

# Transformation invariant SOM clustering in Document Image Analysis

Simone Marinai, Emanuele Marino, Giovanni Soda  
Dipartimento di Sistemi e Informatica  
Università di Firenze  
Via S.Marta, 3 - 50139 Firenze - Italy  
marinai@dsi.unifi.it

## Abstract

*In this paper, we propose the combination of the Self Organizing Map (SOM) and of the tangent distance for effective clustering in Document Image Analysis. The proposed model (SOM-TD) is used for character and layout clustering, with applications to word retrieval and to page classification. By using the tangent distance it is possible to improve the SOM clustering so as to be more tolerant with respect to small local transformations of the input patterns.*

## 1. Introduction

Most systems in Pattern Recognition are based on supervised learning algorithms, however unsupervised learning techniques gained interest in recent years. In unsupervised learning, or clustering, the training algorithm takes into account unlabeled samples and there is no explicit teacher. In clustering the system usually forms clusters of the input patterns. Specific information about the input data can be used to set the initial value for the number of clusters to be found. An appropriate choice of this parameter is crucial for improving the performance of the clustering algorithms. Another important feature is the type of distance function embedded in the algorithm that strongly influences the final clustering achieved.

In this paper, we focus our attention on a particular clustering algorithm, the Self Organizing Map (SOM) [4], 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 (DIA) sub-tasks and the use of the tangent distance to improve the SOM training.

The paper is organized as follows. In Section 2 we summarize the main features of the SOM and those of the tangent distance and we describe the proposed SOM training based on the tangent distance. In Section 3 we highlight some features of the applications of clustering in Document

Image Analysis. In Section 4 and in Section 5 we discuss the application of the SOM-TD to word retrieval based on character clustering, and to page classification based on layout clustering, respectively. Our concluding remarks are drawn in Section 6.

## 2. Self Organizing Maps

The Self Organizing Map is an artificial neural network that performs clustering by means of unsupervised competitive learning [4]. In the SOM the neurons are usually arranged in a two dimensional 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 influ-

enced 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 [1]: 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 [9] 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  $\gamma$ . One example of transformation  $t$  is the rotation of the pattern by the angle  $\gamma$ . In the pattern space the set of all the transformed patterns  $S_P = \{x \mid \forall \gamma x = t(P, \gamma)\}$  can be considered as a one-dimensional curve that is parametrized by  $\gamma$ . In the general case we must consider several transformations together and therefore we should consider a vector of  $p$  parameters ( $\vec{\gamma}$ ) 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:

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

the equations of the tangent planes are given by:

$$\begin{aligned} T_P(\vec{\gamma}_P) &= P + L_P \vec{\gamma}_P \\ T_Q(\vec{\gamma}_Q) &= Q + L_Q \vec{\gamma}_Q \end{aligned} \quad (5)$$

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 [9]. 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:

$$D1(P, Q) = \min_{x \in T_P} \|x - Q\|^2. \quad (6)$$

## 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:

$$\|x(t) - m_{b(x)}\| = \min_i \{D1(x(t), m_i(t))\}. \quad (7)$$

In section 4 and 5 we will describe two applications of the SOM\_TD training algorithm comparing its performance with respect to the standard SOM model.

## 3. Unsupervised learning in Document Image Analysis

The aim of unsupervised learning, or clustering, is to find some structure in a set of patterns without an explicit interaction with a 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 (e.g. [10]).

When using clustering algorithms two important issues should be addressed: the choice of an appropriate similarity measure (or distance function) and the identification of an initial number of clusters to be found. The selection of various distance measures without changing the clustering



Figure 1. Example of SOM built on the Gothic dataset

algorithm can give rise to different groupings for a given data-set with significant differences in the final results.

Pattern recognition applications employ clustering algorithms in several ways ([3], 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 a subsequent step the clusters can be labeled by means of a reduced number of patterns belonging to known classes. We will analyze a method based on this idea in Section 5 for the classification of pages considering the layout similarity.

A second approach relies on the use of unsupervised learning for the extraction of features that can be used for subsequent processing, for instance feeding a discriminant classifier. Features computed by means of unsupervised clustering can be considered also for retrieval systems. An application of this approach in DIA is character clustering that is applied in Section 4 in the context of word indexing.

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.

## 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 [8].

Two main strategies have been considered in the liter-

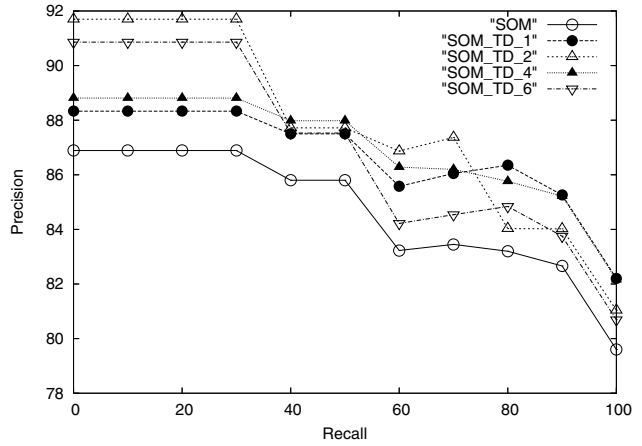
ature: holistic word representation and character-like coding. In holistic word representation each word image is encoded by means of some of its most salient features (e.g. the number of characters or the number of ascenders/descenders) [5]. A particular case of holistic word representation is zoning (e.g. [2]) 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 [7]). 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  $\text{\LaTeX}$  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 [7].

#### 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 [6]. The *COs* 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 *COs*. 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.



**Figure 2. 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.**

#### 4.2. 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 [7]: 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. Therefore, 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 1 the two maps that are computed for this collection starting from the same random map in the initialization step. One quantitative comparison of the two approaches is achieved by testing the retrieval system described in [7] without the use of the word alignment method. The precision-recall plots shown in Figure 2 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

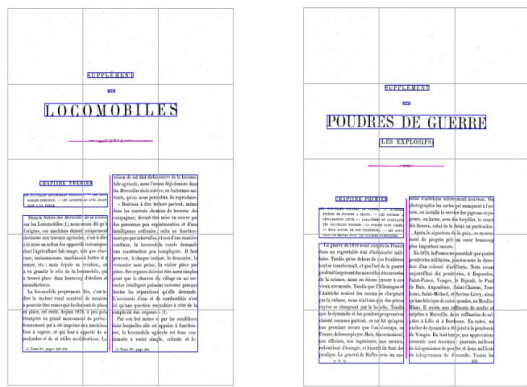


Figure 3. Two actual pages of the issue2 class with the sampling grids.

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. Page Classification

In this section, we discuss the SOM clustering of the page layout for the page classification. We analyze also the use of the SOM\_TD in order to improve the recognition rate for data-sets containing few labeled pages. To compute the SOM map we represent each page with a fixed-size vector. A layout analysis tool is used to extract the homogeneous regions in each page. To adopt the paradigm of tangent vectors in the framework of layout similarity we need to compute a feature vector which smoothly depends on small transformations in the image space.

Basically, we superimpose a uniformly spaced grid over the page (Figure 3) and we compute, for each cell, the percentage of its area that is covered by text, image and line regions. From Figure 3 we can see that the most important transformations for the layout clustering are horizontal and vertical displacements. To simulate these transformations we compute, for each training page, the new representations that are obtained by shifting the grid in the horizontal and vertical directions. By using these transformed page representations we can compute a sort of tangent vectors that can be used to train the SOM\_TD.

### 5.1. Experimental results

We describe the results achieved when using the SOM\_TD model for document layout clustering. Two ex-



Figure 4. Graphical representation of a SOM computed for layout clustering.

periments have been made to analyze the approach with different conditions. In the first experiment we considered a data-set that is typical of Digital Library applications consisting of a collection of pages belonging to a digitized Encyclopedia. The second experiment deals with the classification of invoices on the basis of their layout.

The Encyclopedia data-set is composed by 6 books with 4035 pages that have been grouped by users into more than ten classes. Among these classes we took into account nine of the most important ones. For the training we used the first book that contains 617 pages. The test is made on the remaining five books. After training the SOM either with the standard or with the SOM\_TD algorithm we attach a class label to each neuron on the basis of the pages in the training set clustered in the neuron.

During the classification we take into account all the pages in the test set belonging to the nine classes of interest and we classify them according to the label attached to the SOM neuron that is closest to each test page. The comparison of the two SOMs is made computing the recognition rate for each class when using a standard SOM and a SOM\_TD. To reduce the variations in the training for each trial we used the same initial map for both learning methods.

To summarize the results we compare in Table 1 the average recognition rate for both maps when changing the map

	12x10	15x10
SOM	59.91	70.74
SOM_TD	60.97	71.41

**Table 1. Encyclopedia Data Set. Average recognition rate for various sizes of the SOM.**

	SOM	SOM_TD
<b>ImageText2</b>	81.33	82.14
<b>Text2Image</b>	69.08	67.05
<b>SecM2</b>	36.30	40.84
<b>Image</b>	96.72	96.72
<b>Issue2</b>	84.21	87.50
<b>ITLPa</b>	28.00	32.00
<b>Text2</b>	96.03	93.65

**Table 2. Encyclopedia Data Set. Error rates for each class with the 15x10 maps.**

size. We can notice that in all the cases the tangent map provide better results and also that larger maps give rise to better results. To further investigate the results achieved we compare, in Table 2 the results obtained for each class with the 15x10 map. We can observe that in general the SOM\_TD provides better results for all the classes with the exception of classes **Text2Image** and **Text2**. As a matter of fact some pages of one of the two classes are confused with the others and vice-versa. The reason for this behaviour is, in our opinion, due to the fact that the **Text2** class is the most populated one and has a very simple layout (text on two columns). In this case the use of tangent vectors is not useful since there are already several training patterns of the class, and the addition of local distortions can be confusing.

The experiments performed on the invoice dataset have been made with the aim of evaluating the effectiveness of the proposed SOM\_TD model when dealing with a large collection of patterns that are labeled only in a small percentage.

The SOM training and labeling has been made in two steps: first a map has been trained with the SOM\_TD training algorithm considering all the patterns belonging to the training set. In the second step we labeled some neurons in the map on the basis of a majority voting generated by 64 pages belonging to 15 classes. The classification is made as before, assigning the unknown page to the class of the closest neuron. Since we have a reduced number of labeled samples we made the experiments with a leave-one-out approach. From Table 3 containing the average recognition rate for the invoice data-set we can notice that with the

	12x10	15x10
SOM	59	52
SOM_TD	58	66

**Table 3. Invoice data-set. Average recognition rate for various sizes of the map.**

15x10 map we have better results with respect to the standard SOM.

## 6. Conclusions

In this paper we analyzed the use of the tangent distance for SOM clustering. The proposed approach is evaluated on two applications in Document Image Analysis: the word retrieval based on character clustering and the layout classification based on page clustering. In both cases the experimental results confirm the hint that the tangent distance is particularly suited when dealing with data sets having a small number of labeled training patterns.

## References

- [1] E. Bernard and D. Casasent. Invariance and neural nets. *IEEE Transactions on Neural Networks*, 2(5):498–508, 1991.
- [2] F. Cesarini, M. Gori, S. Marinai, and G. Soda. INFORMys: A flexible invoice-like form reader system. *IEEE Transactions on PAMI*, 20(7):730–745, July 1998.
- [3] R. O. Duda, P. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & sons, 2001.
- [4] T. Kohonen. *Self-organizing maps*. Springer Series in Information Sciences, 2001.
- [5] S. Madhvanath and V. Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transactions on PAMI*, 23(2):149–164, 2001.
- [6] S. Marinai, E. Marino, and G. Soda. Indexing and retrieval of words in old documents. In *Int'l Conference on Document Analysis and Recognition*, pages 223–227, 2003.
- [7] S. Marinai, E. Marino, and G. Soda. Font adaptive word indexing of modern printed documents. *IEEE Transactions on PAMI*, 28(8):1187–1199, 2006.
- [8] M. Mitra and B. Chaudhuri. Information retrieval from documents: A survey. *Information Retrieval*, 2(2/3):141–163, 2000.
- [9] P. Y. Simard, Y. LeCun, and J. S. Denker. Memory-based character recognition using a transformation invariant metric. In *Int'l Conference on Pattern Recognition*, pages 262–267, 1994.
- [10] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.