

SOM clustering for text retrieval and classification with examples on Indian scripts

Simone Marinai

Dipartimento di Sistemi e Informatica - Università di Firenze
Via S.Marta, 3 - 50139 Firenze - Italy
marinai@dsi.unifi.it

Abstract. In this paper, we discuss the use of Self Organizing Maps (SOM) for character and word clustering. The SOM is a particular kind of artificial neural network that computes an unsupervised clustering of the input data arranging the cluster centers in a lattice. After an overview of the previous applications of unsupervised learning and SOM in the field of Document Image Analysis we describe our recent work in the field. The SOM clustering can be used both for retrieval and classification applications. We describe also some preliminary investigations on the clustering of an Indian script.

1 Introduction

Supervised classifiers are key components of most Document Image Analysis systems. In supervised learning some labeled samples are used to train the classifier by reducing the sum of the costs for the training patterns. Few systems rely on unsupervised learning, or clustering, where the training algorithm takes into account unlabeled samples and there is no explicit teacher. In the latter case the system usually forms clusters of the input patterns. When specific information about the input data is available (e.g. knowledge about the nature of the patterns to be processed) then this prior knowledge can be considered in the design of the training algorithms. This information can provide a preliminary estimation of the number of clusters that are expected to be found in the training data. An appropriate choice of this parameter is crucial for improving the performance of the clustering algorithms and it is usually set by the users. Another important feature is the type of distance function to be embedded in the clustering algorithm that strongly influences the final clustering achieved.

In this chapter, we focus our attention on a particular class of clustering algorithms, the Self Organizing Map (SOM) initially proposed by Kohonen [1] that is particularly suited for dimensionality reduction and exploratory data analysis. We will analyze in particular a few applications of SOM-based clustering in some Document Image Analysis sub-tasks and the use of tangent distance in the SOM training. Moreover, we show some preliminary texts for clustering Indian scripts.

The paper is organized as follows. In Section 2 we summarize the main features of SOMs, as well as the standard training algorithm. In Section 3 we survey some recent applications of clustering algorithms in Document Image Analysis with a particular emphasis on the use of SOM. In Section 4 and in Section 5 we discuss our work related to

SOM clustering at the character and word levels. In section 6 we show some preliminary results for clustering non-latin scripts. Lastly, we present our concluding remarks in Section 7.

2 Self Organizing Maps

The Self Organizing Map is an artificial neural network that performs clustering by means of unsupervised competitive learning [1]. In the SOM the neurons are usually arranged in a two dimensional lattice (feature map). Each neuron receives inputs from the input layer and from the other neurons in the map. The input samples are described with real vectors $x(t) \in R^n$, where t is the index of the sample. Each neuron contains a model vector $m_i \in R^n$ that can be regarded as a prototype of the patterns in the cluster. During the learning, the network performs clustering and the model vectors are changed so as to reflect the similarity of neighboring clusters. The goal of the mapping is to represent the points in the source space by corresponding points in a lower dimensional target space. In particular, the training is aimed at preserving as much as possible the distance and proximity relationships among input samples.

The initial values of the model vectors, $m_i(0)$, may be selected at random or can be initialized in some orderly fashion, for instance arranging the vectors along a two-dimensional subspace spanned by the two principal eigenvectors of the input data. The two main SOM learning algorithms are the on-line and the batch ones.

The on-line algorithm computes the mapping by processing each training pattern $x(t)$ with the following steps and repeating the overall loop several times.

1. The vector $x(t)$ is compared with all the model vectors $m_i(t)$ and the *Best Matching Unit (BMU)* on the map is identified. The *BMU* is the node having the lowest distance with respect to the input pattern $x(t)$. The final topological organization of the map is heavily influenced by the distance function considered in this step. In most cases the Euclidean distance is considered, and the *BMU* $m_{b(x)}$ is identified by:

$$\|x(t) - m_{b(x)}\| = \min_i \{\|x(t) - m_i(t)\|\}. \quad (1)$$

2. The model vector of the *BMU* as well as some of its neighboring nodes are changed so as to “move” towards the current input pattern $x(t)$ according to the following equation:

$$m_i(t+1) = m_i(t) + h_{b(x),i}(t)(x(t) - m_i(t)). \quad (2)$$

where $h_{b(x),i}$ is the neighborhood function, implemented with a smoothing kernel that is time-variable and is defined over the lattice points. The neighborhood function is a decreasing function of the distance between the i -th and $b(x)$ -th models on the map grid. The extension of the kernel is also decreasing monotonically during the iterations. A widely used neighborhood function is based on the Gaussian function:

$$h_{b(x),i}(t) = \alpha(t) \exp\left(-\frac{\|r_i - r_{b(x)}\|^2}{2\sigma^2(t)}\right), \quad (3)$$

where $0 < \alpha(t) < 1$ is the learning-rate factor that decreases with the iterations, $r_i \in \mathfrak{R}^2$ and $r_{b(x)} \in \mathfrak{R}^2$ are the locations of the neurons in the lattice, and $\sigma(t)$ defines the width of the neighborhood function and is also decreasing monotonically.

The above steps are repeated until all the patterns in the training set have been processed. To achieve a better convergence towards the desired mapping it is usually required to repeat the previous loop until some convergence criteria are met. When a new loop begins the index t is set to 0 without modifying the models m_i .

One advantage of the use of SOM with respect to other clustering algorithms is the spatial organization of the feature map that is achieved after the learning process. Basically, more similar clusters are closer than more different ones. Consequently, the distance among prototypes in the output layer of the SOM can be considered as a measure of similarity between patterns in the clusters.

2.1 Tangent distance

The basic SOM training algorithm relies on the Euclidean distance to compare the patterns and the model vectors. The Euclidean distance between two patterns is in general very sensitive to small transformations of the patterns. For instance, the distance between one character and the same character subjected to a small horizontal displacement can be quite high, since many pixels in the two patterns are no longer aligned. The invariance of the training algorithm with respect to transformations of the patterns can be achieved in three ways [2]: invariance by structure, invariance by feature extraction, and invariance by training. The tangent distance can be considered an invariance by structure technique whose goal is the incorporation in the distance function of the tolerance with respect to small transformations in the pattern space.

The tangent distance principle is well described in [3] and briefly outlined in the following. Let us suppose to transform a pattern P with a non linear transformation t that is controlled by one parameter β . One example of transformation t is the rotation of the pattern by the angle β . In the pattern space the set of all the transformed patterns $S_P = \{x \mid \forall \beta x = t(P, \beta)\}$ can be considered as a one-dimensional curve that is parametrized by β . In the general case we must consider several transformations together and therefore we should consider a vector of p parameters (β) that characterizes a given combination of transformations. The patterns in S_P define a manifold that can be approximated by a plane tangent to S_P in P . The tangent plane is defined by means of a linear combination of p vectors computed by applying small independent transformations to the original pattern.

In the ideal case the combination of the p transformations describes all the possible deformations that can be applied to the patterns. In this way two objects P and Q belonging to the same class will generate the same manifolds $S_P = S_Q$ and the distance between the two manifolds will be 0.

As a matter of fact, there are practical problems to be solved. The first problem is due to the difficulty of identifying appropriate transformations that may generate actual patterns. In the case of handwriting characters, some standard transformations have been considered, such as rotation, translation, and line thickening. However, real patterns are usually subjected to transformations that are difficult to model and therefore

actual patterns are close to the manifold but are not perfectly described by it. The second problem is related to the computational cost required to evaluate the distance between two manifolds. One solution is to locally approximate the manifolds by means of the hyperplane tangent to S_P in the point P and the hyperplane tangent to S_Q in the point Q . The distance is then defined as the distance between the tangent planes:

$$TD(P, Q) = \min_{x \in T_P, y \in T_Q} \|x - y\|^2, \quad (4)$$

the equations of the tangent planes are given by:

$$\begin{aligned} T_P(\beta_P) &= P + L_P \beta_P \\ T_Q(\beta_Q) &= Q + L_Q \beta_Q \end{aligned} \quad (5)$$

where L_P and L_Q are the matrices containing the tangent vectors that are usually pre-computed. Computing the tangent distance amounts to solving a linear least squares problem as detailed, for instance, in [3]. The distance described by Eq. 4 is usually referred to as double-sided tangent distance, since we compute the distance between the two tangent planes. In some cases it would be simpler to use the one-sided tangent distance where we compute the minimum distance between one pattern and the plane tangent to the other:

$$TD1(P, Q) = \min_{x \in T_P} \|x - Q\|^2. \quad (6)$$

2.2 Tangent SOM

In this paper we propose the use of the tangent distance for computing the SOM map. One important limitation of the SOM-based clustering computed considering the Euclidean distance is the low robustness with respect to small local transformations in the pattern space that can give rise to large distances in the Euclidean space.

To partially reduce this problem we propose to use the tangent distance in the training process as described in the following. For each training pattern we first compute the tangent vectors considering the transformations that are expected to be more relevant. The computation of the tangent vectors can be quite expensive, however it is important to remark that the vectors are computed only once for each training pattern. During the training loop we use the one-sided tangent distance (Eq. 6) to find the *BMU* for each pattern, replacing Eq. (1) with:

$$\|x(t) - m_{b(x)}\| = \min_i \{TD1(x(t), m_i(t))\}. \quad (7)$$

In section 4 we describe one application of the SOM_TD training algorithm comparing its performance with respect to the standard SOM model.

3 Unsupervised learning in Document Analysis

The aim of unsupervised learning, or clustering, is to find some structure in a set of patterns without the interaction with an explicit teacher. In particular, the goal of the clustering is to identify a finite and discrete set of groupings in the patterns. There

is no universally agreed definition of clusters, but in general the similarity between objects in a group is required to be larger than the similarity between objects belonging to different clusters. The various clustering algorithms can be grouped into three main categories (e.g. [4]): the hierarchical, the crisp and the fuzzy clusterings.

When using clustering algorithms two important issues should be addressed: the choice of an appropriate similarity measure (or distance function) and a criteria to select the number of clusters to be found. The selection of various distance measures without changing the clustering algorithm can give rise to different groupings for a given dataset with significant differences in the final results.

Pattern recognition applications employ clustering algorithms in several ways ([5], page 517).

Clustering algorithms are well suited to deal with unlabeled patterns and this is particularly important in applications where the human validation of the pattern membership can be very expensive. In some cases, a useful approach relies on a preliminary identification of clusters on the basis of unlabeled patterns. In the second step the clusters can be labeled by means of a reduced number of patterns belonging to known classes.

A second approach relies on the use of unsupervised learning for the identification of features that can be used for subsequent processing, for instance for feeding a discriminant classifier. Features computed by means of unsupervised clustering can be considered also for retrieval systems. An application of this approach in DIAR is character clustering that is discussed in Section 3.2.

Exploratory data analysis is another application of clustering techniques that is particularly appropriate to discover natural orderings of the patterns that can be suitably used to design complex pattern recognition systems. Several techniques described in this chapter can be used in this context.

In the rest of this section we analyze some recent applications of clustering methods, with a special emphasis on SOM-based approaches, in the field of Document Image Analysis.

3.1 Symbol thinning

Thinning algorithms are used, in pre-processing, to extract features based on the symbol skeleton. These features can be used in handwritten character recognition systems to allow a recognition independent from the stroke thickness. Ahmed proposed in [6] a clustering-based skeletonization algorithm (CBSA) implemented by using SOM. The CBSA is composed by two main steps: in the first step, some clusters corresponding to adjacent pixels are located from the input image; in the second step the skeleton is built connecting together the closest cluster centers. The clustering step is implemented in [6] by means of a particular SOM (the self-organizing graph) where the adjacency of neurons can change during learning. More recently, a topology-adaptive self-organizing neural network has been proposed for skeletonization [7]. The model grows in size over time to improve the adaptation capabilities of the cluster centers with respect to a SOM with a fixed dimension. The system can work on binary patterns, dot patterns and on gray-level patterns; moreover, it can handle rotated patterns. A similar approach is

described also in [8], whereas a multi-scale skeletonization method based on SOM is described in [9].

3.2 Character clustering

Character clustering is used in character-like coding where some objects (that could correspond to characters) are clustered considering the shape similarity. Each word is then represented by concatenating the codes assigned to the individual objects. In character-like coding, in contrast with OCR, no alphabetical class is assigned to symbols. To retrieve the words the query is encoded with the same algorithm used during the indexing and is compared with the indexed words. This matching can be achieved in several ways. For instance in [10] the words are represented with strings and compared by means of an inexact string matching technique. By adopting this symbolic representation indexed words can be sorted allowing users to retrieve words printed with different fonts as well as to satisfy partial-match queries.

Character clustering is used also in some document image compression algorithms. These methods first group similar symbols, that can potentially correspond to characters. In the second step the characters and the background image are compressed with specific algorithms [11, 12]. For instance, in [12] text images are compressed by extracting the *marks* (connected components) in each page, and building a library of marks. In the next step each mark is replaced with a pointer to the closest item in the library. The mark clustering is performed by means of a simple template matching algorithm that can provide good results when processing documents with a small variability of fonts and a low level of noise.

A related method is addressed in [13], where a hierarchical clustering algorithm is used to enhance degraded document images. The five feature vectors are based on: *histograms* of the projection profiles; distances from the bounding box to the characters outer contour; *pixel correlation*; *subsampling* of the normalized image; *stroke* direction distribution. Bitmaps of the symbols belonging to each cluster are identified, and an average symbol for each class is computed. By replacing the original symbols with the average bitmap it is possible to render the document at arbitrary resolutions and enhance degraded document images.

3.3 Handwriting

In the 1990's several methods have been proposed for the recognition of isolated handwritten digits by means of Self Organizing Maps.

A three-stage recognition module for handwritten numerals is described in [14]. The first stage is based on a SOM whose aim is to create prototypes representing allo-graphs. The trained SOM captures the similarities between input digits, and the gradual variations in shape from class to class is reflected in the feature map. Prototypes that are close in the array generally represent similar patterns. Therefore, neighborhood information can be integrated to estimate class confidence values. The second stage maps distance values into membership values. To obtain a fuzzy membership for each neuron a set of "sigmoid functions" is added in order to convert the distances between the input (unknown) pattern and the prototypes into membership values.

The third stage performs the final classification by means of a fully connected MLP that uses the topologically ordered array of allograph membership values as input features.

A three-dimensional SOM for unconstrained handwritten numeral recognition is described in [15]. The third dimension is basically defined by taking into account 11 layers of 9x9 SOMs. The neighborhood in this case consists of the units that are within a cube that is centered on the winning node. The initial neighborhood size is 6. Another important aspect of the system is the combination of unsupervised and supervised learning principles by using a LVQ training algorithm together with the standard SOM training.

A hybrid handwritten word recognition using self-organizing feature map, discrete HMM, and evolutionary programming has been proposed in [16]. The purpose of the SOM clustering is to partition the feature space into a set of codeword vectors to limit the number of observation symbols in discrete HMM training. The feature vectors computed from more than 400,000 word frames are training data for a map having a 7x7 hexagonal topology. After convergence, the weight vectors of the map are used as codewords to describe the frames of a word, that is represented by a sequence of 2-dimensional codeword positions in the map. The neighborhood information preserved by the SOM is used for smoothing the trained HMM parameters.

In [17] handprinted character recognition is addressed. The SOM is used in conjunction with a robust feature representation for the character that captures the local structure of the strokes. The character is represented with N ellipses, described by 4-D feature vectors containing the center, the length of mayor axis and its orientation, for each ellipse. A modified SOM is used to accomplish the elastic matching and find the correspondence between the feature points. When the network converges, a mapping between the input feature patterns and the neuron support is obtained. The geometry of the neuron support is not a fixed square, but is roughly similar to the skeleton of the standard character.

3.4 Text

The WEBSOM [18] is a SOM-based system that is able to organize large document collections according to textual similarities. The feature vectors describing documents are statistical representations of their vocabularies. The main goal of the work described in [18] is the scaling up of the SOM algorithm in order to process large collections of high-dimensional data. In the experiments 6,840,568 patent abstracts have been mapped onto a 1,002,240-node SOM. To reduce the feature vector size, some random projections of weighted word histograms are computed, thus obtaining 500-dimensional vectors. A similar application has been described also in [19].

In [20] hierarchical feature maps are built considering feature vectors containing the occurrences of 489 terms in each document. In so doing it is possible to reveal the similarity of documents contained in a text archive. The hierarchical representation achieved by the proposed method is claimed to be well suited for text archive organization.

The text clustering features of a SOM are considered in [21] to build a lexical analyzer designed to focus on a very limited sub-set of the whole dictionary. Each string S



Fig. 1. Example of SOM built on a French dataset.

is represented by a vector $X = [X_0, X_1, \dots, X_{25}]$ where X_i corresponds to the number of characters C_i ($C_0 = 'A'$, $C_{25} = 'Z'$) in the string S . The anagrams of S share also the same representation. The map is a two dimensional array, organized as a torus to avoid singularity effects on the sides. The neighborhood relations in the projected space of the map are used to define a short list of hypothesis considered for spell checking.

3.5 Discussion

In this section we analyzed some applications of SOM in the domain of DIAR. The main advantage of SOM clustering with respect to other clustering algorithms, like K-means, is the spatial organization of the neurons that reflects cluster similarity into prototype proximity in the 2D space. The distance among prototypes in the SOM map can therefore be considered as an estimate of the similarity between objects belonging to clusters. It is important to remark that in general the use of SOM multivariate data projection on large data sets is not advisable due to the high computational cost [22]. However, usually a reduced number of objects (obtained from a few random pages) are used to compute the 2D mapping. The mapping obtained with this smaller number of objects is subsequently used to label all the patterns to be processed.

4 Word Indexing

Word indexing can either process the output of Optical Character Recognition (OCR) engines or directly work on the document image. When the use of OCR is not advisable, either due to the low quality of images or to the presence of non-standard fonts, then image-based word retrieval is a viable alternative [23].



Fig. 2. Example of SOM built on the Gothic dataset.

Two main strategies have been considered in the literature: holistic word representation and character-like coding. In holistic word representation each word image is encoded by means of some of its most salient features (e.g. the number of characters or the number of ascenders/descenders) [24]. A particular case of holistic word representation is zoning (e.g. [25]) that consists of overlapping the word image with a fixed-size grid and computing some values (e.g. the density of black pixels) in each grid region. In methods based on character-like coding some objects (that potentially correspond to characters) are extracted from each word. The word is then represented by concatenating the codes assigned to the objects on the basis of shape similarity. In the research described in this paper we deal with modern printed documents that frequently contain text printed with legacy fonts. Moreover, due to changes in the language, the contemporary dictionaries can provide a little help for the recognition.

In this section we describe the use of SOM clustering for performing the word indexing with a character-based approach (see also [26]). During the indexing, each document image is first processed with a layout analysis tool that identifies the text regions and extracts the words. The words are then split into six disjoint index partitions on the basis of the image aspect-ratio. The organization of words into index partitions allows us to reduce the number of words to be compared against a given query and to build uniform representations for the words having approximately the same number of characters.

In the retrieval step a query word typed by the user is processed with the \LaTeX software and one word image is generated. This query image is then associated with some partitions, and a suitable word representation based on character clustering is computed. The indexed words are then sorted on the basis of their similarity to the query by appropriate steps that are described in detail in [26].

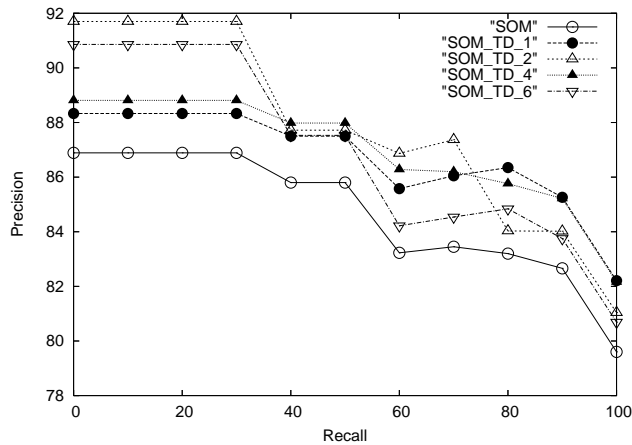


Fig. 3. Comparison of precision-recall plots with and without the use of tangent distance. The transformations 1,2, 4, and 6 correspond to vertical shift, hyperbolik, scale, and line thickness, respectively.

4.1 Character clustering

One important component of word indexing is the Character Object (*CO*) clustering. One *CO* is a part of the word image that usually corresponds to a character. Character Objects are identified by merging overlapping connected components [27]. The *CO*s extracted from a few random pages are used to compute appropriate collection-specific character prototypes that are needed to represent the indexed words. The character prototypes are identified by clustering the *CO*s. Each *CO* image is scaled to fit an 8 by 10 grid, resulting in an 80-dimensional feature vector that is used to train a SOM.

4.2 Experimental Results

To evaluate the effectiveness of the proposed SOM.TD model we compared the precision-recall plots that are obtained by using the SOM and the SOM.TD for character clustering. In the experiments we used two datasets described in [26]: the Gothic and the French one. The two datasets have complementary features. The first collection is composed by few pages printed with a font that is not recognized by current off-the-shelf OCR packages. On the opposite, the second collection contains more than 600 pages printed with a standard font.

Since the tangent distance is a technique particularly suited to deal with data-sets where the number of training samples is low, we expect to have better results with the Gothic collection. This idea is confirmed by the experimental results that do not show significant differences between the SOM and the SOM.TD when dealing with the French data-set. As a reference, we show in Figure 1 one map that has been computed for this collection. We will not report results for this collection.

On the other hand, when dealing with the Gothic data-set we have some interesting results that it is worth to describe. As a reference, we show in Figure 2 one map that has



Fig. 4. Graphical representation of a word SOM (French language).

been computed for this collection. One quantitative comparison of the two approaches is achieved by testing the retrieval system described in [26] without the use of the word alignment method. The precision-recall plots shown in Figure 3 are obtained by running the word retrieval system with 26 query word representative of both frequent and rare words and averaging the single plots. The five plots correspond to the standard map (“SOM”) and to the results that can be achieved by using some relevant transformations. From this figure we can see that the SOM_TDs computed with these transformations improve the precision-recall plots with respect to the standard SOM.

5 Word clustering

In holistic word indexing the SOM is used to cluster together most similar words (from a graphical point of view). The word images extracted with the RLSA algorithm are linearly scaled to the normalized dimensions, computed for each partition, obtaining a vectorial representation where each vector item contains the average gray level of the pixels belonging to the corresponding grid cell. The main problem of this approach is the high vector size (hundreds of items) that is reflected into a long training time. However, it should be remarked that the training is performed during the indexing, that can be considered an off-line step.

In Figure 4 we show a SOM computed by processing the pages in a French book. A deeper analysis of the contents of a few neurons is reported in Figure 5 where the words are ordered on the basis of the distance from the centroid of the cluster. In general the farthest words are loosely related with those closer to the centroid. The large vector size affects also the retrieval performance for problems related to the curse of dimensionality. To speed-up the search in high dimensional spaces we proposed in [28] a method based on the combination of SOM clustering with the search in a low dimensional space obtained by PCA projection.

pendant		une	
pendant		une	
pendant		une	
pendant		une	
pendant		une	
pendant		une	
pendant	nom	une	
pendant	nom	une	
pendant	nom	une	
pourrait	nom	une	
peuvent	nom	une	
paraffine	nom	une	
plusieurs	nom	non	une
préparé	nom	non	une
portions	man	aux	une
potasses	rien	non	une
passion	rien	aux	une
changer	CRIS	aux	une
agissent	ama	aux	une
prétend	BOIS	non	une
appareil	noir	aux	une
emploi	ama	aux	une
général	zinc	aux	une
vestiges	voir	aux	une
general	rine	eux	une
passion	verl	aux	une
troque	zinc	aux	une

Fig. 5. Contents of four neurons of the word SOM shown in Figure 4.

The main steps performed in the word retrieval are summarized as follows. We first identify the three clusters closer to the query. In the second step we search the most similar words sorting the PCA-projected vectors. Lastly, to merge the three lists and refine the final ranking, we compute the distance between the query word and each word in the three lists in the original space. Some detailed experiments on the use of SOM for holistic word indexing are described in [28].

In this paper we show some pictures summarizing the results achievable. Figure 4 contains a graphical representation for a word SOM, whereas the contents of few neurons are shown in Figure 5.

6 Non-latin scripts

One of the most important features of the clustering approaches described in this paper is their ability to adapt to various fonts and languages without a large amount of prior



Fig. 7. Graphical representation of a word SOM for partition 1 (Devnagari script).

References

1. Kohonen, T.: Self-organizing maps. Springer Series in Information Sciences (2001)
2. Bernard, E., Casasent, D.: Invariance and neural nets. IEEE Transactions on Neural Networks 2(5) (1991) 498–508
3. Simard, P.Y., LeCun, Y., Denker, J.S.: Memory-based character recognition using a transformation invariant metric. In: Int'l Conference on Pattern Recognition. (1994) 262–267
4. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3) (2005) 645–678
5. Duda, R.O., Hart, P., Stork, D.G.: Pattern Classification. John Wiley & sons (2001)
6. Ahmed, P.: A neural network based dedicated thinning method. PRL 16(6) (1995) 585 – 590
7. Datta, A., Parui, S.K., Chaudhuri, B.B.: Skeletonization by a topology-adaptive self-organizing neural network. Pattern Recognition 34 (2001) 617–629
8. Sasamura, H., Saito, T.: A simple learning algorithm for growing self-organizing maps and its application to skeletonization. In: Int'l Joint Conference on Neural Networks. Volume 1. (2003) 787–790
9. Palenichka, R.M., Zaremba, M.B.: Multi-scale model-based skeletonization of object shapes using self-organizing maps. In: Int'l Conference on Pattern Recognition. (2002) 143–146
10. Lu, Y., Tan, C.: Information retrieval in document image databases. IEEE Transactions on Knowledge and Data Discovery 16(11) (2004) 1398–1410
11. Haffner, P., Bottou, L., Howard, P.G., LeCun, Y.: DjVu: analyzing and compressing scanned documents for Internet distribution. In: Int'l Conference on Document Analysis and Recognition. (1999) 625–628
12. Witten, I.H., Moffat, A., Bell, T.C.: Managing gigabytes: compressing and indexing documents and images. Academic Press (1999)
13. Hobby, J.D., Ho, T.K.: Enhancing degraded document images via bitmap clustering and averaging. In: Int'l Conference on Document Analysis and Recognition. (1997) 394–400

र	क	म	य	त्र	आ	य	दा	दा	त	च	ल	उ	न	की	वा	णी	जि	त	नी	आ	ली						
वा	च	पा	च	शा	प	ग्र	ह	ण	ह	ल	न	हीं	है	उ	न्	हो	ने	आ	दे	श	प्रा	र्था	ना	अ	र्च	न	
य	त्न	म	ध	य	ज	म	जा	य	कर	त	द	प	त्ति	स	मी	प	यो	ज	ना	पा	च	वों	का	रणों			
त	र	ह	स	त्य	मा	य	ज	म	व	श्य	प	डि	त	प	ति	दे	व	नि	ल	य	नि	क्षे	प	स	ब	धी	
व	क्त	न	म	बा	र	त	त्त्व	ज	ब		वि	द्वान	नि	श्च	य	शि	ष्यो	कि	सी	मि	ल	ने					
बा	द	दा	ता	च	क्र	म	र	ण	म	ल	म	स्थि	त	ले	कि	न	लि	ख	ते	सै	नि	क	यो	ज	न		
स	प	द	म	नु	ष्य	सू	च	ना	अ	द्भु	त	इं	डि	या	प्र	द	श	क	ले	कर	खो	ल	ने	मा	न	ते	
शु	ल्क	वा	क्	व	सू	ल	१	७	ति	रु	चा	नू	सि	चा	ई	नि	रु	त्तर	ग	व	र्न	र	मी	जा	कर	ने	
त	र	ह	जु	म्ल	अ	ब्दु	ल	सु	ल्तानों	अ	न्न	पू	र्णा	के	लि	ए	भ	क्ति	त	त्त्वो	उ	स	से	व्य	क्ति		
न	ग	र	हा	थ	अ	ग	र	गु	णि	तो	अ	र्था	त्	ऐ	श्व	र्य	आ	दे	श	सा	ला	स	ब	धी	उ	स	के

Fig. 8. Graphical representation of a word SOM for partition 2 (Devnagari script).

ति	रु	प	ति	ति	रु	म	ल	वि	श्वा	स	ति	रु	प	ति	मि	ल	क	प्र	ब	ंध	क	को	ल	दा	र	दा	ता	ओं	रा	जा	ओं	स	द	स्यो
वे	द	व्या	स	अ	न्त	र्ग	त	शि	व	जी	जि	स	क	म	हेश्वरः	म	ह	रा	ठो	जो	त	दा	र	ब	ना	यी	न	वा	बो	मा	म	लो		
वे	द	व्या	स	रा	ज	की	य	मि	ल	ती	पा	र्व	ती	न	हीं	कर	ते	आ	चा	र्य	पर	मेश्वर	आ	द	शो	या	त्रि	यो	अ	ग्ने	जो			
अ	धी	न	दे	व	स्थान	चे	ता	व	नी	व्या	स	जी	अ	ति	थि	अ	धि	क	पी	ढियो	थो	डी	दे	र	जा	ये	गी	ध	न	रा	शि			
मि	ल	ती	है	ध्या	न	दि	य	स	स्था	ए	वै	क	टा	दि	धो	खा	दे	प्र	ति	दि	न	ली	जि	ये	अ	न्न	पू	र्णा	दि	डि	ग	जा	य	ग
म	हा	वि	ष्णु	की	जि	ए	मु	ख्यो	दि	श्य	इ	स	लि	ए	इ	स	क	ये	ढ	क	स	क	त	कर	त	म	न	स	का	म				
ल	गु	जा	री	स	हू	लि	तो	दा	तृ	व	सु	ल्तान	स	ला	ह	प्र	ब	ल	प्र	ब	ध	प्र	ब	ध	प्र	थ	म	व्या	स					
अ	च्यु	त	अ	ज्ञान	त	त्क्षे	ण	उ	न	क	क्र	म	श	दा	ता	उ	अ	न्य	वा	प	स	स्व	त्वर	क	म									
प	श्चा	त्	ए	क	द	म	ह	जा	र	आ	प	ब	द	ल	म	त	व्य	प्र	भा	व	र	ख	न	का	रण	ज	न							
अ	नु	ग्र	ह	र	ह	कर	जा	कर	अ	ग	र	अ	ग	र	अ	न	त	का	रण	न	वा	ब	कर	ता	स	म								

Fig. 9. Graphical representation of a word SOM for partition 3 (Devnagari script).

14. Chiang, J.H., Gader, P.: Recognition of handprinted numerals in VISA card application form. *MVA* **10**(3) (1997) 144–149
15. Reddy, N.S., Nagabhushan, P.: A three-dimensional neural network model for unconstrained handwritten numeral recognition: a new approach. *Pattern Recognition* **31**(5) (1998) 511–516
16. Dehghan, M., Faez, K., Ahmadi, M.: A hybrid handwritten word recognition using self-organizing feature map, discrete HMM, and evolutionary programming. In: *Int'l Joint Conference on Neural Networks*. (2000) 515–520
17. Liou, C.Y., Yang, H.C.: Handprinted character recognition based on spatial topology distance measurement. *IEEE Transaction on PAMI* **18**(9) (1996) 941–945
18. Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Honkela, J., Paatero, V., Saarela, A.: Self organization of a massive document collection. *IEEE Transactions on Neural Networks* **11**(3) (2000) 574–585
19. O'Neil, P.: An incremental approach to text representation, categorization, and retrieval. In: *Int'l Conference on Document Analysis and Recognition*. (1997) 714–717
20. Merkl, D.: Text classification with self-organizing maps: some lessons learned. *Neurocomputing* **21**(1–3) (1998) 61–78
21. Ménier, G., Lorette, G.: Lexical analyzer based on a self-organizing feature map. In: *Int'l Conference on Document Analysis and Recognition*. (1997) 1067–1071

