it can handle a large data set with a rather small map size and provide a high-resolution map at the mean time. Therefore, the presented procedure of mapping input vectors to the output grid using the RCP alleviates computational complexity considerably, making it possible to process large data sets.

5.2.4 Selecting the Ranking Parameter R

The weighting function in Equation (V.6) implies that only the nearby R map units are significant in computing the projection. The performance of the RCP depends heavily on the value of the ranking parameter R . It would be beneficial to determine the optimal R value automatically for each map based upon certain performance metrics.

5.2.4.1 Effect of R on the Projection Result

From the RCP illustrations in Section 5.2.3, we can see that different R values result in different mapping positions of the input vector. The effect of the ranking parameter R can be further illustrated in the following example.

A three-dimensional data set is shown in Figure V.8, which consists of 300 data points randomly drawn from three Gaussian sources. The mean vectors of the three

Gaussian sources are $[0,0,0]^T$, $[3,3,3]^T$, and $[9,0,0]^T$ respectively, while the variances are all 1. A SOM of 2×2 units is used to project the data points onto a two-dimensional space.

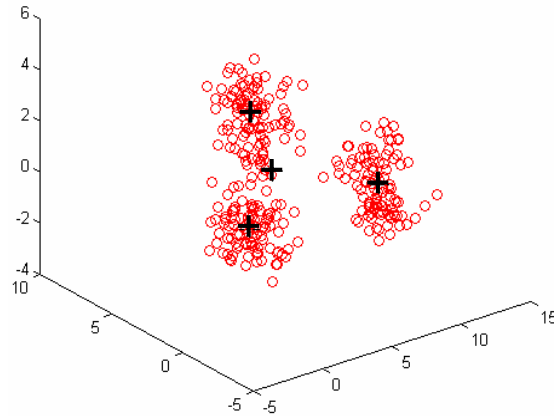


Figure V.8 Data set I: Samples in a three-dimensional data space marked as small circles and prototype vectors as plus signs.

After training, the prototype vectors of the SOM are shown as plus signs in Figure V.8, which span the input space with three of the map units representing the three cluster centers respectively. The input vectors are then projected using the RCP method. The projection results produced with different R values are presented in Figure V.9. The effect of R can be seen in this figure. For the case of $R=1$, where only the BMU is considered in the projection, the map is actually a hit histogram (a small random noise is added to the coordinates of each data point in order to show the volume of data points projected onto each map unit). Because it can only project input vectors to the map units on a rigid grid, this map does not provide much information about the global shape of the data. For all possible R values, three major clusters can be observed from the map. With R getting larger, the structure and shape of the data become more prominent. It is also noticeable that the cluster borders become obscure as R increases.

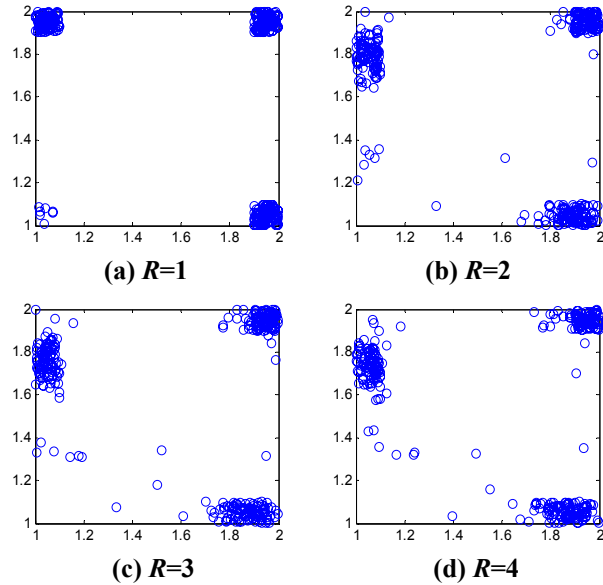


Figure V.9 Projection results with different R values

5.2.4.2 Criteria for Selecting R

A two-dimensional representation produced by the RCP enables one to visualize underlying structure present in the data, and check for dimensionality reduction and clustering tendencies. As shown in Figure V.9, different results are obtained depending on the chosen ranking parameter R . We must decide which R value produces the best result.

If meaningful conclusions are to be drawn from the projection result, as much of the geometric relationships among the data patterns in the original space as possible should be preserved through the projection. At the mean time, it is desirable for the projection result to provide as much information about the shape and cluster structure of the data as possible. Members of each cluster should be close to each other while the clusters should be widely spaced from each other. Thus, a combination of two quantitative measures, Sammon's stress [Sammon 69] and the Davies-Bouldin (DB) index [Davies and Bouldin 79], is used in this work to determine the optimal R .

Sammon's stress measures the distortion between the pairwise distances in both the original and the projected spaces. In order to achieve good distance preservation, Sammon's stress should be minimized. The DB index attempts to maximize the inter-cluster distance while minimizing the intra-cluster distance at the same time. It is commonly used as a clustering validity index, low values indicating good clustering results.

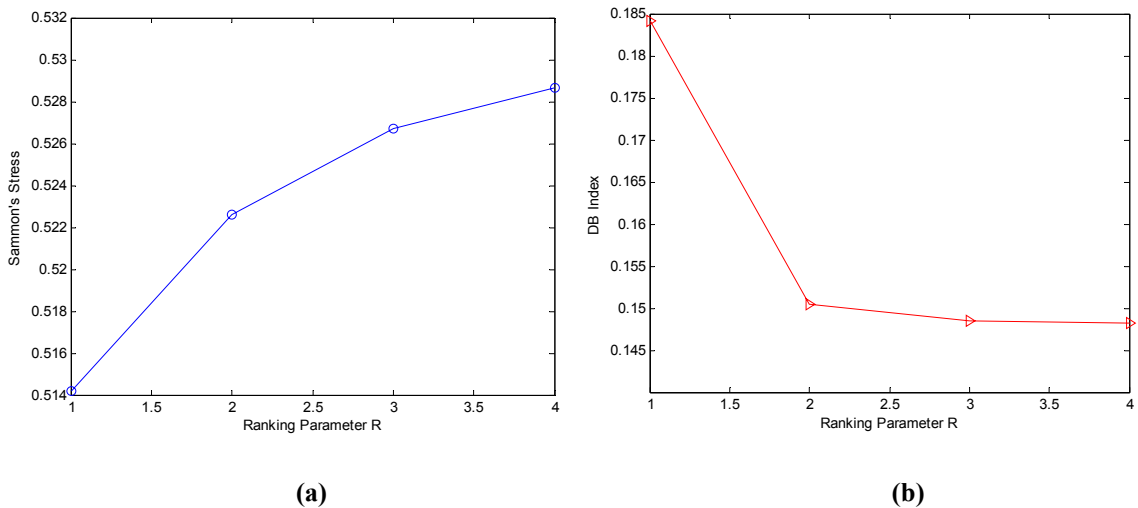


Figure V.10 (a) Sammon's stress and (b) DB index for $R = 1, 2, 3, 4$

For the projection results in Figure V.9, both Sammon's stress and the DB index are calculated for each R value, which are shown in Figure V.10(a) and Figure V.10(b) respectively. It can be seen from Figure V.10 that the two quantitative measures have contradicting trends. As R grows larger, Sammon's mapping increases while the DB index decreases. It is hence impossible to optimize both of the objectives at the same time. We must identify the best compromise, which serves as the optimal R in this context. The task of selecting the optimal R now boils down to a bi-objective optimization problem. A typical way to solve the problem is to use the weighted sum method, which is stated in [Kim and Weck 05]:

$$\min \alpha \frac{J_1(x)}{J_{1,0}(x)} + (1 - \alpha) \frac{J_2(x)}{J_{2,0}(x)}, \quad (\text{V.12})$$

where J_1 and J_2 are two objective functions to be mutually minimized, $J_{1,0}$ and $J_{2,0}$ are normalization factors for J_1 and J_2 respectively, and α is the weighting factor revealing the relative importance between J_1 and J_2 . In the context of this work, J_1 and J_2 correspond to Sammon's stress and the DB index. Assuming these two objective functions have equal importance, α is set to be 0.5. By taking the weighted sum, the two objective functions are combined into a single cost function, which is shown in Figure V.11. The optimization problem is therefore reduced to minimizing a scalar function. As shown in Figure V.11, the objective function reaches its minimum when R equals to 2. Consequently, the condition of $R = 2$ leads to the best compromise between good distance-preservation and good clustering quality. For the three-dimensional example used in Subsection 5.2.4.1, Figure V.9(b), obtained with $R = 2$, is the optimal projection result in terms of the two quantitative measures.

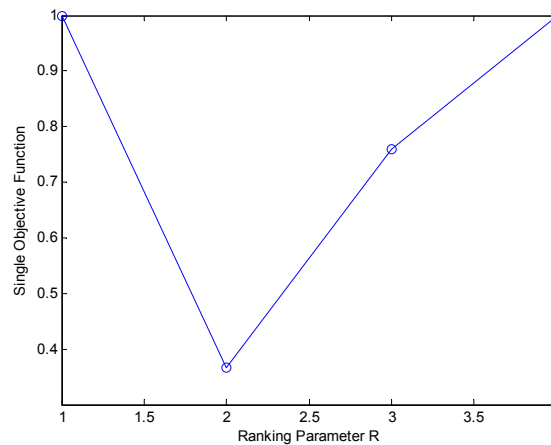


Figure V.11 The optimal R is found at the minimal point of the single cost function.

5.3 Incremental Clustering for Dynamic Document Databases

5.3.1 The Need for Incremental Clustering

Document collections are dynamic in nature. As time passes, new documents are published and added to the database. The size of the document collection is therefore constantly increasing in a real-world environment. Many techniques used for clustering and mapping documents are based on processing the entire collection of papers at once [Booker et al. 99, Morris et al. 02, Morris et al. 03, Vesanto and Alhoniemi 00]. However, there are often situations where additional documents are created and available when the map is already fully built. It is highly desirable for a document clustering model to equip with an ability to perform incremental clustering. Incremental clustering works by assigning data objects to appropriate clusters as they arrive, without having to perform complete reclustering from scratch. In this section, we discuss how the proposed SOM-based approach can be used to incrementally cluster new data. This technique is based on the growth of the document-reference matrix as new documents are added to the collection.

5.3.2 Formation of the Similarity Matrix

A mathematical model introduced in [Morris 05] is used in this study to encode inter-document similarities based on the bibliographic coupling counts, which also handles the dynamic growth of the document collection. For a collection of m documents

$\{d_1, d_2, \dots, d_m\}$ and n references $\{r_1, r_2, \dots, r_n\}$, an $m \times n$ document-reference matrix can be defined to depict the direct citation links between the papers and references:

$$\mathbf{M}_{dr}(i, j) = \begin{cases} 1 & \text{if } d_i \text{ cites } r_j \\ 0 & \text{otherwise} \end{cases} \quad (\text{V.12})$$

Each document is thus represented by a vector, which describes the citations that appear in the document. In this case, the matrix \mathbf{M}_{dr} is binary, whose rows correspond to documents and columns correspond to references. Furthermore, the rows and columns are ordered in the sequences in which the documents and references are published respectively. The following example is the document-reference matrix for a document collection with $m=3$ and $n=7$:

$$\mathbf{M}_{dr} = \begin{array}{c} \left[\begin{array}{ccccccc} \leftarrow \text{references} \rightarrow \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array} \right] \\ \begin{array}{c} \uparrow \\ \text{documents} \\ \downarrow \end{array} \end{array} \quad (\text{V.13})$$

The bibliographic coupling relation, as an indirect citation link, can be derived from the document-reference matrix \mathbf{M}_{dr} . An $m \times m$ matrix \mathbf{M}_{bc} , which contains the bibliographic coupling counts bc_{ij} between all pairs of documents in the collection, can be found by multiplying the paper-reference matrix by its transpose:

$$\mathbf{M}_{bc} = \mathbf{M}_{dr} \mathbf{M}_{dr}^T. \quad (\text{V.14})$$

The element in the i th row and j th column of \mathbf{M}_{bc} is the number of documents cited by both document d_i and document d_j . As an example, using the \mathbf{M}_{dr} given in Equation (V.13), we will have the following bibliographic coupling matrix:

$$\mathbf{M}_{bc} = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 4 & 0 \\ 1 & 0 & 3 \end{bmatrix}. \quad (\text{V.15})$$

The inter-document similarities are then calculated by taking the cosine coefficient of the bibliographic coupling counts [Salton 89]:

$$s_{ij} = \frac{bc_{ij}}{\sqrt{N_i N_j}}, \quad (\text{V.16})$$

where s_{ij} is the similarity value between document i and document j , while N_i and N_j are the total number of document citations for documents i and j respectively. As a result, the similarity matrix is a symmetric matrix that contains the normalized bibliographic coupling counts between all pairs of documents in a collection. Given the \mathbf{M}_{bc} in Equation (V.15), the resulting similarity matrix will be

$$\mathbf{S} = \begin{bmatrix} 1 & 0.58 & 0.33 \\ 0.58 & 1 & 0 \\ 0.33 & 0 & 1 \end{bmatrix}. \quad (\text{V.17})$$

The columns or rows, representing individual documents, are used as input patterns to train a GHSOM.

5.3.3 Dynamic Growth of the Similarity Matrix

In a real-world environment, a document collection grows from an initial set of documents by sequential addition of papers in the order of their publication dates. When a new document is added, it is associated with the existing references and also brings additional references in the collection.

Consider the document collection containing m documents and n references. The document-reference matrix \mathbf{M}_{dr} is an $m \times n$ matrix, whose rows and columns are ordered according to the publication dates. Assume a new document d_{m+1} is added to the

collection and introduces n_1 new references. This addition results in a new row and n_1 new columns in the original \mathbf{M}_{dr} :

$$\mathbf{M}'_{dr} = \begin{array}{c} \left| \begin{array}{cccccc|ccc} \leftarrow & n & \rightarrow & & & & \leftarrow n_1 \rightarrow & & & \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{array} \right| \begin{array}{c} \uparrow \\ m \\ \downarrow \\ \leftarrow (m+1)\text{th row} \end{array} \end{array} \quad (\text{V.18})$$

The new row vector corresponds to the added document d_{m+1} . It can be partitioned into a $1 \times n$ vector, which describes d_{m+1} 's citations to the n existing references, and a $1 \times n_1$ vector of all ones, which corresponds to d_{m+1} 's citations to the new references. This $(m+1)$ th row can hence be represented as $[\boldsymbol{\delta} \mathbf{1}]$, where $\boldsymbol{\delta}$ denotes the $1 \times n$ vector and $\mathbf{1}$ denotes the $1 \times n_1$ vector of all ones. Because no citation exists between the newly added references and the original set of documents, the corresponding elements are all zeros, resulting in an $m \times n_1$ zero matrix on the upper right part of the new matrix \mathbf{M}'_{dr} . This matrix therefore can be represented in the following recursive matrix equation:

$$\mathbf{M}'_{dr} = \begin{bmatrix} \mathbf{M}_{dr} & \mathbf{0} \\ \boldsymbol{\delta} & \mathbf{1} \end{bmatrix}, \quad (\text{V.19})$$

where $\mathbf{0}$ is the $m \times n_1$ zero matrix. The new bibliographic matrix \mathbf{M}'_{bc} can be obtained by applying Equation (V.14):

$$\mathbf{M}'_{bc} = \mathbf{M}'_{dr} \cdot \mathbf{M}'_{dr}{}^T = \begin{bmatrix} \mathbf{M}_{dr} \cdot \mathbf{M}_{dr}{}^T & \mathbf{M}_{dr} \cdot \boldsymbol{\delta}^T \\ \boldsymbol{\delta} \cdot \mathbf{M}_{dr}{}^T & \boldsymbol{\delta} \cdot \boldsymbol{\delta}^T + \mathbf{1} \cdot \mathbf{1}^T \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{bc} & \mathbf{M}_{dr} \cdot \boldsymbol{\delta}^T \\ \boldsymbol{\delta} \cdot \mathbf{M}_{dr}{}^T & \lambda \end{bmatrix} \quad (\text{V.20})$$

where λ is the total number of 1's in the $(m+1)$ th row in \mathbf{M}'_{dr} . For the example document collection, $\boldsymbol{\delta} = [0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0]$, $[\boldsymbol{\delta} \mathbf{1}] = [0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1]$, and thus $\lambda = 5$. Therefore the additional row vector v_{m+1} in \mathbf{M}'_{bc} can be computed as

$$v_{m+1} = [\boldsymbol{\delta} \cdot \mathbf{M}_{dr}{}^T \ \lambda] = [1 \ 1 \ 1 \ 5], \quad (\text{V.21})$$

and the new column is v_{m+1}^T . \mathbf{M}'_{bc} is a $(m+1) \times (m+1)$ matrix. It becomes the similarity matrix \mathbf{S}' after normalization.

5.3.4 An Incremental Document Clustering Algorithm

The additional row (or column) vector in \mathbf{S}' represents the new document. While this new data item is added to the map by the original training, the initial topology of the map should be preserved. To achieve this result, a single-pass clustering technique can be applied to the new documents, which processes the new documents sequentially and compares each of them to all existing clusters. Each new document can then be mapped using the RCP algorithm.

However, a challenge exists for the incremental clustering of documents. As the original training was completed by using the rows (or columns) of the original similarity matrix \mathbf{S} , the prototype vectors of the trained SOM which correspond to the existing cluster centers, have a dimension of m . Every time a new document is added to the database, the dimension of the new row (or column) vector representing the new document grows by one. Therefore the new data vectors can not be compared with the prototype vectors directly because of the mismatch in dimensions. To accommodate the dynamic growth of the similarity matrix, an incremental clustering algorithm for dynamic document processing is proposed. The main steps are summarized in Table V.3.

As a new document is added, the prototype vectors are first expanded by one dimension, representing the bibliographic coupling relation to the new document. The corresponding element in each prototype vector is randomly initialized. The incremental clustering algorithm then updates the cluster center associated with the BMU of the new

data point and its neighboring cluster centers as the new data point is presented. Subsequently the new data point is projected to the map by finding the weighted average of the positions of the nearby existing cluster centers. By taking these steps repetitively for each new document, the additions of new documents will be reflected in the map without affecting the original map topology.

Table V.3 Basic steps of the incremental clustering algorithm

1.	For a new document d_i , expand the dimension of the prototype vectors by one. Initialize the value of the new element in each prototype vector with a random number.
2.	The standard SOM training algorithm is applied to the map so that the expanded prototype vectors are adapted to the new input vector.
3.	Apply the Ranked Centroid Projection to the new data and allocate it to the original map space.
4.	Repeat steps 1-3 if there is any new document to be added to the database.

CHAPTER VI

SIMULATION RESULTS

6.1 Overview

The simulations presented in this chapter serve as demonstrations of the applicability of the proposed SOM-based visualization method, in which three different data sets have been applied. Each of the sections focuses on a separate data set. The first data set is a two-dimensional artificial data set used for illustrative purpose. The other two data sets are real-world document sets, which are the core subject of this chapter. These two simulations aim at solving practical problems in text-based data mining. They are examples of a pragmatic, data-centered approach. In these examples the SOM-based approach is used to capture and reveal relational features of a collection of documents covering different scientific fields.

The general goal is to examine the usefulness and limitations of the proposed approach in clustering and visualizing high-dimensional data, and more in particular to illustrate the capabilities of this approach in contributing to a more efficient and meaningful knowledge representation of document collections. The emphasis is placed on the description of the methodology and not on the explanations of the configuration on the map.

6.2 Software Tools Used

A number of software tools were used to perform the simulations. The functions of the tools and their URLs are given as follows:

- DIVA package [Morris et al 02] in Matlab for citation analysis, calculating the similarity matrices and extracting the cluster labels, of which the author is one of the main developers. (http://samorris.ceat.okstate.edu/web/diva_web/)
- SOM Toolbox [Vesanto et al. 00] in Matlab for creating the SOM (<http://www.cis.hut.fi/projects/somtoolbox>)
- GHSOM Toolbox [Chan and Pampalk 02] in Matlab for creating the GHSOM (<http://www.ofai.at/~elias.pampalk/ghsom/>)
- Statistics Toolbox in Matlab for implementing the two-dimensional clustering tasks (<http://www.mathworks.com/products/statistics/>)

6.3 An Illustrative Data Set

The first data set was used to illustrate how the proposed approach performs. This data set is the zoo dataset obtained from the UCI Machine Learning Repository [Newman et al. 98]. The zoo data set comprises 100 artificial data items representing animals with 16 attributes (besides names and classes). The animals are divided into 7 classes. An additional numeric-valued class attribute is given to indicate the class distribution, which is excluded from the training data.

A GHSOM is trained for this data set, which starts with a 2×2 SOM. The training process continues with additional units being added until the quantization error drops

below a certain percentage of the overall quantization error of the unit at the first layer. In this example, a three-layer GHSOM was generated by setting the thresholds $\tau_1=0.6$ and $\tau_2=0.0002$. The GHSOM is illustrated in Figure VI.1, where the top-layer map is depicted in gray and the bottom layer maps are in white. The projection result generated using the RCP is shown in Figure VI.2(a), in which the complete set of animals is mapped across the 3×2 map grid. The map was generated with the ranking parameter $R=3$, which is selected using the method described in Subsection 5.2.4.2. This value results in the best compromise between the projection accuracy and clustering quality. Members of each class are marked by different symbols. Meaningful clustering of the animals is shown in the map, with several well-separated clusters located in different regions of the map. The resulting projection gives a rough representation of the global shape of the data. For comparison, the projections by the basic SOM, PCA, and Sammon's mapping are presented in Fig. VI.2(b)-(d) respectively.

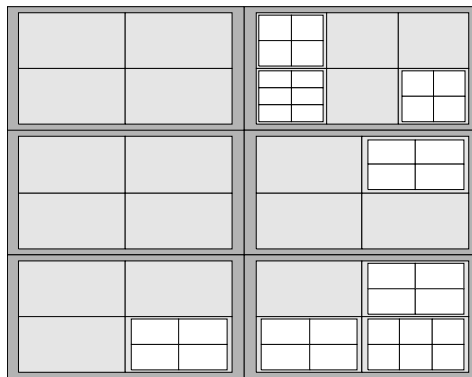


Figure VI.1The resulting 3-layer GHSOM for the zoo data set

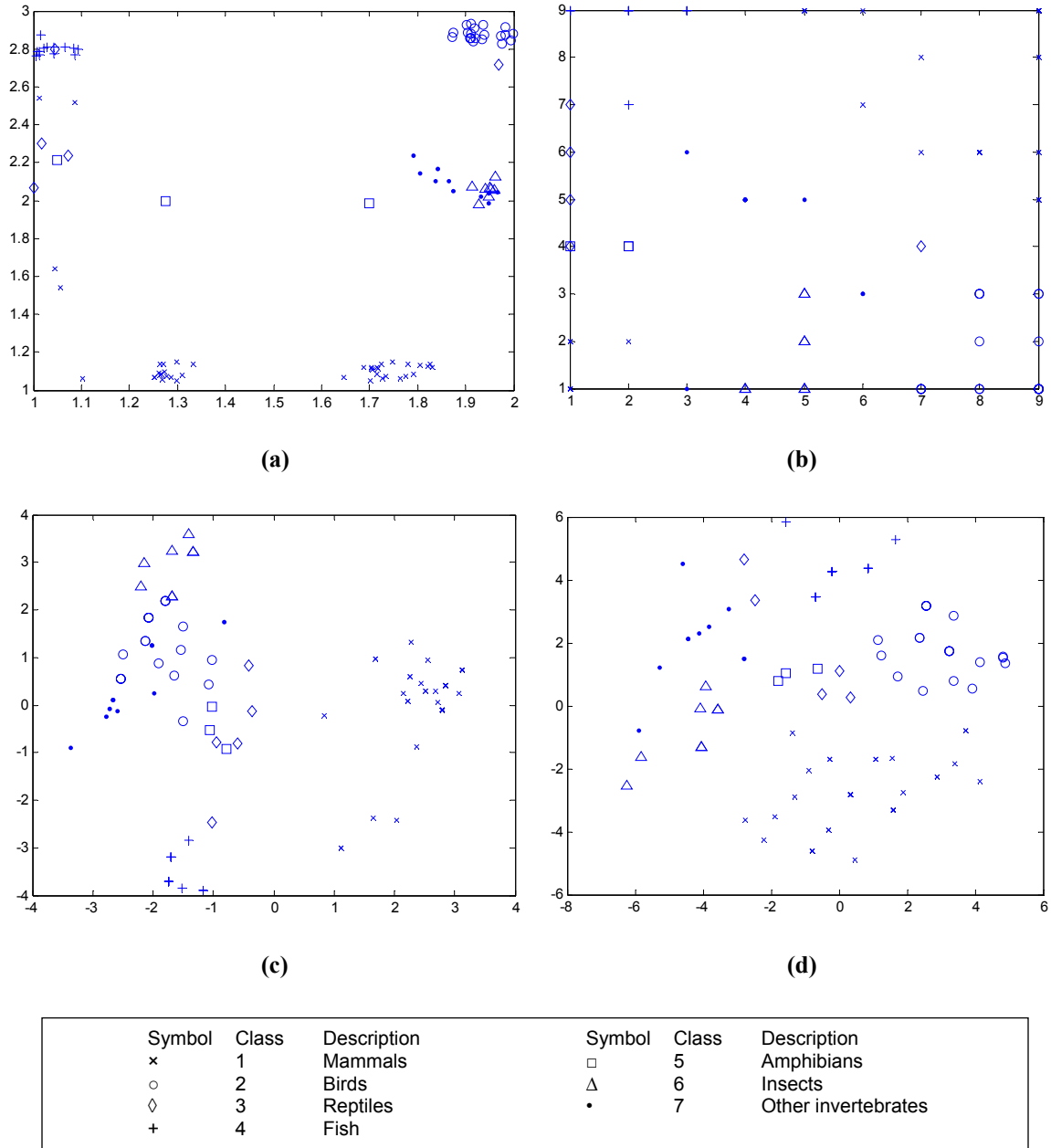


Figure VI.2 Two-dimensional projections of the zoo data. (a) The first-layer map using the proposed approach ($R=3$). (b) A 9×9 SOM with data projected to the corresponding BMUs. (c) PCA. (d) Sammon's mapping.

Visually inspecting the results presented in Figure VI.2, Figure VI.2(a) is evidently the best, showing good distribution and clean separation of the clusters. The basic SOM projection in Figure VI.2(b) was unable to produce isolated clusters, although the clusters are somewhat identifiable. PCA and Sammon's mapping failed to capture and

present the cluster structure, as shown in Figure VI.2(c) and VI.2(d) respectively. The proposed approach appears to outperform the other methods in visualizing the underlying structure of data. Fairly good resolution was achieved in the 3×2 map in Figure VI.2(a), while a satisfactory resolution requires a considerably larger map size for the basic SOM, which has 9×9 units in this case. Besides, using the proposed approach, refined maps showing the intra-cluster details are available for clusters of interest.

Based on the initial separation of clusters in the first layer, further maps were automatically generated and trained to represent the sub-clusters in more details. In this simulation, 6 submaps are formed on the second layer. One example is shown in Figure VI.3, which is the submap expanded from the right-middle unit in the top-layer map. As can be seen in Figure VI.2(a), classes 6 and 7 are somewhat mixed and form a large-scale cluster in the vicinity of the right-middle unit. This large-scale cluster is distinguished from the other classes in that it represents the group of invertebrates, animals without a backbone. Figure VI.3 gives a zoomed-in view of this group. Interesting small-scale clusters can be noticed in this submap of 2×2 units. Members of class 6, the insects, are mapped to the right half of the map, while most of the animals of class 7 occupy the left part. Further grouping is also discernible. The animals in the upper-left corner are aquatic. The two land insects, flea and termite, are close to each other, indicating significant similarity between them, while far away from the airborne insects. Gnat, ladybird, moth and housefly, all of which are non-venomous and airborne, form a tight sub-cluster. Scorpion is mapped close to honeybee and wasp, as they are all venomous.

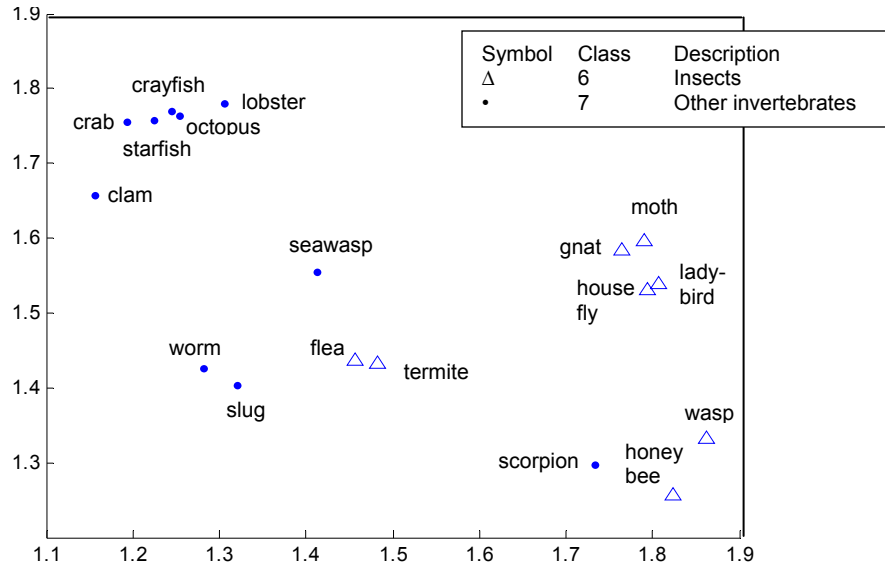


Figure VI.3 Submap of invertebrates generated with $R=3$

In this simulation, two quantitative measures, Sammon's stress and the DB index, are computed for each method. The results are summarized in Table VI.1. Although inadequate as sole performance metrics for the goodness of the projection, they can be used to some degree to characterize the resulting projection. As introduced in Subsection 5.2.4.2, Sammon's stress measures the distortion between the pairwise distances in both the original and the projected spaces. The DB index is commonly used as a clustering validity index with low values indicating good clustering results. The data set was projected a number of times with each method to obtain the results in Table VI.1. Sammon's mapping scores the best value on the Sammon's stress. This is not surprising, since the other three methods do not aim at minimizing Sammon's error measure alone. The result of the RCP method is calculated based on the first-layer map of only 6 units. The projection in this layer is coarse, which may explain why the corresponding Sammon's stress is significantly higher than the others. Doubling the number of first-layer units will make the resulting stress of the RCP comparable to the 9×9 SOM. The

results of the DB index show that the RCP produces a better result, which is consistent with the visual inspection. This implies that for the purpose of classification and categorization, the RCP is to be preferred over the other methods presented here.

Table VI.1 Quantitative evaluation results

	PCA	Sammon's Mapping	SOM (9×9)	Ranked Centroid Projection
Sammon's stress	0.1769	0.0796	0.1944	0.6667
DB index	0.2283	0.2562	0.2357	0.1666

6.4 Document Data Sources

In the next two sections of this chapter, two collections of journal papers covering different scientific fields are used as examples. To visualize the document collections, it is necessary to find document links based on the inter-document citations in the first stage of the proposed approach. Journal papers with citations were extracted from the Science Citation Index (SCI) [Garfield 94], the first database compiled by the Institute for Scientific Information (ISI) in the mid-1960's. This database provides an accumulating body of readily available catalogued bibliographic data on scientific research. The SCI has an annual coverage of about 500,000 individual documents with roughly 8,000,000 citations in the reference lists. The documents stem from over 3,500 professional core journals, especially the important periodicals in the natural and life sciences [Tijssen 92]. The journal papers used in this study were retrieved on line from ISI Web of Science.

The raw files of documents generated from queries to the citation service form a database and are stored in MS Access.

6.5 A Collection of Journal Papers on Self-Organizing Maps

The second data set is constructed based on a collection of journal papers from the SCI on the subject of Self-Organizing Maps. Using the term ‘Self Organizing Maps’ in the general search function of ISI Web of Science, a set of 1,349 papers was collected corresponding to journal articles published from 1990 to early 2005. A SQL query was used to count the number of bibliographic couplings for each pair of documents in the data set. The poorly related papers, i.e. papers that did not have at least 5 bibliographic couplings with another document, were discarded, reducing the total number of papers to 638.

Citation based similarity matrices are used extensively when working with patents and journal articles from sources that provide citation data such as the SCI [Morris et al. 02, Morris et al. 03]. In this study, document citation patterns are used to describe the inter-document relationships between pairs of documents, based on which a similarity matrix is built to store pair-wise similarity values between papers. Each dimension of the similarity matrix corresponds to one document in the data set and the value of each element is equal to the relative strength of the citation relationship between the corresponding document pair. Similarities calculated from citations generally produce meaningful document maps whose patterns expose clusters of documents and relations among those clusters.

In this study, the similarity matrix is constructed based on one type of inter-document citations, namely the bibliographic coupling, as discussed in Chapter IV. The similarity matrix is therefore a symmetric matrix that contains the bibliographic coupling counts between all pairs of documents in the database. Each element in the similarity matrix is calculated as discussed in Equation (IV.4). After the document encoding process, a 638×638 similarity matrix is constructed. Each row (or column) of the similarity matrix depicts the citation pattern of a document in a 638 dimensional space, which is then used as the input vector to train a GHSOM network.

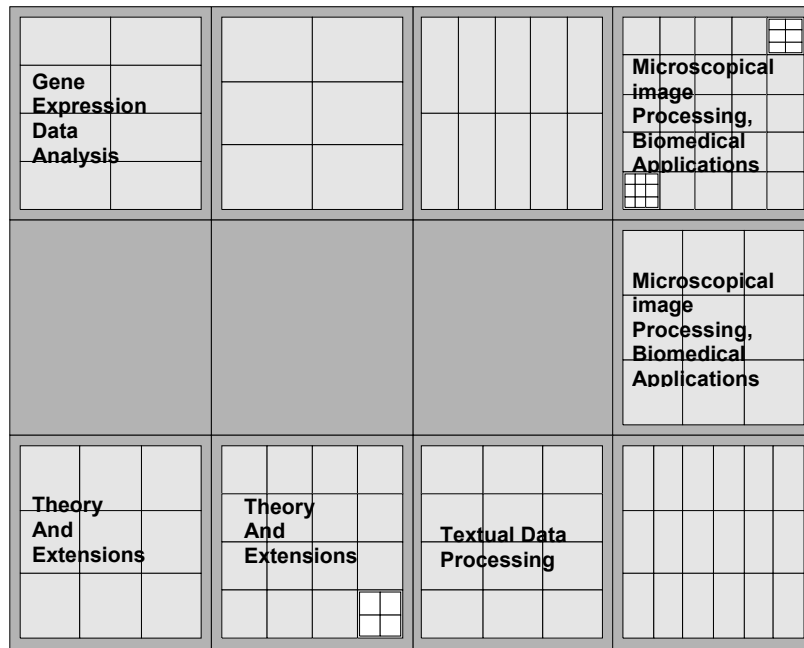


Figure VI.4 The resulting 3-layer GHSOM for the collection of SOM papers

Starting with a 2×2 SOM at the first layer, a 3-layer GHSOM was generated by setting the thresholds $\tau_1=0.8$ and $\tau_2=0.008$, as illustrated in Figure VI.4. The first-layer map, consisting of 3×4 units, shows four major clusters in the collection of journal papers. The topics associated with each map unit are labeled in Figure VI.4. The labels

are derived after examining manually the paper titles from each cluster for common subjects.

The projection of all the papers onto the first layer map is shown in Figure VI.5, where documents are marked as circles in the rectangular area. This projection is obtained by using a ranking parameter $R = 5$. Four major clusters can be observed from the figure, which are located around the four corners of the map.

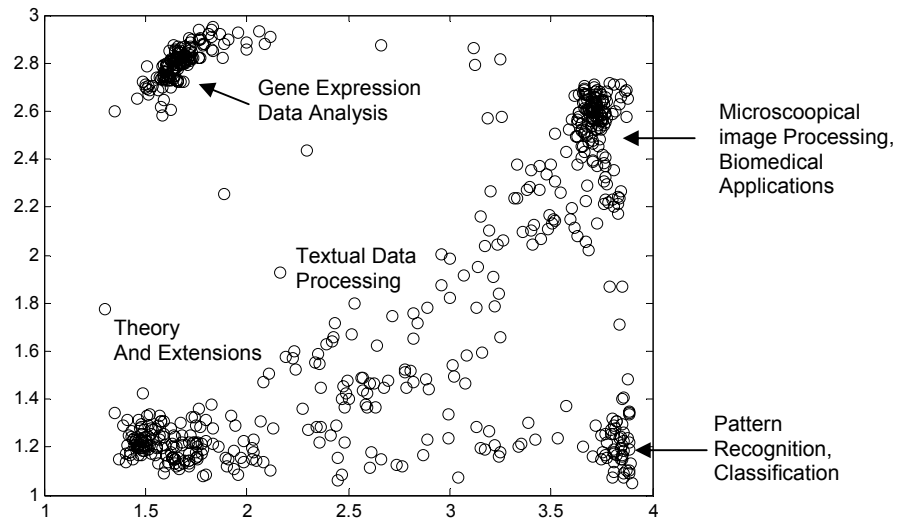


Figure VI.5 The projection result of the journal papers on the SOM, where documents are marked as circles in the rectangular area.

To enhance the visual representation of the paper collection, the size of the document marker can be made proportional to the number of times a document has been cited. Figure VI.6 illustrates such a map, in which highly cited papers are displayed as large circles with a darker color while the less cited ones are displayed as smaller light-color circles. Important papers, usually distinguished by large citation counts, are thus made standing out on the document map. In this collection of SOM papers, three papers are extraordinarily heavily cited, as marked in Figure VI.6. The two large circles in the top part of the map, [Toronen et al. 99] and [Tamayo et al. 99], which appear to be

closely related to each other, correspond to two important works on applying SOM for clustering gene expression data. Tamayo and colleagues used the SOM to cluster genes into various patterned time courses and also devised a gene expression clustering software, GeneCluster. Another implementation of the SOM was developed by Toronen et al. in 1999 for clustering yeast genes. Yet another heavily cited paper is [Kohonen 90], which is cited by a large portion of the documents in this document set. The foundational papers Kohonen published in 1980's, such as [Kohonen 82], are not available from the ISI web service.

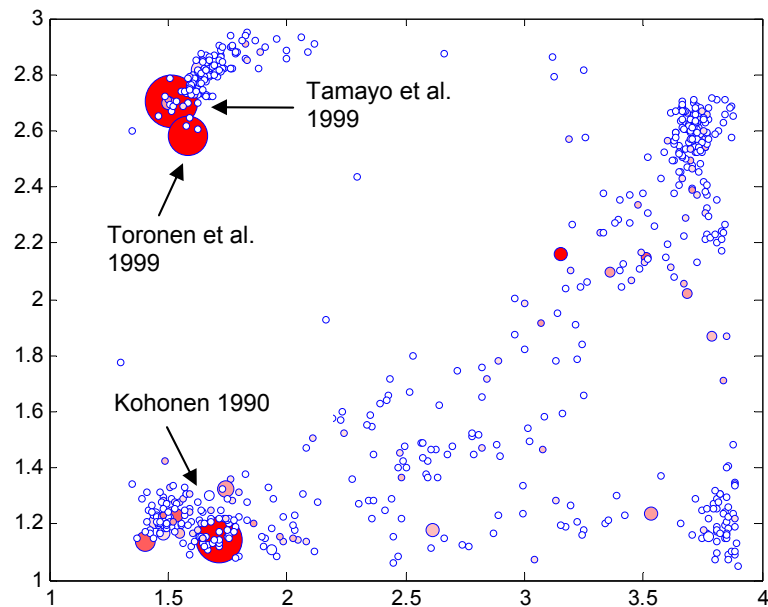


Figure VI.6 An enhanced visualization of the SOM papers, where the size of the document marker is proportional to the number of times a document has been cited.

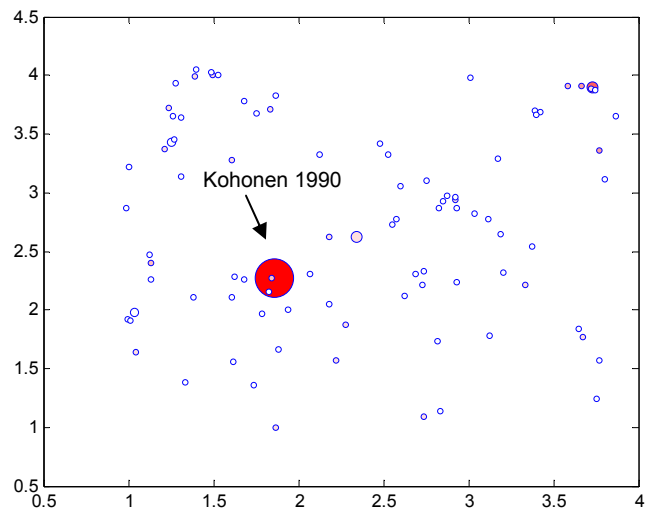
Several map units in the first layer SOM are expanded to the second layer. One example is shown in Figure VI.7. Figure VI.7(a) shows the submap expanded from the second node from the left in the bottom row in the first-layer map and Figure VI.7(b) shows the projection result of this submap. This map, consisting of 3×3 units, represents

a cluster of papers covering the theoretical aspect of the SOM, within which Kohonen's seminal paper is located.

Some of the units in the second-layer maps are further expanded as distinct SOMs in the third layer. Due to the incompleteness of this document collection, limited information about this subject domain is revealed from the map displays.

Topographic Mapping		SOM Extensions
	Analysis of the SOM	

(a) Submap for cluster number 2 with labels



(b) The projection result of the submap

Figure VI.7 A submap of the resulting GHSOM for the SOM paper collection

6.6 A Collection of Journal Papers on Anthrax Research

The third data set is a collection of journal papers on anthrax research, which is also obtained from ISI website. Anthrax research makes an excellent example for testing the performance of document clustering and visualization. The subject is well covered by the SCI. A great deal of the research has been performed in the past 20 years. A review paper [Bhatnagar and Batra 01] is available where the names of key papers in this field are identified and discussed. The anthrax paper set collected for this simulation contains 987 documents corresponding to journal papers published from 1981 to the end of 2001 [Morris 05].

A 987×987 similarity matrix is formed to train a GHSOM. A 3-layer GHSOM was resulted by setting the thresholds $\tau_1=0.78$ and $\tau_2=0.004$, as illustrated in Figure VI.8. The first-layer map consists of 3×4 units. The projection of data samples onto this layer is shown in Figure VI.9, which was produced by setting the ranking parameter $R = 3$.

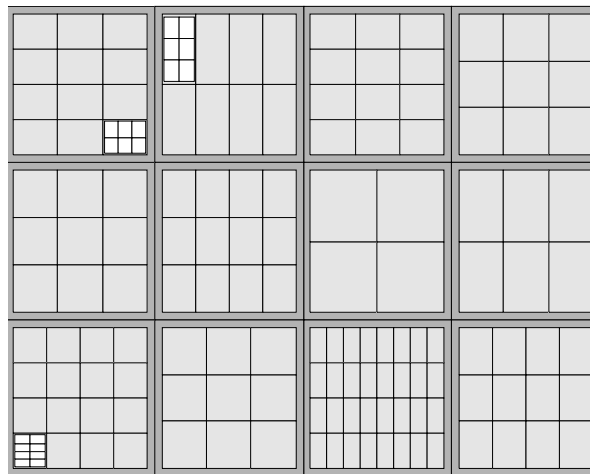


Figure VI.8 The resulting 3-layer GHSOM for the collection of anthrax papers

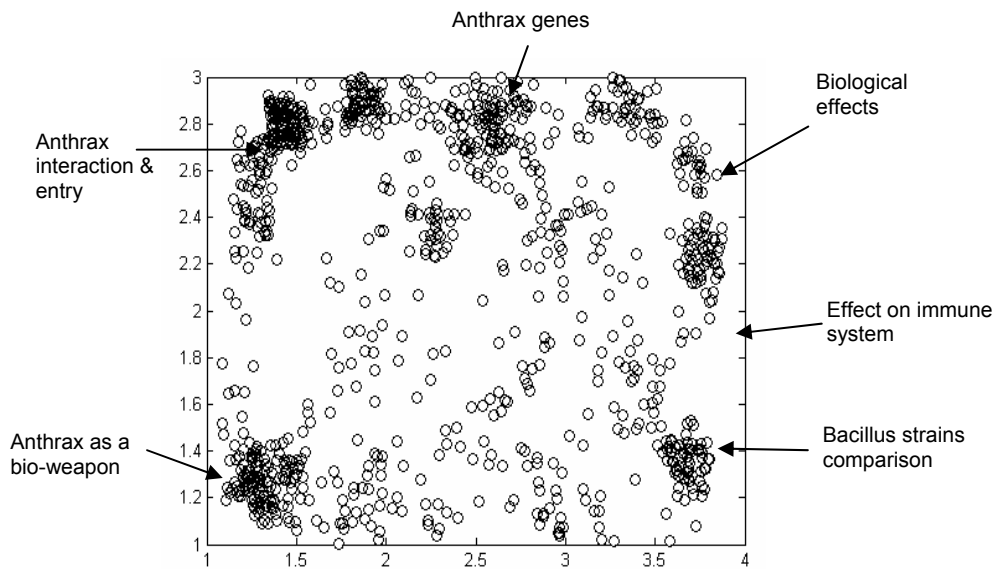


Figure VI.9 First-layer projection of the anthrax journal papers, where documents are marked as circles in the rectangular area. The cluster labels are added manually showing the major subject of each cluster of papers.

In Figure VI.9, several major clusters can be seen on the map with their subjects labeled. The labels are manually created after browsing the paper titles in each individual cluster. Starting from the upper left corner of the map and going clockwise, we can see that the topics of the papers change with different locations on the map. The cluster of papers in the upper left corner is focused on how anthrax moves, interacts with, and enters host cells. Note that several smaller groups are visible inside this cluster, which implies expanding this cluster to a further layer would reveal several sub-topics. To the right, the papers in the upper center of the map are found to deal with anthrax genes. The cluster located in the upper right corners cover biological effects of anthrax, while the cluster right below it covers the effect of anthrax on the immune system. In the lower right corner of the map, another group of papers exists, which deals with the comparison of anthrax and other *Bacillus* strains. A tight cluster is formed in the lower left corner of the map, which discusses the use of anthrax as a bio-weapon. As a whole, several obvious

groups of documents are formed on the map, which relate to different research focuses in the context of anthrax research. Fundamental research topics are located in the upper portion of the map, which are somewhat in vicinity of each other. There are no obvious borders between these groups as the topics are closely interrelated. On the contrary, other relevant topics on anthrax are mapped to the lower portion, which are rather far away from the fundamental research topics and from each other. It can be seen that the geometric distance indicates the degree of relevance between documents. It is also noticeable that many documents sitting between clusters, which is the result of the heavily overlapped research coverage.

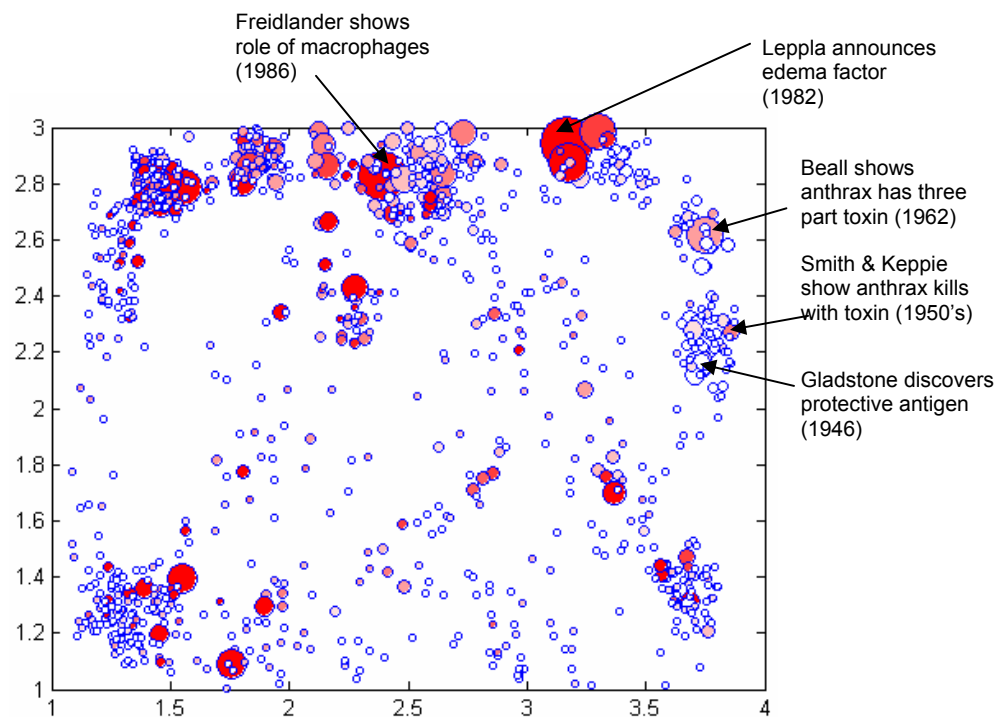


Figure VI.10 Labeled map of the anthrax paper set with paper marker sizes proportional to the number of times the corresponding papers have been cited.

More information about the collection of papers can be obtained by identifying the seminal papers among all papers. For this purpose, the document marker sizes are

made proportional to the number of times they have been cited. The result is shown in Figure VI.10.

Several seminal papers can be identified from Figure VI.10, five of which are marked in the figure as examples. The earliest seminal paper is the Gladstone paper published in 1946, in which he reported on the discovery of protective antigen. In the 1950's, Smith and Keppie showed in their paper that anthrax kills through a toxin. These papers are landmark papers forming the foundation for anthrax research and they fall into the cluster of anthrax effect on immunity. Later another influential paper was published in 1962 when Beall showed anthrax has a three-part toxin. Leppla announced edema factor in his paper in 1982. These papers mainly deal with the effect of anthrax on host cells. Another heavily cited paper was on macrophages published by Freidlander in 1986, which became the key paper in this area.

Based on the initial separation of the most dominant topical clusters in the document collection, further maps can be automatically created to represent the topics in more detail. One second-layer submap is presented in Figure VI.11, which consists of 4×3 neurons. This map is expanded from the neuron in the upper left corner in the first-layer map. It was trained using a total of 192 papers, which are represented by the parent neuron in the preceding layer. The corresponding projection of the data samples is shown in Figure VI.12. In the second layer, the papers are further clustered into three groups: anthrax effect on macrophages, anthrax delivery, and anthrax interaction. This result is consistent with the first-layer representation. One unit on this second-layer map is further expanded as a separate SOM in the third layer.

		Anthrax Delivery						
	Anthrax Effect on Macro-phages							
		<table border="1"> <tr> <td>Anthrax</td> <td></td> <td></td> </tr> <tr> <td>Interaction</td> <td></td> <td></td> </tr> </table>	Anthrax			Interaction		
Anthrax								
Interaction								

Figure VI.11 One submap expanded from the upper left neuron in the first-layer map.

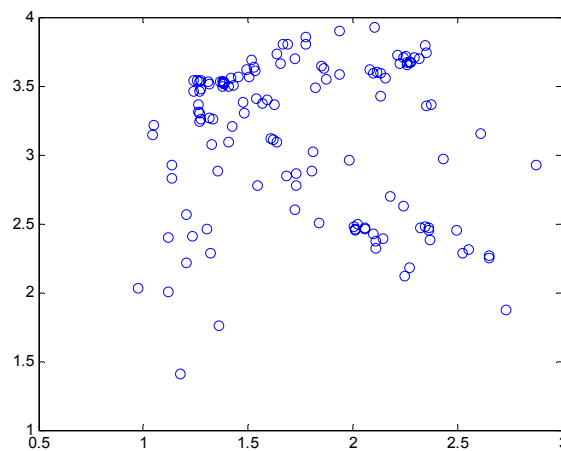


Figure VI.12 The second-layer document projection of the upper-left cluster in the preceding layer

6.7 Adding New Documents to an Existing Map

This section is to demonstrate the performance of the presented projection method under a dynamic environment, where new documents are added to the existing document collection. In this example, the collection of 638 SOM papers is used again. To simulate the dynamic data set, we assume that the first 600 papers in the collection are the initially available data, which are transformed by using the similarity representation. The remaining 38 papers, which were published later than the existing paper set, are treated as

new papers. An initial map was trained using the set of 600 papers, which is illustrated in Figure VI.13. Several clusters can be seen from the map, which are labeled with the corresponding subjects.

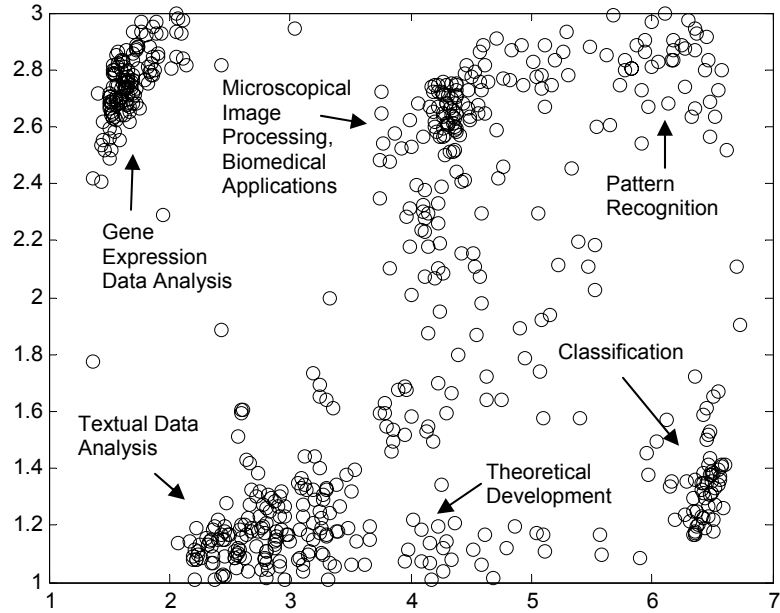


Figure VI.13 The first-layer map of the initial 600 papers for training

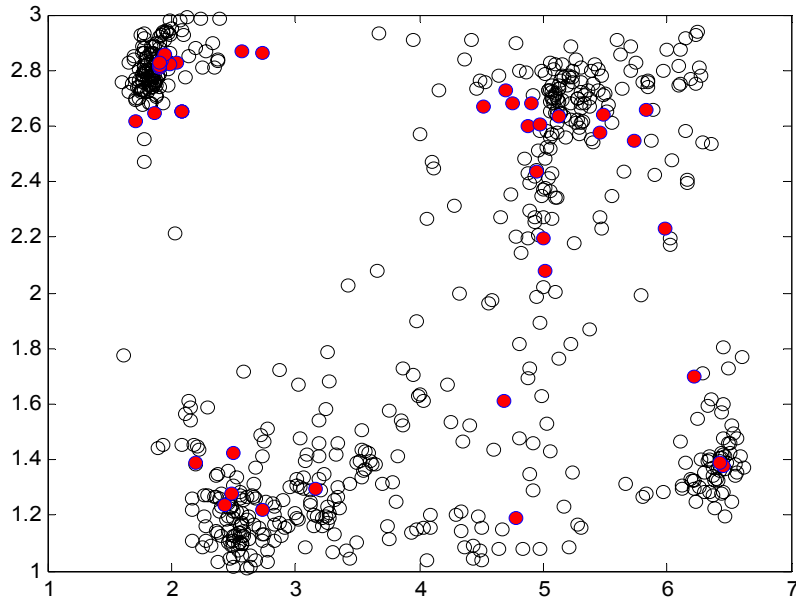


Figure VI.14 The first-layer map with the additional 38 papers added as new data. The red circles represent newly added papers.

Assume the remaining 38 papers were received after the initial map was created. The addition of these new papers results in the dynamic growth of the similarity matrix of the original document set. For every additional paper, a new row and column will be added to the original similarity matrix. Using Equation (V.21), the new vector, which represents the new paper, can be calculated by simple matrix operations. Thereafter each new document can be mapped sequentially to the existing map using the incremental clustering method as described in Section 5.4.4. The resulting map with the 38 additional papers is shown in Figure VI.14, in which the new papers are displayed as red circles. Most of the new papers are mapped to existing clusters. Take one cluster for example: the top left cluster in Figure VI.13 covers the usage of SOMs in gene expression data analysis. 10 out of the 38 new papers are mapped to this cluster, as shown in Figure VI.14. Figure VI.15 shows the 3-dimensional timeline plot of the papers. In this figure, papers are mapped in the order of their publication dates along a third axis, so that vertical streams of circles represent clusters of papers in chronological order. The new papers are highlighted in the figure, which are all located on the top portion of the 3-dimensional display space indicating they were published comparatively late. The cluster of gene expression data analysis is quite distinguished from others, which was formed around 2000 as can be seen from the map. The titles, journals, authors and publication dates of these papers are listed in Table VI.2. Most of the papers in the list are from biomedical science journals, with a topic matching the subject of the existing cluster.

The incremental clustering result updates the existing map with new publications and presents changes over time in a specific research field. Visual information of the

amount of activity in different research fields can be obtained by comparing two maps, since the algorithm does not change the initial topology of the map.

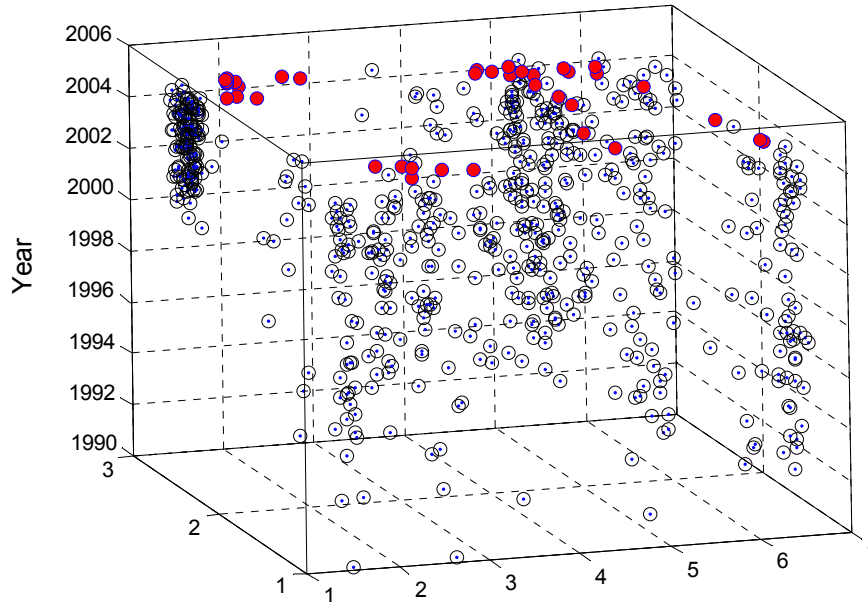


Figure VI.15 3D timeline of the 638 SOM papers

Table VI.2 List of new papers mapped to the cluster labeled “gene expression data analysis”

Title	Source	Year	Author
Open source clustering software	BIOINFORMATICS	6/12/2004	de Hoon MJL
Modulation of gene expression by alloimmune networks following murine heart transplantation	MOL GENET GENOMICS	7/1/2004	Christopher K
Environment-dependent one-body score function for proteins by perceptron learning and protein threading	J KOREAN PHYS SOC	8/1/2004	Cheon M
A hybrid self-organizing maps and particle swarm optimization approach	CONCURR COMPUT-PRACT EXP	8/10/2004	Xiao X
Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering	BMC BIOINFORMATICS	8/23/2004	de Brevem AG
Global gene expression patterns spanning 3T3-L1 preadipocyte differentiation	CAN J ANIM SCI	9/1/2004	Hansen C
Analysis of the major histocompatibility complex in graft rejection revisited by gene expression profiles	TRANSPLANTATION	9/27/2004	Christopher K

Title	Source	Year	Author
Clustering binary fingerprint vectors with missing values for DNA array data analysis	J COMPUT BIOLOGY	10/1/2004	Figuroa A
Mapping high-dimensional data onto a relative distance plane - an exact method for visualizing and characterizing high-dimensional patterns	J BIOMED INFORM	10/1/2004	Somorjai RL
Modelling and optimal control of fed-batch processes using a novel control affine feedforward neural network	NEUROCOMPUTING	10/1/2004	Xiong ZH

CHAPTER VII

DESIGN OF THE SOFTWARE TOOLBOX

7.1 Overview

To implement the document clustering and visualization proposed in this study, a Document Visualization and Analysis Toolbox has been developed. The majority of the exploration and analysis functions of the toolbox are implemented in Matlab, while document storage is done using MS Access.

The toolbox is a software tool supporting intuitive, user friendly exploration of document collections by visualizing documents on a two-dimensional map. The resulting maps usually exhibit a clearly detectable structure by clustering documents by topic. This structure allows us to gain insight into the contents of the document collection as well as to get a rough overview of the relationships among documents and clusters of documents. It can thus be used to identify inherent knowledge contained in huge document datasets.

In this chapter, a detailed discussion of the overall analysis process and exploration functions the toolbox offers will be provided. Figure VII.1 shows the main graphic user interface (GUI) of the toolbox, which is used to manage projects and control the exploration of document maps. Along the menu bar at the top of the GUI are five pull

down menus. Four groups of information display and control are below the menu bar. Section 7.2 describes the functioning of the toolbox as a system, showing each step in the document visualization and analysis process. Section 7.3 discusses the general procedure involved in setting up the workspace using the menu bar. Section 7.4 provides details on displaying and manipulating maps using the control groups. An example of visualizing a set of journal papers is also given in the following sections to illustrate the usage of the toolbox.

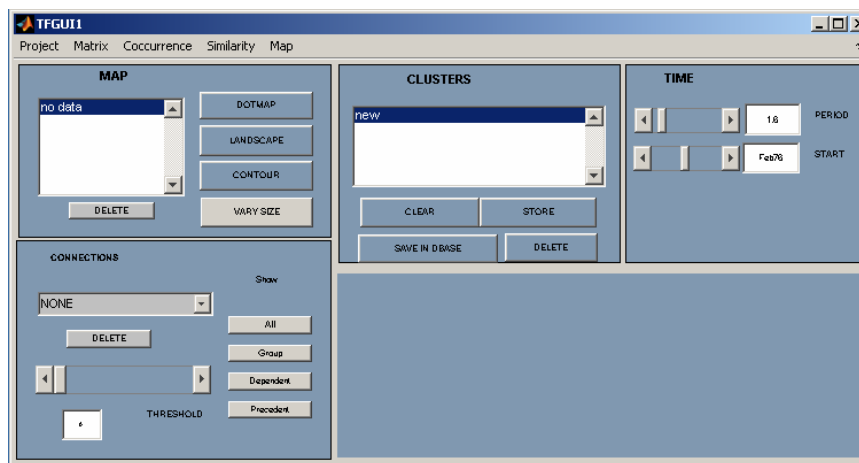


Figure VII.1 Main GUI of the Document Visualization and Analysis Toolbox

7.2 System Description

The main design framework of the toolbox follows the procedure of the proposed SOM-based approach described in Chapter V. Figure VII.2 shows a flow chart of the process involved while developing a visual representation using the toolbox.

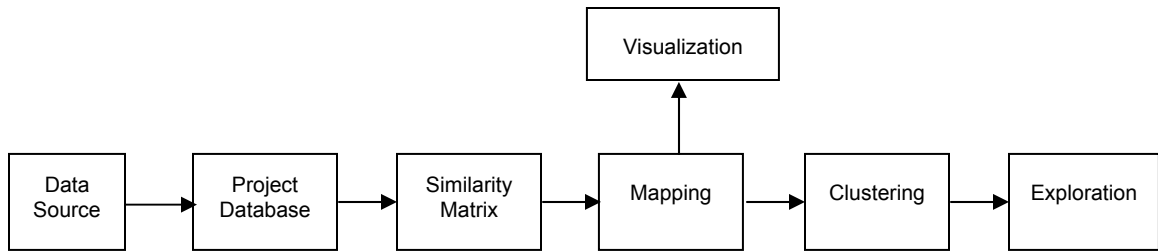


Figure VII.2 Flow of work of the document visualization and analysis toolbox

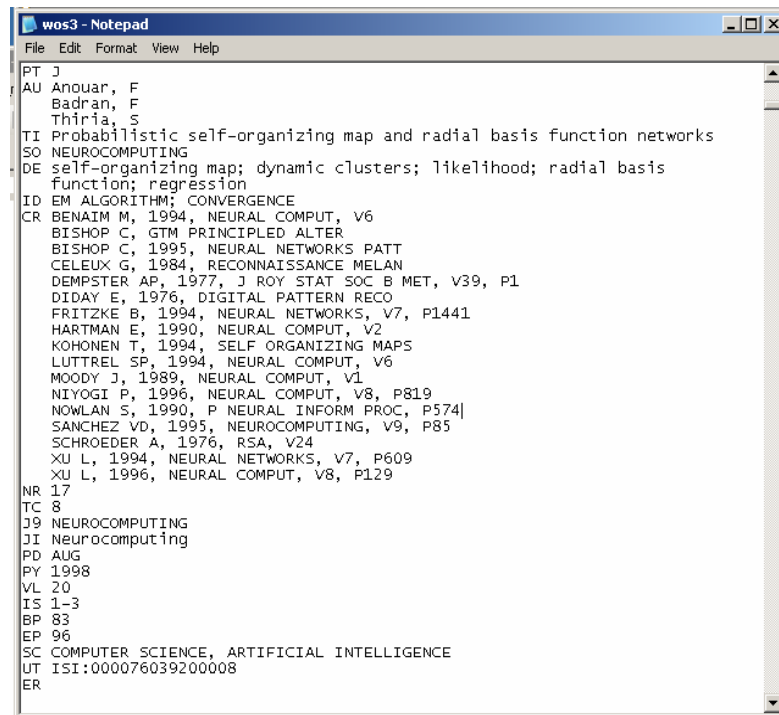
A typical process for analyzing large document sets begins with identifying suitable electronic database sources for the problem at hand. Once the relevant electronic data sources have been located, a set of documents of interest can be collected and loaded into the project database from the sources. Following this, a similarity matrix, based on document citations, is created to measure the strength of the links between all pairs of documents in the database. Based on these links, the documents are mapped onto a two-dimensional space for visualization. Such mapping is done by a GHSOM, which usually organizes documents in a multi-layered manner and groups documents into clusters on each individual layer. The clusters can be manually identified and labeled by the user, who can then employ additional exploration functions to visualize relations among documents and document clusters. A detailed description of the individual parts of the visualization system follows.

7.2.1 Document Sources

A variety of electronic document databases can be used as data sources for the toolbox, such as patent services and citation services [Morris et al. 02]. In this dissertation, journal papers available from the Science Citation Index (SCI) are used as a major data source. The SCI provides access to scientific literature published in the

physical, biological and medical fields. More information about the SCI can be found in Chapter VI. For scientific literature in other technical domains, web-based services such as Citeseer [Lawrence et al. 99] or IEEE Xplore (<http://www.ieee.org/ieeexplore>) can be used.

Once the SCI has been selected as the data source, it can be queried using general keywords to extract only relevant documents of interest to the user. Results of the querying process will then be downloaded and saved as plain text files. A typical record of a journal paper in text format is shown in Figure VII.3, which is one of the papers retrieved by using the query term ‘Self Organizing Maps.’ Author, institute, keywords, abstract text, and other information are printed in tagged format, i.e. field labels precede each piece of information. Citations of the papers are available as well, which are listed in the field labeled with *CR*. They provide explicit links among the papers.



```

wos3 - Notepad
File Edit Format View Help
PT J
AU Anouar, F
  Badran, F
  Thiria, S
TI Probabilistic self-organizing map and radial basis function networks
SO NEUROCOMPUTING
DE self-organizing map; dynamic clusters; likelihood; radial basis
  function; regression
ID EM ALGORITHM; CONVERGENCE
CR BENAIM M, 1994, NEURAL COMPUT, V6
  BISHOP C, GTM PRINCIPLED ALTER
  BISHOP C, 1995, NEURAL NETWORKS PATT
  CELEUX G, 1984, RECONNAISSANCE MELAN
  DEMPSTER AP, 1977, J ROY STAT SOC B MET, V39, P1
  DIDAY E, 1976, DIGITAL PATTERN RECO
  FRITZKE B, 1994, NEURAL NETWORKS, V7, P1441
  HARTMAN E, 1990, NEURAL COMPUT, V2
  KOHONEN T, 1994, SELF ORGANIZING MAPS
  LUTTREL SP, 1994, NEURAL COMPUT, V6
  MOODY J, 1989, NEURAL COMPUT, V1
  NIYOGI P, 1996, NEURAL COMPUT, V8, P819
  NOWLAN S, 1990, P NEURAL INFORM PROC, P574]
  SANCHEZ VD, 1995, NEUROCOMPUTING, V9, P85
  SCHROEDER A, 1976, RSA, V24
  XU L, 1994, NEURAL NETWORKS, V7, P609
  XU L, 1996, NEURAL COMPUT, V8, P129
NR 17
TC 8
J9 NEUROCOMPUTING
JI Neurocomputing
PD AUG
PY 1998
VL 20
IS 1-3
BP 83
EP 96
SC COMPUTER SCIENCE, ARTIFICIAL INTELLIGENCE
UT ISI:000076039200008
ER

```

Figure VII.3 A typical journal paper printed in a text file

7.2.2 Project Database

The downloaded documents, as shown in Figure VII.3, are in unstructured text format. To analyze them, it is necessary to convert the unstructured text-based literature into a structured database format, such as one readable by MS Access. The project database is created by loading documents from source files using procedures written in MS Access Visual Basic. Figure VII.4 is a snapshot of the project database, in which documents are stored in a tabular format. The toolbox accesses these stored documents using SQL queries.

Number	Type	Title	Date	Inventor	Assignee	Class	Citation
04243481	0	Sizing compositions	1/6/81	Dumas; David H.	Hercules Incorporated	162/158	19601100, 19630400, 19660400, 19691200, 19720
04243482	0	Forming paper using a curved fin to fa	1/6/81	Seppanen; Erkki O.		162/202	19250500, 19310100, 19320100, 19330700, 1934C
04243483	0	Head box for a paper making machine	1/6/81	Schiel; Christian, Schmalstieg;	J. M. Voith GmbH	162/343	19700500, 19780100, 19781100, 19790100, 1979C
04243484	0	Method and apparatus for use to exch	1/6/81	Tsuji; Teruaki, Watanabe; Shige	Tokyo Shibaura Denki Kabushiki K.	176/30	19650400, 19650500, 19650900, 19660500, 1971C
04243485	0	Recirculating drainage channel for the	1/6/81	Chabin; Michel	Societe Franco-Americaine de Con	176/37	18990700, 19650200, 19650600, 19651000, 1977C
04243486	0	Method of mounting filter elements an	1/6/81	Neumann; Gerhard M., Karelin;	Delbag-Luftfilter GmbH	176/37	19520700, 19660400, 19671000, 19710100, 1972C
04243487	0	Gas-cooled high temperature nuclear	1/6/81	Schweiger; Fritz	Hochtemperatur-Kernkraftwerk Gmi	176/38	19381000, 19600300, 19640300, 19681100, 1969C
04243488	0	Coke compositions and process for m	1/6/81	Sugimura; Hidehiko, Koba; Kei	Mitsui Coke Co., Ltd.	201/6	19340300, 19510600, 19640200, 19640700, 1966C
04243489	0	Pyrolysis reactor and fluidized bed co	1/6/81	Green; Norman W.	Occidental Petroleum Corp.	201/12	19340400, 19520100, 19520800, 19551100, 1956C
04243490	0	Radial cutter type cleaning apparatus	1/6/81	Tsuzuki; Akira, Miura, Atsushi	Konitsu Kikai Kogyo Company Limi	202/241	19740700, 19761100, 19780600, 19790100, 1979C

Figure VII.4 Documents are saved in tabular format in the project database.

7.2.3 Similarity Matrix

The similarity matrix describes the magnitude of links between all pairs of documents in the database. As discussed in Chapter IV, bibliographic coupling, which provides links between documents citing common references, is used to build the similarity matrix in our approach. This is implemented by using a SQL query that counts the number of total bibliographic couplings for each pair of documents in the project database.

Finally, inter-document similarities are calculated as the normalized bibliographic coupling counts between each pair of documents, using Equation (V.16). A more detailed discussion of the formation of the similarity matrix can be found in Section 5.3.2.

7.2.4 Mapping of Documents

Once similarity values are obtained for the documents, a GHSOM is trained and used to map the documents onto a hierarchy of rectangular planar surfaces. In practice, the method starts by training the GHSOM with the rows of the similarity matrix. Starting from a small single top-layer SOM, the GHSOM grows both in width and in depth to represent data at a certain level of detail. This enables a large-to-small-scale presentation of the conceptual structure of the document collection. As a result, traversing the hierarchical structure from top down will provide a zoomed-in view of the document collection.

Following the training of the GHSOM, the Ranked Centroid Projection is applied to each individual map in the hierarchy to locate the coordinates of the documents relative to each other on the corresponding two-dimensional plane.

7.2.5 Clustering of Documents

Once the above steps are successfully accomplished, it is ready to generate cluster maps, which are visual representations of the documents constituting the database. The toolbox allows the user to do all clustering manually by identifying interesting groups of documents on the map display. Lists of documents in each cluster are stored in the project database as the user identifies them. Most clusters on the map correspond to specific

technological fields or sub-fields in the document set. Document titles from each cluster can be examined manually for common themes in order to derive labels for each cluster.

7.2.6 Visual Exploration

Further exploration and analysis of the documents can be carried out using the different functions and tools available in the toolbox. Each cluster of documents can be selected and analyzed individually. Some of the exploration approaches include analyzing most frequent terms, authors, institution sources, links between cluster groups and the like.

An enhanced visual representation, as the one shown in Figure VI.6, can be created by displaying the size of the document marker proportional to the number of citations the document receives. This function helps identify seminal papers in the database. Publication dates of the documents can also be visualized by varying the color of the document marker, which helps identify the chronological trend in the documents.

7.3 Getting Started with the Menu Bar

As shown in Figure VII.1, there are five drop-down menus at the top of the GUI. The user can start the document visualization and analysis process by using these menus.

7.3.1 Working with a Project

The first step in the process is to start a project. A project is a workspace involved in analyzing a particular database, including the associated setup and all information needed for generating the document maps.

Assume we have collected a number of journal papers, as those used in Section 6.5, and converted them from source files to a MS Access database named *som*. The user can set up a new project by clicking on *New Project* under the Project menu (Figure VII.5). Following this, a link between the project and the relevant database needs to be set up. This can be done by selecting the *Set Database Link* under the Project menu. A window, as shown in Figure VII.6, containing the names of all available databases will pop up next. Select the one of interest, *som* in this case, to link it to the current project.

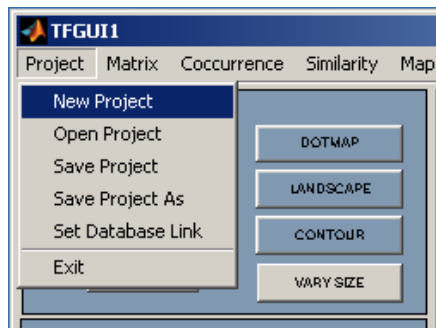


Figure VII.5 Click on *New Project* to create a new project



Figure VII.6 Selecting a database from the list

After the completion of the above steps, a project with a database link is set up. An existing project can be recalled by selecting *Open Project* under the Project menu and locating the name of the project from the directory.

7.3.2 Building Similarity Matrix

The procedure described in Section 5.3.2 is followed to build the similarity matrix. First the document-reference matrix \mathbf{M}_{dr} is loaded by selecting *Load paper to reference matrix* under the Matrix menu (Figure VII.7). Following this, the similarity matrix, as defined in Equation (V.16), can be calculated by selecting *Use bib coupling default* under the Coccurrence menu (Figure VII.8). Next click *Bib to Diva* under the Similarity menu (Figure VII.9) to load the similarity matrix to the current project.

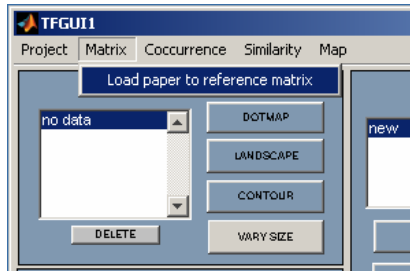


Figure VII.7 Load paper-reference matrix

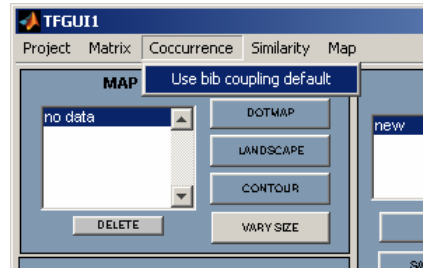


Figure VII.8 Create similarity matrix

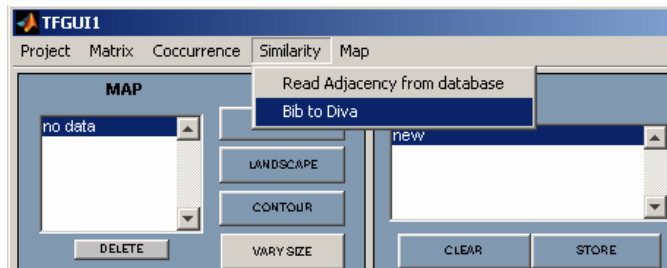


Figure VII.9 Load similarity matrix to current project

An adjacency matrix is needed later for executing the exploration functions, which represents the citation network illustrated in Figure IV.3 as a directed graph. The adjacency matrix can be generated and loaded by selecting *Read Adjacency from database* under the Similarity menu.

7.3.3 Training the GHSOM

Once the similarity matrix is created and loaded, the user can follow the steps below to start training a GHSOM neural network:

- a. Select the similarity matrix to be the training data by highlighting *similarity* in the Connections window (Figure VII.10).

- b. Select *Train GHSOM* under the Map menu (Figure VII.11) to start training. If the similarity matrix is not selected before this step, an error message as shown in Figure 12, will display.
- c. Set training parameters τ_1 and τ_2 respectively in the window popping up next and click on OK (Figure VII.13).

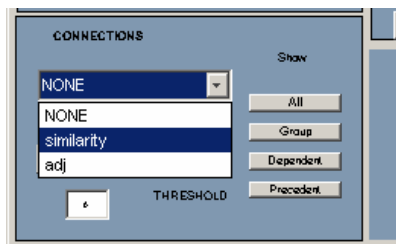


Figure VII.10 Select similarity matrix

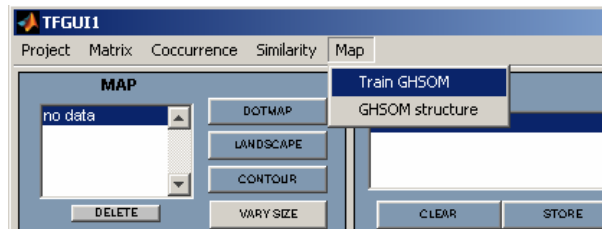


Figure VII.11 Start GHSOM training

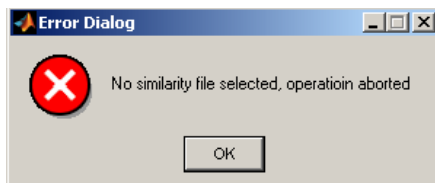


Figure VII.12 Error message

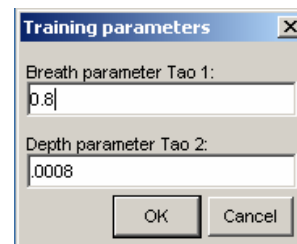


Figure VII.13 Set training parameters

When the training is completed, a two-dimensional representation of the GHSOM structure will display as shown in Figure VII.14. The first-layer ordinations are saved to the project, which will be used to generate the document map. A dialog box (Figure VII.15) will pop up asking the user to name the set of ordinations.

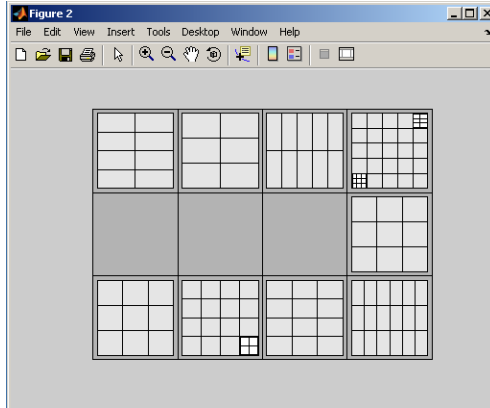


Figure VII.14 The two-dimensional representation of the GHSOM structure

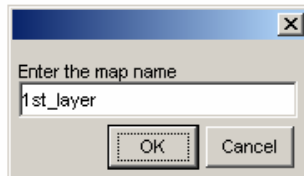


Figure VII.15 Enter the name of the map

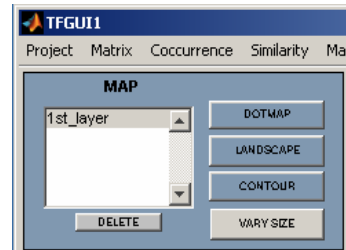


Figure VII.16 The Map control group

7.4 Displaying and Working with Maps

7.4.1 Displaying Maps

The MAP control group below the top menu bar is used for displaying maps (Figure VII.16). The map ordinations previously produced can be found in the window on the left of the control group. The user can select a map by highlighting the corresponding name and display the document map by pressing the *DOTMAP* button. Once the above steps are completed, a map as shown in Figure VII.17 will be displayed.

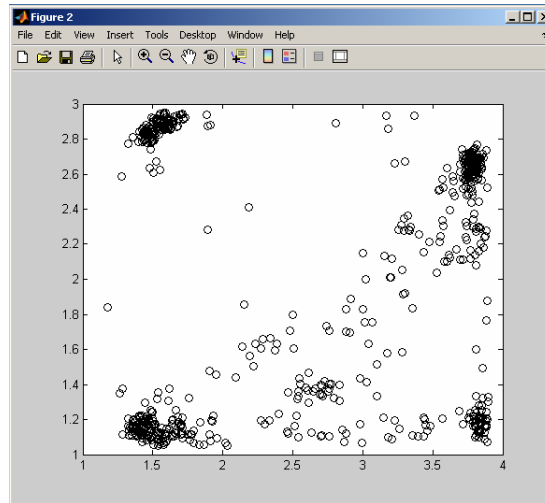
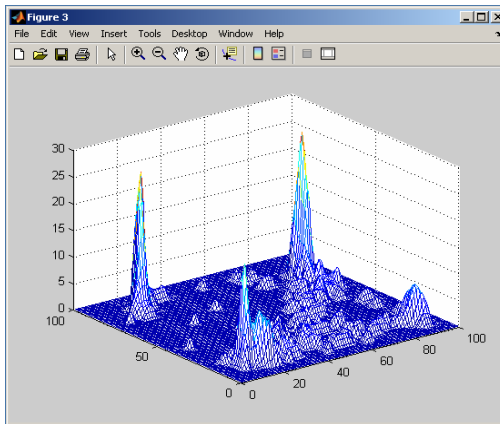
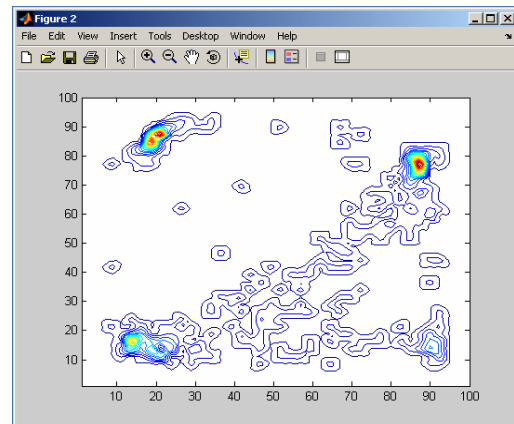


Figure VII.17 The dot map for the first-layer map



(a)



(b)

Figure VII.18 (a) Landscape map (b) Contour map

The map in Figure VII.17 is usually called a dot map, as the documents are shown as small circles, or dots, on a rectangular area. The dot map is the most convenient and common map representation used for exploration. However, when many documents are mapped on top of each other, it is difficult to gauge the density of documents on the map. Two alternate map formats, landscapes and contour maps, are also available, which can be generated by clicking on the corresponding buttons on the right of the control group.

Figure VII.18(a) shows the landscape map for the same data plot, in which the height of the surface above plane is proportional to the density of documents at that position. Figure VII.18(b) shows the contour map for the same data plot, which offers another way of visualizing document densities on the map.

7.4.2 Exploring Maps

In addition to the Map control group, there are three other sub-windows in the toolbox GUI: Time, Clusters, and Connections (see Figure VII.1). These control groups are designed for specific exploration functions, which will be discussed respective in the following.

7.4.2.1 Clustering and Marking Documents

The CLUSTERS control group, as shown in Figure VII.19, is used to identify, display and mark clusters of documents on dot maps. On a dot map, clusters can be created by dragging the mouse to draw a rectangle over the group(s) of dots of interest. Upon selection, the documents inside the rectangle will be highlighted (Figure VII.20).

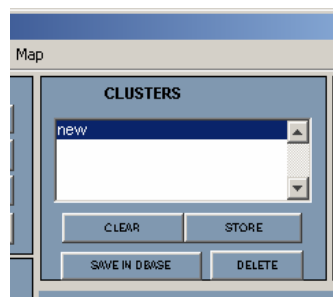


Figure VII.19 The CLUSTERS control group

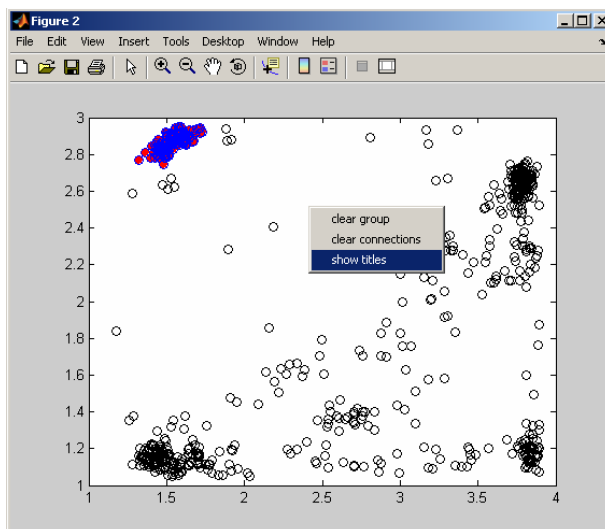


Figure VII.20 The selected cluster is highlighted.

The user can label the cluster by browsing the document titles for a common topic. Upon right click on the map, a menu pops up (Figure VII.20). Selecting the *show titles* option will enable a window with all the titles of the selected group listed (Figure VII.21).

Author	Title
Altman, RB	Whole-genome expression analysis: challenges beyond clustering
Banerjee, N	Functional genomics as applied to mapping transcription regulatory networks
Bensmail, H	Postgenomics: Proteomics and bioinformatics in cancer research
Bicciato, S	Pattern identification and classification in gene expression data using an autoassociative neural network model
Bilke, S	Probabilistic estimation of microarray data reliability and underlying gene expression
Brown, MPS	Knowledge-based analysis of microarray gene expression data by using support vector machines
Brunet, JP	Metagenes and molecular pattern discovery using matrix factorization
Burke, HE	Discovering patterns in microarray data

Figure VII.21 Titles of the highlighted documents

Examining the paper titles in Figure VII.21, we find microarray data analysis is the common subject, using which we can label the selected cluster. To save this cluster for later use, press the *STORE* button and name the cluster in the window popping up

next (Figure VII.22). The name of the cluster just stored will appear in the cluster window (Figure VII.23).

The *CLEAR* button is used to clear a group selection. The user can then make a new selection and define it under a previously named cluster. The *DELETE* button is used to delete a stored cluster. Pressing the *SAVE IN DBASE* button saves the selected group of documents in the MS Access database that the current project is linked to.

Using Matlab figure editor, the user can also directly label each cluster on the dot map. A labeled map is shown in Figure VII.24.

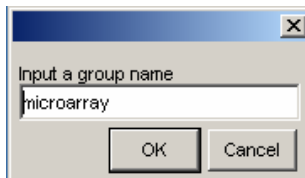


Figure VII.22 Enter the cluster name

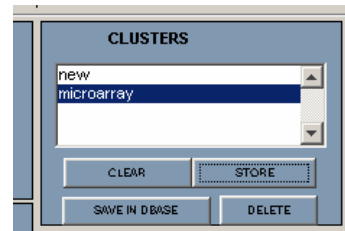


Figure VII.23 The cluster name appears in the window

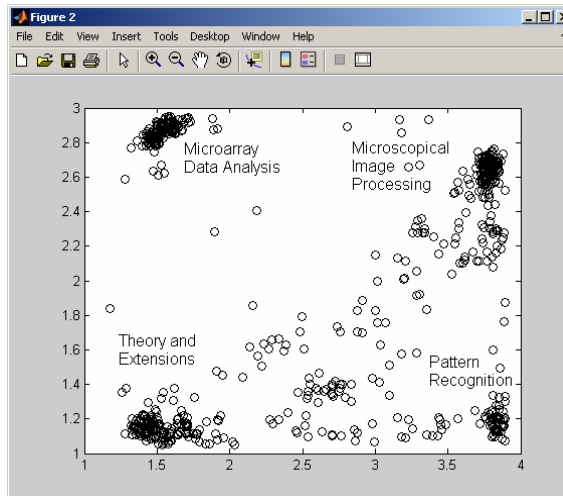


Figure VII.24 Labeled map

7.4.2.2 Visualization of Document Links

Another important exploration function available in the toolbox is the option of displaying links among individual documents and document clusters. A pair of documents is linked if they have a non-zero similarity value. These links convey information vital to understand the inherent relationship in the data set. The CONNECTIONS window (Figure VII.25) is used to generate and manipulate connections.

The user can limit the display of links by setting the threshold slider bar. Only links corresponding to similarity values above a threshold will be displayed. Before displaying the links, the desired type of links, similarity matrix or adjacency matrix, must be selected from the pull-down menu (Figure VII.26).

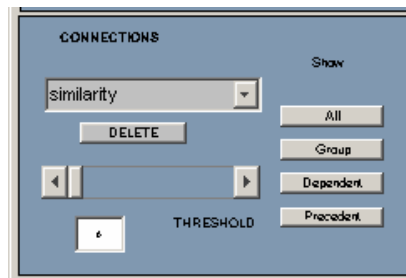


Figure VII.25 The CONNECTIONS window

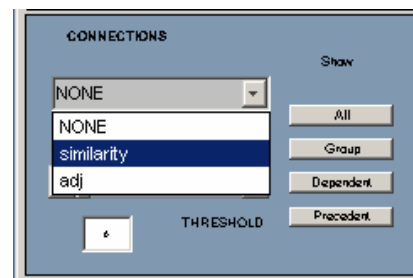


Figure VII.26 Select the desired link

Four types of document links can be visualized by pressing the four buttons on the right of the window. The *All* button enables the display of all similarity connections above the threshold value (Figure VII.27(a)). The *Group* button enables the display of only links connecting to user selected documents and clusters. Figure VII.27(b) shows the citation links that connect to the selected cluster in the lower right corner of the map. The *Dependent* button is used to visualize all documents citing the selected documents both

directly and indirectly (Figure VII.27(c)). The *Precedent* button visualizes the opposite citation relation (Figure VII.27(d)).

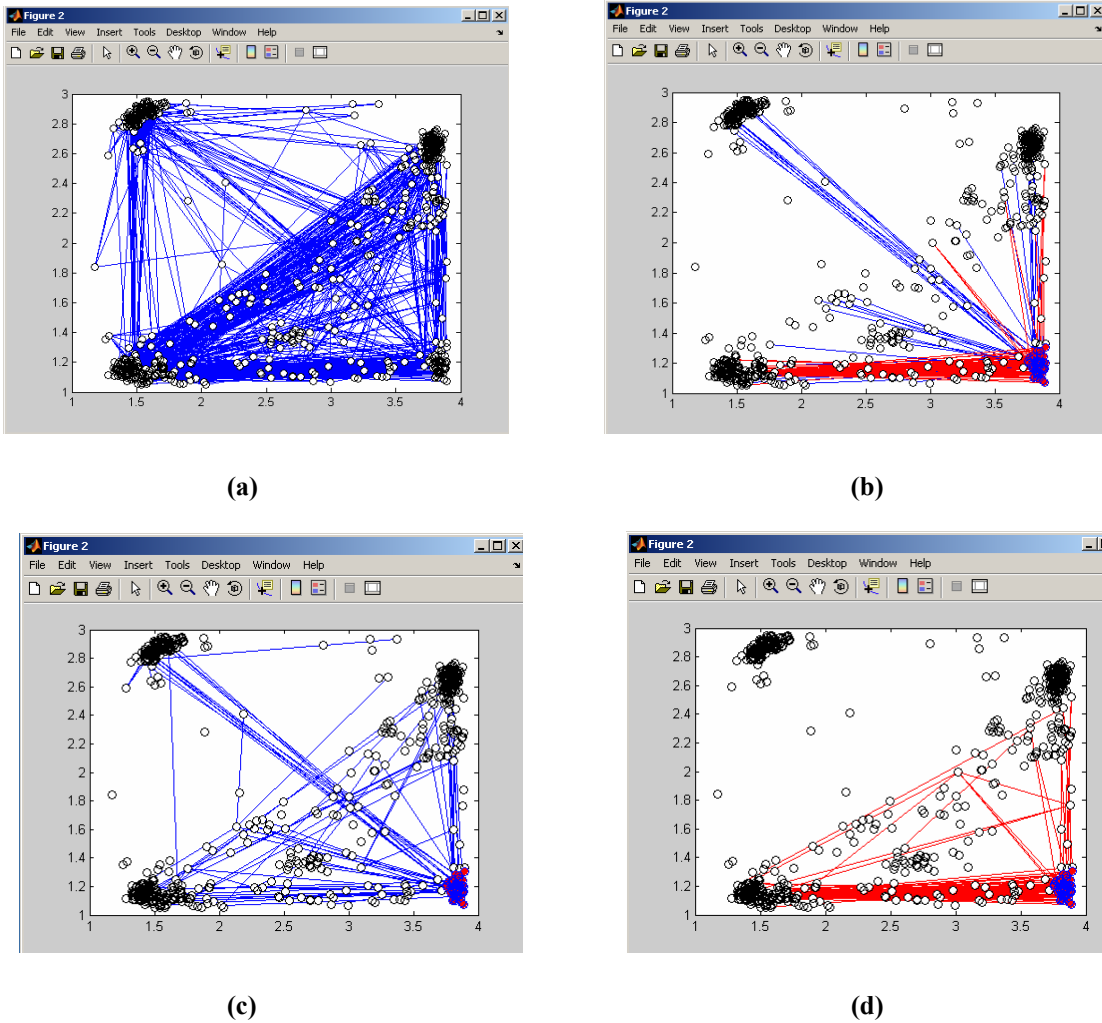


Figure VII.27 Different types of connections: (a) All, (b) Group, (c) Dependent, (d) Precedent.

7.4.2.3 Visualization of Document Dates

Visualization of the publication date is very useful for investigating the trends in document citation patterns and cluster sizes. The TIME control group (Figure VII.28) provides functions to visualize date information by highlighting documents on a dot map within a specific time span.

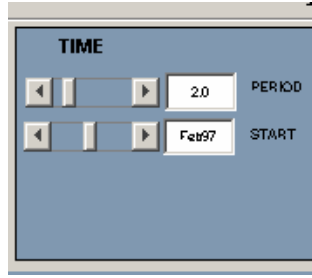


Figure VII.28 The TIME control window

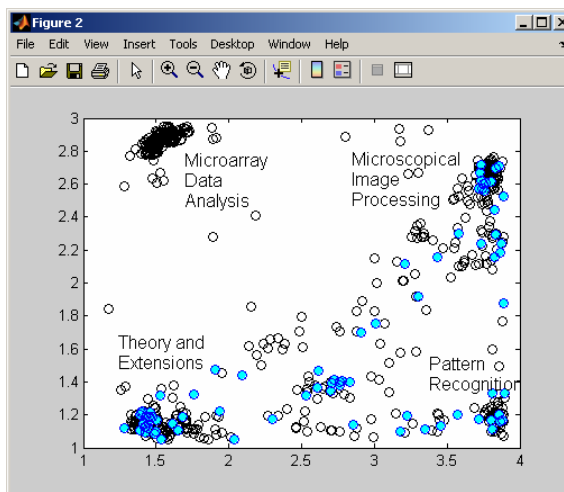


Figure VII.29 Documents published within a specific time period are highlighted

The bottom slider bar in the control window is used to set the beginning date desired for display and the top slider bar is used to set the period or span of time needed in the time stratification visualization. For instant, if the user needs to analyze documents published in two years starting from February 1997, the starting date should be set at “Feb. 97” and 2 should be specified for the period (Figure VII.28). Figure VII.29 shows the result, in which the documents published within the above time period are highlighted. From the figure we can find that no papers were published on microarray data analysis during this time period. Setting this function over specific adjacent time periods, such as every two years, helps visualize activity in the document set.

7.4.2.4 Visualization of Document Citation Counts

One of the most useful functions provided by the toolbox is to the visualization the document citation counts. This function can be executed by pressing the *Vary Size* in the MAP control group (Figure VII.16). The resulting map is shown in Figure VII.30, in which the size of the document marker is proportional to the number of times a document has been cited. This operation causes important papers, as judged from large citation counts, to stand out on the document map as shown in Figure VII.30. A labeled version of this map can be found in Figure VI.7.

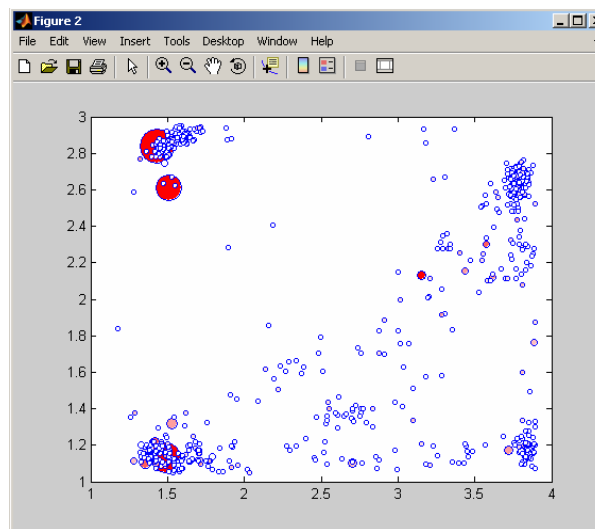


Figure VII.30 A dot map with document citation counts visualized

CHAPTER VIII

CONCLUSION AND FUTURE WORK

8.1 Summary of the Work

With the rapid growth in the production and availability of textual data, algorithms to explore it, organize it, and extract knowledge from it, are in great need. This study introduces an approach for clustering and visualizing high-dimensional data, especially textual data. The devised approach, which is an extension of the Self-Organizing Map, acts both as an analysis tool as well as a direct interface to the data, making it a very useful tool for processing textual data. It clusters documents and presents them on a two-dimensional display space. Documents covering similar topics are grouped into the same cluster and clusters with similar concepts are located nearby on a map.

In the training phase, the proposed approach employs a GHSOM architecture, which grows both in depth according to the data distribution, allowing a hierarchical decomposition and navigation in portions of the data, and in width, implying that the size of each individual map adapts itself to the requirements of the input space. After convergence of the training process, a novel projection method, the Ranked Centroid Projection, is used to map the input vectors to the hierarchy of two-dimensional output

maps of the GHSOM. The performance of the presented approach has been illustrated using one artificial data set and two real-world document collections. The two document collections are scientific journal articles on two different subjects obtained from the Science Citation Index. The document representation relies on a citation-based model that depicts the inter-document similarities using the bibliographic coupling counts between all pairs of documents in the collection.

Although the resulting SOM maps are graphical artifacts, the simulation results have demonstrated that the approach is highly effective in producing fairly detailed and objective visualizations that are easy to understand. These maps therefore have the potential of providing insights into the information hidden in a large collection of documents. In addition, for the given document collections, it is rather easy to judge the quality of the clustering result.

8.2 Advantages and Limitations

The proposed clustering and visualization approach is based on the SOM model and incorporates several unique features, such as the growing hierarchical structure, the ranked centroid projection of the data vectors, and the incremental clustering for dynamic databases. The advantages of the presented approach for processing document data are as follows:

- 1) The GHSOM structure employed by the presented approach overcomes two major limitations of the standard SOM: the static architecture of the SOM and its poor capability in representing hierarchical relations of the data.

- 2) The novel projection method, the RCP, involves a ranking scheme based on a membership degree factor. This enables the proposed projection technique not only to reveal the clustering tendency in the data but also to visualize information on cluster memberships.
- 3) The RCP allows the data points to be projected to arbitrary locations across the SOM network. It is therefore possible to handle a large data set with a rather small number of nodes while providing a high-resolution map at the mean time.
- 4) The number of nodes required to represent a set of data is less than that of a comparable SOM. This will result in a lower computation load and hence less processing time.
- 5) The RCP may be used to incrementally process new data points without affecting the original map topology.

Advantages 1) and 5) become especially useful when dealing with document databases, which often process a hierarchical and dynamic nature. The size of the network is also an important issue, as it will directly affect the processing time, especially in cases of large data sets. Due to the dynamic structure it adopts, the proposed approach achieves the same level of detail with a lesser number of nodes, which is a critical advantage in mapping large data sets. In addition, the flexible mapping of the RCP provides visualizations with satisfactory resolutions using a comparably small number of nodes. Therefore, it is evident that the proposed approach is a very useful tool in knowledge discovery tasks.

Several limitations also exist in the ability of the presented approach to depict the document databases. The spatial representation of documents provides only an approximate view of the documents of interest. Such spatial representations are sometimes criticized as being lack of validity and reliability. Particularly, it is because many graphical representations of the same set of data are possible. A decision should be taken for choosing the best spatial representation for the given data set. In order to support such a decision, performance metrics for measuring the quality of a spatial representation are needed. However, the construction of such performance metrics is especially difficult provided that, in the real world data, the underlying data distribution is usually unknown or there is no certainty about a clustering structure. As a result, the evaluation of a data projection is very difficult. Moreover, further attempts to interpret the findings or to develop new ideas on the basis of the spatial data may require the involvement of domain experts.

Although the above limitations exist, a spatial representation may still be the best initial approach in a theory-deficient area like quantitative studies of the underlying conceptual structure of document collections. In such cases, the most important contribution of these maps is the detection of the previously invisible structure.

8.3 Future Directions

The introduction (Chapter I) outlined a large research field, which can not possibly be completely covered in a single study. The main results presented here may be characterized as an extension to the use of the SOM in document clustering and

visualization and the development of the presented projection method. Many issues outlined in the introduction have been addressed in the contents of this study. However, some concerning problems still await to be attended, which are summarized in the following.

The incremental clustering method is based on the assumption that it is possible to consider new documents one at a time and assign them to the existing clusters. As time progresses, new documents are added while old documents may also become obsolete. The same is true for the references. This implies that an incremental clustering algorithm for a dynamic database should be able to handle both additions of new data points and deletions of obsolete data points.

Another possibility is the development of suitable performance metrics for the various tasks described in this report, as well as the application of such principles in evaluating document maps and text visualization models in general. So far, two quantitative measures, the DB index and Sammon's stress, have been used in this study to characterize the resulting projection. However, these two factors are inadequate to be used as sole performance metrics for assessing the document map, where the underlying data structure is complicated and mostly unknown. Some new measure(s) would be needed to accommodate the characteristics of the document map.

Last but not the least, a continuing challenge is to demonstrate the usage of the proposed approach in different large-scale real-world applications. In this report, the proposed approach acts as an excellent analysis tool for investigating the inner structure of the transformed textual data, providing the capability of clustering and visualizing the document collection of interest. Yet, there are application domains other than textual

information to be explored, where the proposed approach can provide a unique value. It should be particularly beneficial in dealing with large-scale data, hierarchical data or/and dynamic data.

REFERENCES

- [Alahakoon et al. 00] D. Alahakoon, S. K. Halgarmuge, and B. Srinivasan, "Dynamic self-organizing maps with controlled growth for knowledge discovery," *IEEE Transactions on Neural Networks*, vol. 11, pp. 601-614, 2000.
- [Bhatnagar and Batra 01] R. Bhatnagar and S. Batra, "Anthrax toxin," *Critical Reviews in Microbiology*, vol. 27, no. 3, pp. 167-200, 2001.
- [Bienfait 94] B. Bienfait, "Applications of high-resolution self-organizing maps to retrosynthetic and QSAR analysis," *Journal of Chemical Information and Computer Sciences*, vol. 34, no. 4, pp. 890-898, 1994.
- [Blackmore and Miikkulainen 93] J. Blackmore and R. Miikkulainen, "Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map," *Proceedings of the IEEE International Conference on Neural Networks (ICNN'93)*, vol. 1, San Francisco, CA, pp. 450-455, 1993.
- [Newman et al. 98] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, UCI Repository of machine learning databases [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.
- [Booker et al. 99] A. Booker, M. Condliff, M. Greaves, F. B. Holt, A. Kao, D. J. Pierce, S. Poteet, and Y. J. Wu, "Visualizing text data sets," *IEEE Computing in Science & Engineering*, Vol. 1, no. 4, pp. 26-35, 1999.
- [Bouton and Pagés 93] C. Bouton and G. Pagés, "Self-organization and a.s. convergence of the one-dimensional kohonen algorithm with non-uniformly distributed stimuli," *Stochastic Processes and Their Applications*, vol. 47, pp. 249-274, 1993
- [Chan and Pampalk 02] A. Chan and E. Pampalk, "Growing Hierarchical Self-Organizing Map (GHSOM) Toolbox: visualizations and enhancements," *Proceedings of the 9th International Conference on Neural Information Processing*, vol. 5, pp. 2537-2541, 2002.
- [Cherkassky and Lari-Najafi 91] V. Cherkassky and H. Lari-Najafi, "Constrained topological mapping for nonparametric regression analysis," *Neural Networks*, vol. 4, pp. 27-40, 1991.

- [Cottrell and Fort 87] M. Cottrell and J.-C. Fort, Étude d'un processus d'auto-organisation. *Annales de l'Institut Henri Poincaré*, vol. 23, no.1, pp. 1-20, 1987.
- [Davies and Bouldin 79] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 4, pp. 224-227, 1979.
- [Davison 83] M. L. Davison, *Multidimensional Scaling*. New York, NY, John Wiley and Sons, 1983.
- [Deerwester et al. 90] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas and R. A. Harshman. "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391-407, 1990.
- [Demuth 98] H. Demuth, and M. Beale, *Neural Network Toolbox User's Guide*, the Mathworks, Inc., 1998.
- [Erwin et al. 91] E. Erwin, K. Obermayer, and K. Schulten, "Convergence properties of self-organizing maps," in Kohonen, T., Mäkisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks*, pp. 409-414, Amsterdam, Netherlands, Elsevier, 1991.
- [Erwin et al. 92a] E. Erwin, K. Obermayer, and K. Schulten, "Self-organizing maps: Ordering, convergence properties and energy functions," *Biological Cybernetics*, vol. 67, no. 1, pp. 47-55, 1992.
- [Erwin et al. 92b] E. Erwin, K. Obermayer, and K. Schulten, "Self-organizing maps: Stationary states, metastability and convergence rate," *Biological Cybernetics*, vol. 67, no. 1, pp. 35-45, 1992.
- [Fisher 36] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annual Eugenics*, vol. 7, pp. 178-188, 1936.
- [Fritzke 94] B. Fritzke, "Growing cell structures - a self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, pp. 1441-1460, 1994.
- [Fritzke 95a] B. Fritzke, "A growing neural gas network learns topologies," *Advances in Neural Information Processing Systems 7*, MIT press, Cambridge MA, pp. 625-632, 1995.
- [Fritzke 95b] B. Fritzke, "Growing grid - a self-organizing network with constant neighborhood range and adaption strength," *Neural Processing Letters*, vol. 2, pp. 9-13, 1995.

- [Fritzke 06] B. Fritzke, "Some competitive learning methods," Retrieved January 19 2006, from the World Wide Web.
<http://www.ki.inf.tu-dresden.de/~fritzke/JavaPaper/t.html>
- [Garfield 94] E. Garfield, "The concept of citation indexing: a unique and innovative tool for navigating the research literature," *Current Contents*, January 3, 1994.
- [Golub and Van Loan 89] G. Golub and C. Van Loan, *Matrix Computations*. Johns-Hopkins, Baltimore, Maryland, second edition, 1989.
- [Hartigan 75] J. Hartigan, *Clustering Algorithms*. New York, NY, John Wiley and Sons, 1975.
- [Hollmén 96] J. Hollmén, *Process Modeling Using the Self-Organizing Map*. M.S. thesis, Helsinki University of Technology, 1997.
- [Honkela 97] T. Honkela, *Self-Organizing Maps in Natural language Processing*. Ph.D. thesis, Helsinki University of Technology, 1997.
- [Jain and Dubes 88] A. K. Jain, and R. C. Dubes, *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- [Kaski 97] S. Kaski, *Data Exploration Using Self-Organizing Maps*. Ph.D. thesis, Helsinki University of Technology, 1997.
- [Kessler 63] M. M. Kessler, "Bibliographic coupling between scientific papers," *American Documentation*, vol. 14, pp. 10-25, 1963.
- [Kraaijveld et al. 95] M. A. Kraaijveld, J. Mao, and A. K. Jain, "A nonlinear projection method based on Kohonen's topology preserving maps," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 548-559, 1995.
- [Kohonen 82] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59- 69, 1982.
- [Kohonen 90] T. Kohonen, "The self-organizing map," *Proceedings of IEEE*, vol. 78, no. 9, pp. 1464-1480, 1990.
- [Kohonen 95] T. Kohonen, *Self-Organizing Maps*. Berlin, Germany, Springer-Verlag, 1995.
- [Kohonen et al. 00] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela, "Self Organization of a Massive Document Collection," *IEEE Transactions on Neural Networks*, Special Issue on Neural Networks for Data Mining and Knowledge Discovery, vol. 11, no. 3, pp. 574-585, 2000.

- [Lampinen and Oja 92] J. Lampinen and E. Oja, "Clustering properties of hierarchical self-organizing maps," *Journal of Mathematical Imaging and Vision*, vol. 2., pp. 261-272, 1992.
- [Lang and Warwick 02] R. Lang and K. Warwick. "The plastic self organising maps", *Proceedings of International Joint Conference on Neural Networks*, vol. 1, pp. 727 – 732, 2002.
- [Lampinen and Oja 92] J. Lampinen and E. Oja, "Clustering Properties of Hierarchical Self-Organizing Maps," *Journal of Mathematical Imaging and Vision*, Vol. 2, pp. 261-272, 1992.
- [Lawrence et al. 99] S. Lawrence, C. L. Giles, and K. Bollacker, "Digital libraries and autonomous citationindexing," *IEEE Computer*, vol. 32, no. 6, pp. 67-71, 1999.
- [Mao and Jain 95] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Transactions on Neural Networks*, Vol. 6, no. 2, pp. 296-317, 1995.
- [Marsland et al. 02] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *IEEE Transactions on Neural Networks*, Vol. 15, pp. 1041-1058, 2002.
- [Martinetz and Schulten 91] M. Martinetz and K. J. Schulten, "A "neural-gas" network learns topologies," *Artificial Neural Networks*, K. M. T. Kohonen, O. Simula, and J. Kangas, Ed. Amsterdam: North-Holland, pp. 397-402, 1991.
- [Merkl and Rauber 97] D. Merkl and A. Rauber, "Alternative ways for cluster visualization in self-organizing maps," *Proceedings of Workshop on Self-Organizing Maps (WSOM'97)*, Espoo, Finland, pp. 106-111, 1997.
- [Merkl and Rauber 98] D. Merkl and A. Rauber, "CIA's view of the world and what neural networks learn from it: A comparison of geographical document space representation metaphors," *Proceedings of the 9th International Conference on Database and Expert Systems Applications*, Vienna, Austria, pp. 816-825, 1998.
- [Miikkulainen 90] R. Miikkulainen, "Script recognition with hierarchical feature maps." *Connection Science*, vol. 2, pp. 83-101, 1990.
- [Morris et al. 01] S. A. Morris, Z. Wu, and G. Yen, "A SOM mapping technique for visualizing documents in a database," *Proceedings of the IEEE International Conference on Neural Networks*, Washington DC, USA, pp. 1914-1919, 2001.
- [Morris et al. 02] S. A. Morris, C. Deyong, Z. Wu, S. Salman, and D. Yemenu, "DIVA: a visualization system for exploring document databases for technology forecasting," *Computer and Industrial Engineering*, vol. 43, pp. 841-862, 2002.

- [Morris et al. 03] S. A. Morris, G. Yen, Z. Wu, and B. Asnake, "Timeline visualization of research fronts," *Journal of the American Society for Information Science and Technology*, vol. 55, no. 5, 413-422, 2003.
- [Morris 05] S. A. Morris, *Unified Mathematical Treatment of Complex Cascaded Bipartite Networks: the Case of Collections of Journal Papers*, Ph.D. dissertation, Oklahoma State University, 2005.
- [Mulier and Cherkassky 94] F. Mulier and V. Cherkassky, "Learning rate schedules for self-organizing maps," *Proceedings of the 12th International Conference on Pattern Recognition*, vol. II, pp. 224-228, Jerusalem, Israel, 1994.
- [Noel et al. 02] S. Noel, V. Raghavan, C.-H. H. Chu, "Document Clustering, Visualization, and Retrieval via Link Mining," *Clustering and Information Retrieval*, W. Wu, H. Xiong, and S. Shekhar (eds.), Kluwer Academic Publisher, 2002.
- [Olsen et al. 93] K. A. Olsen, R. R. Korfhage, K. M. Sochats, M. B. Spring, and J. G. Williams, "Visualization of a document collection with implicit and explicit links," *Scandinavian Journal of Information Systems*, vol. 5, pp. 79-95, 1993.
- [Olsen 05] K. A. Olsen, "Data visualization," *BookRags*. Retrieved November 2 2005, from the World Wide Web.
<http://www.bookrags.com/sciences/computerscience/data-visualization-csci-03.html>
- [Osareh 96] F. Osareh, "Bibliometrics, citation analysis and co-citation analysis: a review of literature I," *Libri*, vol. 46, no. 3, pp. 149-158, 1996.
- [Pampalk et al. 02] E. Pampalk, A. Rauber, and D. Merkl, "Using smoothed data histograms for cluster visualization in self-organizing maps," *Proceedings of the International Conference on Artificial Neural Network*, vol. 2415, pp. 871-876, Madrid, Spain, 2002.
- [Pözlbauer et al. 05] G. Pözlbauer, M. Dittenbach, and A. Rauber, "A visualization technique for Self-Organizing maps with vector fields to obtain the cluster structure at desired levels of detail," *Proceedings of the International Joint Conference on Neural Network*, Montreal, Canada, pp. 1558-1563, 2005.
- [Rauber et al. 02] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1331 – 1341, 2002.
- [Ridder and Duin 97] D. de Ridder and R. P. W. Duin, "Sammon's mapping using neural networks: a comparison," *Pattern Recognition Letters*, vol. 18, pp. 1307-1316, 1997.

- [Ritter et al. 92] H. J. Ritter, T. M. Martinetz, and K. J. Schulten, *Neural Computation and Self-Organizing Maps*. Addison-Wesley, Reading, MA, 1992.
- [Salton et al. 75] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, Vol. 18, No. 11, pp. 613-620, 1975.
- [Salton 89] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, MA, 1989.
- [Sammon 69] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 18, pp. 401-409, 1969.
- [Simula et al. 98] O. Simula, J. Vesanto, and P. Vasara, "Analysis of industrial systems using the self-organizing map," *Proceedings of Second International Conference on Knowledge-Based Intelligent Electronic Systems*, vol. 1, pp. 61-68, April 1998.
- [Small 97] H. Small, "Update on science mapping: creating large document spaces," *Scientometrics*, vol. 38, no. 2, pp. 275-93.
- [Small 99] H. Small, "Visualizing science by citation mapping," *Journal of the American Society for Information Science*, vol. 50, no. 9, pp. 799-813, 1999.
- [Tamayo et al. 99] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences of the USA*, vol. 96, pp. 2907-2912, 1999.
- [Tijssen 92] R. J. W. Tijssen, *Cartography of Science: Scientometric Mapping with Multidimensional Scaling Methods*, DSWO Press, Leiden University, Leiden, the Netherlands, 1992.
- [Toronen et al. 99] P. Toronen, M. Kolehmainen, G. Wong, and E. Castren, "Analysis of gene expression data using self-organizing maps", *FEBS Letters*, vol. 451, no. 2, pp. 142-146, 1999.
- [Ultsch 03a] A. Ultsch, "Maps for the Visualization of high-dimensional Data Spaces," *Proceedings of Workshop on Self-Organizing Maps (WSOM 2003)*, Kyushu, Japan, pp. 225-230, 2003.
- [Ultsch 03b] A. Ultsch, "U*-Matrix: a Tool to visualize Clusters in high dimensional Data," *Technical Report No. 36*, Dept. of Mathematics and Computer Science, University of Marburg, Germany, 2003.

- [Ultsch and Mörchen 05] A. Ultsch, and F. Mörchen, "ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM," Technical Report, Dept. of Mathematics and Computer Science, University of Marburg, Germany, no. 46, 2005.
- [Ultsch and Siemon 90] A. Ultsch and H. P. Siemon, "Kohonen's self-organizing feature maps for exploratory data analysis," *Proceedings of the International Neural Network Conference (INNC'90)*, Dordrecht, Netherlands, pp. 305-308, 1990.
- [Vesanto 97] J. Vesanto, *Data Mining Techniques Based on the Self-Organizing Map*. M.S. thesis, Helsinki University of Technology, 1997.
- [Vesanto and Alhoniemi 00] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 11, pp. 586-600, 2000.
- [Vesanto et al. 00] J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas, "Self-organizing map in Matlab: the SOM toolbox," *Proceedings of the Matlab DSP Conference*, pp. 35-40, Espoo, Finland, 1999.
- [Vicente and Vellido 04] D. Vicente and A. Vellido, "Review of Hierarchical Models for Data Clustering and Visualization," In R. Giráldez, J.C. Riquelme, J.S. Aguilar-Ruiz (Eds.) *Tendencias de la Minería de Datos en España*, Red Española de Minería de Datos, 2004.
- [Kim and Weck 05] I. Y. Kim and O. de Weck, "Adaptive Weighted Sum Method for Bi-objective Optimization", *Structural and Multidisciplinary Optimization*, vol. 29, pp. 149 - 158, 2005.
- [Willshaw and von der Malsburg 76] D. J. Willshaw and C. von der Malsburg, "How patterned neural connections can be set up by self-organization," *Proceedings of the Royal Society London*, vol. B194, pp. 431-445, 1976.
- [Wu and Yen 03] Z. Wu, and G. Yen, "A SOM projection technique with the growing structure for visualizing high-dimensional data," *International Journal of Neural Systems*, Vol. 13, No. 5, pp. 353-365, 2003.
- [Yin 02] H. Yin, "ViSOM - A novel method for multivariate data projection and structure visualization," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 237-243, 2002.

VITA

Zheng Wu

Candidate for the Degree of

Doctor of Philosophy or Other

Thesis: RANKED CENTROID PROJECTION: A DATA VISUALIZATION
APPROACH BASED ON SELF-ORGANIZING MAPS

Major Field: Electrical Engineering

Biographical:

Personal Data: Born in Beijing, China, on November 8, 1975, the daughter of Junjiao Wu and Younan Yi.

Education: Graduated from Beijing Jiaotong University, Beijing, China, in July 1997 with the Degree of Bachelor of Engineering in Electrical Engineering. Completed the requirements for the Doctor of Philosophy Degree with a major in Electrical Engineering at the Electrical and Computer Engineering Department at Oklahoma State University in July 2006.

Experience: Engineer, Circuits and Systems Laboratory, Electronic Engineering Department, Tsinghua University, Beijing, China (1997-1999). Research Assistant, Electrical and Computer Engineering Department, Oklahoma State University (1999-2004). Teaching Assistant, Electrical and Computer Engineering Department, Oklahoma State University (2004-2006).

Professional Memberships: Member of the Institute of Electrical and Electronics Engineers (IEEE) since 2003; Member of the IEEE Computational Intelligence Society (CIS) since 2003.

Name: Zheng Wu

Date of Degree: July, 2006

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: RANKED CENTROID PROJECTION: A DATA VISUALIZATION
APPROACH BASED ON SELF-ORGANIZING MAPS

Pages in Study: 120

Candidate for the Degree of Doctor of Philosophy

Major Field: Electrical Engineering

The Self-Organizing Map (SOM) is an unsupervised neural network model that provides topology-preserving mapping from high-dimensional input spaces onto a commonly two-dimensional output space. In this study, the clustering and visualization capabilities of the SOM, especially in the analysis of textual data, i.e. document collections, are reviewed and further developed. A novel clustering and visualization approach based on the SOM is proposed for the task of text data mining. The proposed approach first transforms the document space into a multi-dimensional vector space by means of document encoding. Then a growing hierarchical SOM (GHSOM) is trained and used as a baseline framework, which automatically produces maps with various levels of details. Following the training of the GHSOM, a novel projection method, namely the Ranked Centroid Projection (RCP), is applied to project the input vectors onto a hierarchy of two-dimensional output maps. The projection of the input vectors is treated as a vector interpolation into a two-dimensional regular map grid. A ranking scheme is introduced to select the nearest R units around the input vector in the original data space, the positions of which will be taken into account in computing the projection coordinates.

The proposed approach can be used both as a data analysis tool and as a direct interface to the data. Its applicability has been demonstrated in this study using an illustrative data set and two real-world document clustering tasks, i.e. the SOM paper collection and the Anthrax paper collection. Based on the proposed approach, a software toolbox is designed for analyzing and visualizing document collections, which provides a user-friendly interface and several exploration and analysis functions.

The presented SOM-based approach incorporates several unique features, such as the adaptive structure, the hierarchical training, the automatic parameter adjustment and the incremental clustering. Its advantages include the ability to convey a large amount of information in a limited space with comparatively low computation load, the potential to reveal conceptual relationships among documents, and the facilitation of perceptual inferences on both inter-cluster and within-cluster relationships.

ADVISER'S APPROVAL: Dr. Gary G. Yen
