

## MM-WEBSOM: A VARIANT OF WEBSOM BASED ON ORDER STATISTICS

*A. Georgakis, C. Kotropoulos, A. Xafopoulos and I. Pitas*

Department of Informatics  
 Aristotle University of Thessaloniki  
 Thessaloniki 54006, Greece  
 e-mail: {costas, pitas}@zeus.csd.auth.gr

### ABSTRACT

A variant of the WEBSOM architecture for information retrieval is proposed in this paper. WEBSOM is based on the self-organizing map that employs a linear LMS adaptation rule for updating the weight vector of each neuron. Accordingly, the weight vector converges asymptotically to the conditional cluster mean of the feature vectors assigned to the class represented by the weight vector of the neuron. We propose to replace the updating rule by employing the marginal median. The objective is to overcome the drawbacks of the standard technique in the presence of outliers in the training set and to use robust estimators of the reference vectors for each class. Experimental results demonstrate a superior performance of the proposed variant against the standard algorithm, in terms of the number of training iterations needed so that the mean square error (i.e., the average distortion) drops to the  $\frac{1}{e}$  of its initial value. We provide precision-recall curves as a measure of the quality of the clustering procedure as well. Both techniques are tested using a corpus that comprises web pages selected over the Internet.

### 1. INTRODUCTION

Artificial neural networks (NN) has been an active research area for the past three decades [1, 2]. A large variety of learning algorithms (error-correction, memory-based, Hebbian, competitive learning, Boltzmann machines, supervised or unsupervised, etc) are employed during the training phase of the NN. Self-organizing maps (SOMs) are feedforward artificial neural networks with one layer of input nodes and a single computational layer of neurons arranged on a two or three-dimensional lattice [3, 4, 5, 6]. Every neuron in the computational layer is fully connected with the input layer, while the topology on the computational layer can be either hexagonal or orthogonal.

SOMs are capable of forming a non-linear transformation or mapping from an arbitrary dimensional data manifold onto the low-dimensional discrete map. The algorithm takes into consideration the relations of the input feature vectors and computes an optimal representation that approximates these features in the sense of some error criterion, usually the mean square error (MSE).

WEBSOM is a two-layer SOM. It is a method for organizing document collections onto map displays in order to enhance document organization and subsequently, to improve information

---

This work was supported by the European Union IST Project "HYPERGEO: Easy and friendly access to geographic information for mobile users" (IST-1999-11641).

retrieval. The map is organized according to the contextual similarities of the full-text documents.

Let  $\mathcal{X}$  denote the set of vector-valued observations  $\{\mathbf{x}_j(t) = (x_{1j}(t), x_{2j}(t), \dots, x_{N_w j}(t))^T \in \mathbb{R}^{N_w}\}$  and  $\mathcal{W}$  the set of reference vectors  $\{\mathbf{w}_m(t) \in \mathbb{R}^{N_w}, m = 1, \dots, K\}$ . The parameter  $t$  denotes discrete time and  $\mathbf{w}_m(0)$  is randomly initialized. Competitive learning finds the best-matching (winning) reference vector  $\mathbf{w}_s(t)$  to a specific feature vector  $\mathbf{x}_j(t)$  with respect to a certain metric. The metric usually employed is the Euclidean distance. The index  $s$  of the winning reference vector is given by:

$$s = \arg \min_k \|\mathbf{x}_j(t) - \mathbf{w}_k(t)\|. \quad (1)$$

In the sequel, reference vectors are the weight vectors of the neurons.

In the standard SOM the weight vector of the winner as well as the weight vectors of the neurons in its neighborhood are modified toward  $\mathbf{x}_j(t)$  as follows:

$$\mathbf{w}_i(t+1) = \begin{cases} \mathbf{w}_i(t) + \eta_{s_i}(t) [\mathbf{x}_j(t) - \mathbf{w}_i(t)], & \forall i \in \mathcal{N}_s \\ \mathbf{w}_i(t), & \forall i \notin \mathcal{N}_s \end{cases} \quad (2)$$

where  $\eta_{s_i}(t)$  is the learning rate parameter and  $\mathcal{N}_s$  denotes the neighborhood of the winner. Eq. (2) can be rewritten in the following form:

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + a(t) c_{ij}(t) [\mathbf{x}_j(t) - \mathbf{w}_i(t)] \quad (3)$$

where  $a(t)$  is the adaptation step and  $c_{ij}(t) = 1$  if the  $j$ th feature vector is assigned to the  $i$ th neuron at the iteration  $t$ , otherwise  $c_{ij}(t) = 0$ . The weight vector of any neuron at the iteration  $(t+1)$  of the training phase is a linear combination of the input vectors assigned to it during the past iterations:

$$\begin{aligned} \mathbf{w}_i(t+1) = & \mathbf{w}_i(0) \prod_{k=0}^t \left( 1 - a(t-k) \sum_j c_{ij}(t-k) \right) + \\ & \sum_{k=1}^t a(t-k) \sum_j c_{ij}(t-k) \mathbf{x}_j(t-k) \cdot \\ & \prod_{m=0}^{k-1} \left( 1 - a(t-m) \sum_j c_{ij}(t-m) \right) + \\ & a(t) \sum_j c_{ij}(t) \mathbf{x}_j(t). \end{aligned} \quad (4)$$

It can be shown that only in the special case of one neuron when  $c_{ij}(t) = 1, \forall j$ , and the adaptation step sequence  $a(t) = 1/(t+1)$  the reference vector is the arithmetic mean of the observations. That is, the maximum likelihood estimator (MLE) of location. In any other case, the reference vectors are not the optimal estimators of the cluster means. A summary of the disadvantages of the standard SOM algorithm used in WEBSOM is as follows:

1. SOM does not use optimal estimators for obtaining the reference vectors  $\mathbf{w}_i(t)$  that match the probability density function (pdf)  $f_i(\mathbf{x})$  of each class,  $i = 1, \dots, K$ .
2. It lacks robustness against erroneous choices for the winner vector because it is well known that linear estimators have poor robustness properties [7, 8, 9].
3. It does not possess robustness against outliers.

In order to overcome these problems and to enhance the performance of the WEBSOM architecture, a variant of the standard architecture is proposed that employs multivariate order statistics [10, 11, 12]. This variant treats efficiently the outliers, because it inherits the robustness properties of the order statistics [9].

The outline of the paper is as follows. Section 2 describes briefly the marginal ordering principle and the proposed variant. Sections 3.1 and 3.2 describe the formation of the corpus and the language modeling employed to construct the feature vectors. The high dimensionality nature of the feature vectors is reduced in Section 3.3. The word/document clustering achieved by the proposed variant is discussed in Section 3.4. Experimental results are presented in Section 4 and conclusions are drawn in Section 5.

## 2. SELF-ORGANIZING MAPS BASED ON MARGINAL ORDERING

The lack of any obvious and unambiguous means of ranking multivariate observations is surpassed through the definition of methods that employ various types of *sub-ordering* principles such as *marginal ordering*, *reduced (aggregate) ordering*, *partial ordering* and *conditional (sequential) ordering*. A discussion on these sub-ordering principles can be found in [13].

The variant employed in this paper relies on the concept of marginal ordering. In marginal ordering, the samples are ordered independently along each of the  $N_w$ -dimensions:

$$x_{j(1)}(t) \leq x_{j(2)}(t) \leq \dots \leq x_{j(N)}(t), \quad j = 1, 2, \dots, N_w. \quad (5)$$

The marginal median  $\mathbf{x}_{med}$  of  $N$  feature vectors is defined by

$$\begin{aligned} \mathbf{x}_{med} &= \text{median} \{ \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \} \\ &= \begin{cases} (x_{1(\nu+1)}, x_{2(\nu+1)}, \dots, x_{N_w(\nu+1)})^T & \text{for } N = 2\nu + 1 \\ \left( \frac{x_{1(\nu)} + x_{1(\nu+1)}}{2}, \dots, \frac{x_{N_w(\nu)} + x_{N_w(\nu+1)}}{2} \right)^T & \text{for } N = 2\nu. \end{cases} \quad (6) \end{aligned}$$

The concept of the marginal median can be used in the following way. Let  $\mathbf{X}_i(t-1)$  denote the set of the feature vectors that have been assigned to each class  $i$ ,  $i = 1, 2, \dots, K$  until iteration  $(t-1)$ . At iteration  $t$  we find the winner vector  $\mathbf{w}_s(t)$  which is then updated by

$$\mathbf{w}_s(t+1) = \text{median} \{ \mathbf{x}_j(t) \cup \mathbf{X}_s(t-1) \}. \quad (7)$$

The neighboring neurons,  $i \in \mathcal{N}_s(t)$ , are updated as follows:

$$\mathbf{w}_i(t+1) = \text{median} \{ a(t)\mathbf{x}_j(t) \cup \mathbf{X}_i(t-1) \}. \quad (8)$$

The proposed variant shall be referred as the MM-WEBSOM.

## 3. APPLICATION TO INFORMATION RETRIEVAL

### 3.1. Corpus description and preprocessing steps

Throughout this process a training corpus, that is a collection of sample texts, is used. It comprises 650 full-text HTML files containing nearly 120,000 words (word tokens), which were manually collected over the Internet. The HTML files are web pages of touristic content and in its current state is biased in the sense that web pages related to Greece, Spain and Germany form the majority [14]. The selected files are annotated by dividing them into 18 categories related to tourism such as accommodation, history, geography, etc., so that ground truth is incorporated.

Before testing the standard WEBSOM technique as well as the proposed variant a series of actions had to be taken in order to encode the textual data into vectors. The first step deals with *HTML* as well as plain *text cleaning*. HTML cleaning refers to the removal of the HTML tags and entities, and the appropriate treatment of some special tags, while plain text cleaning refers to the removal of URLs, email addresses, numbers, punctuation marks and the formation of word tokens. The sole punctuation mark left intact is the full stop, providing a rough sentence delimiter. Collocations, i.e., expressions consisting of two or more words are meaningful only within the limits of a sentence [15]. Text cleaning also includes the removal of some common English words (such as articles, determiners, prepositions, pronouns, conjunctions, complementizers, abbreviations) and some non-English frequent terms in a processing step called *stopping*. All the remaining words were converted to lowercase except for some cases where acronyms were detected. The aforementioned step resulted in a corpus of 70,000 word tokens.

Subsequently, *stemming* was performed. Stemming refers to the elimination of word suffixes so that the resultant vocabulary shrinks, though keeping the informative context of the text. It can be considered as an elementary clustering technique, with the word roots (stems) regarded as the clusters. The underlying assumption for the successful usage of a stemming program, called a stemmer, is that morphological variants of words are semantically related [16, 17]. The application of the commonly used Porter stemmer [18] resulted in a vocabulary size of  $N \simeq 8700$  stem types (distinct occurrences).

### 3.2. Language Modeling

The last step was the computation of the contextual statistics for every word  $i$  in the corpus. For this purpose, the second version of the CMU-Cambridge Statistical Language Modeling Toolkit was used [19]. In a first attempt, the following statistics (i.e., maximum likelihood estimates of conditional probabilities) can be used to encode the  $i$ th word stem in the vocabulary [20]:

$$x_{il} = \frac{n_{il}}{N_i}, \quad l = 1, 2, \dots, N \quad (9)$$

where  $n_{il}$  is the number of times the pair ( $i$ th word stem,  $l$ th word stem) occurred in the corpus,  $N_i$  is the number of times the  $i$ th word stem occurred in the corpus, and  $N$  is the number of word

stems in the vocabulary. By using Eq. (9), the following word vectors,  $\mathbf{x}_i$ , can be computed [21]:

$$\tilde{\mathbf{x}}_i = \frac{1}{N_i} \sum_{l=1}^N n_{il} \mathbf{e}_l \quad (10)$$

or

$$\tilde{\mathbf{x}}_i = \frac{1}{N_i} \begin{bmatrix} \sum_{\substack{l=1 \\ l \neq i}}^N n_{il} \mathbf{e}_l \\ \epsilon \mathbf{e}_i \\ \sum_{\substack{m=1 \\ m \neq i}}^N n_{im} \mathbf{e}_m \end{bmatrix} \quad (11)$$

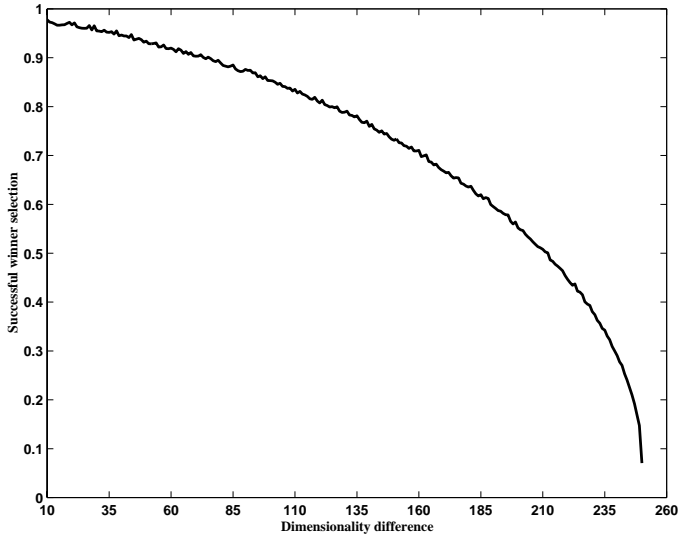
where  $\mathbf{e}_i$ ,  $i = 1, 2, \dots, N$ , denotes the  $(N \times 1)$  unit vector having one in the  $i$ th position and zero elsewhere.

### 3.3. Dimensionality Reduction

If Eq. (11) is used to model each word stem, feature vectors of dimension  $3N \times 1$  results. The problem of the high dimensionality of the feature vectors is tackled by using any dimensionality reduction technique. For example dimensionality reduction to  $N_w$  is achieved by a linear projection

$$\mathbf{x}_i = \Phi \tilde{\mathbf{x}}_i \quad (12)$$

where  $\tilde{\mathbf{x}}_i$  is a  $3N \times 1$  feature vector,  $\mathbf{x}_i$  is the projected vector and  $\Phi$  is an appropriate matrix of dimensions  $N_w \times 3N$ .



**Fig. 1.** Percentage of successfully identified winning neurons in the formation of the word categories map using the standard WEBSOM.

Kaski *et al.* have shown that a suboptimal but faster approach to the previous problem is the use of a random matrix that has the following properties [22]:

- The components in each column are chosen to be independent, identically distributed Gaussian variables with zero mean and unit variance.

- Each column is normalized to unit norm.

In the case under consideration the dimensionality of the projected space was chosen to be equal to 300 ( $N_w = 300$ ).

An additional step toward dimensionality reduction can be achieved using the following technique. The sample variance of the  $m$ th component in the feature vector is computed

$$u_m(t) = \sum_{j=1}^N (x_{mj}(t) - \bar{x}_m(t))^2, \quad m = 1, 2, \dots, N_w \quad (13)$$

where  $\bar{x}_m(t) = \frac{1}{N} \sum_{j=1}^N x_{mj}(t)$  is its sample mean, and  $N$  is the number of feature vectors. The components of the feature vectors  $\mathbf{x}_j(t)$  and the neuron weights are rearranged in descending order with respect to their sample variance  $u_m(t)$ . The Euclidean distance used in Eq. (1) is then decomposed as follows:

$$\|\mathbf{x}_j(t) - \mathbf{w}_k(t)\|^2 = \sum_{n=1}^{d'} (x_{(n)j}(t) - w_{(n)k}(t))^2 + \sum_{n=d'+1}^{N_w} (x_{(n)j}(t) - w_{(n)k}(t))^2 \quad (14)$$

where  $x_{(n)j}(t)$  and  $w_{(n)k}(t)$  denote the  $n$ th component of the ordered feature vector  $\mathbf{x}_j(t)$ , and the weight vector respectively. Moreover,  $d'$  is an arbitrary number such that  $d' < N_w$ . The first sum in Eq. (14) contains those components of the feature vectors with the largest sample variance, whereas the second sum contains the components whose impact on the selection of the winning neuron is more or less the same for each feature vector. By selecting the parameter  $d'$  and omitting the second sum in Eq. (14) an accurate estimation of the winning neuron can be achieved. Figure 1 presents the percentage of the successfully identified winner neurons with respect to the dimensionality difference  $N_w - d'$ . It is seen that the performance of the proposed technique is satisfactory even if the feature vector dimensionality is reduced to one half of its original value.

### 3.4. Clustering

After the completion of the preprocessing phase all feature vectors  $\mathbf{x}_i$  are presented iteratively an adequate number of times to both the standard WEBSOM and the proposed variant that employs the marginal median. Both neural networks perform a clustering of the word feature vectors  $\mathbf{x}_i$  in an effort to build clusters of semantically related words. This process yield the so-called *word categories map* (WCM). The WCM created by the marginal median WEBSOM is depicted in Fig. 2. The grey levels of the map correspond to different word densities in each neuron/cluster. Hexagons with grey levels near 255 (white colour) imply that fewer word stems have been assigned to those neurons/clusters, whereas, grey levels near 0 (black colour) imply larger densities.

Subsequently, for each document in the corpus, a histogram of word categories is computed to derive the so-called *document vector*,  $a_k$ . The standard architecture as well as its variant is used to construct clusters of contextually similar documents. The resulted map is called *document map* (DM). The document map computed by the marginal median WEBSOM is depicted in Fig. 3. It can be seen that the documents assigned to the highlighted neurons are contextually related to Spain.

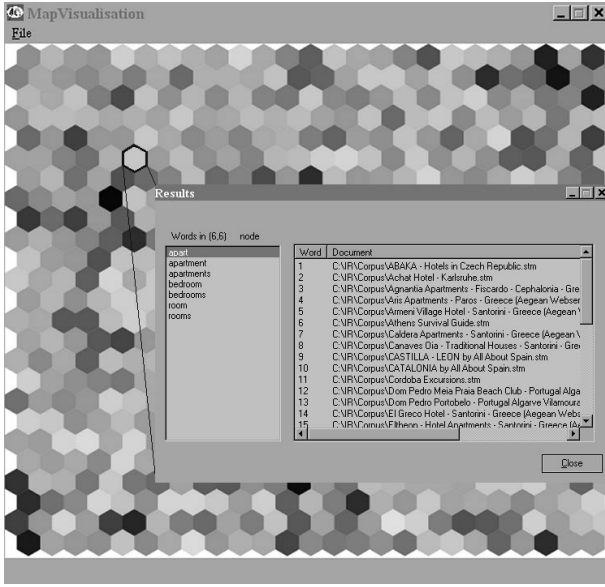


Fig. 2. The words categories map of the marginal median WEBSOM for 637 neurons/clusters.

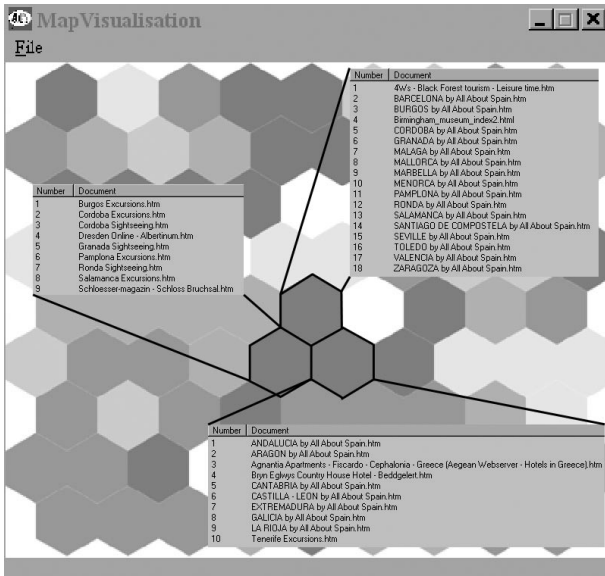


Fig. 3. The document map for 86 neurons and 650 documents.

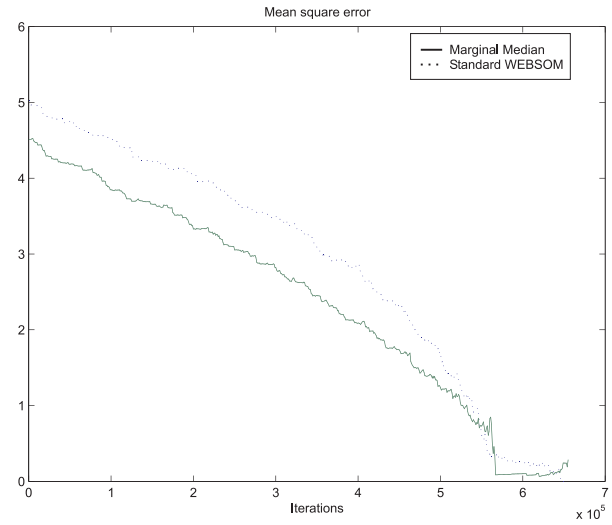


Fig. 4. The MSE curves for the standard WEBSOM and the MM-WEBSOM variant.

#### 4. EXPERIMENTAL RESULTS

The performance of the proposed MM-WEBSOM is measured against the standard WEBSOM using the training mean square error between the neuron weights and the document vectors assigned to each cluster as a figure of merit. Figure 4 depicts the MSE curves for both techniques. It can be seen that even from the beginning, the marginal median variant outperforms the standard technique. This is due to the presence of many outliers in the first iterations of the training procedure.

Furthermore, for the standard WEBSOM the number of training iterations needed so that the MSE drops to the  $\frac{1}{e}$  of the initial value is nearly 11% higher than the marginal median variant.

The quality of the clusters is measured by querying the retrieval system using a sample test document. The system retrieves the training corpus documents that are represented by the best matching neuron of the document map. The training documents retrieved are ranked according to their Euclidean distance from the sample test document. Subsequently they are classified as either being relevant or not to the sample test document according to their annotation. Table 1 is the  $2 \times 2$  contingency table which shows how the training corpus is divided.

For both techniques the *precision-recall* curves are calculated [23]. Precision is defined as the proportion of retrieved documents that are relevant,

$$P = \frac{r}{n_2} \quad (15)$$

and, recall is the proportion of relevant documents that are retrieved,

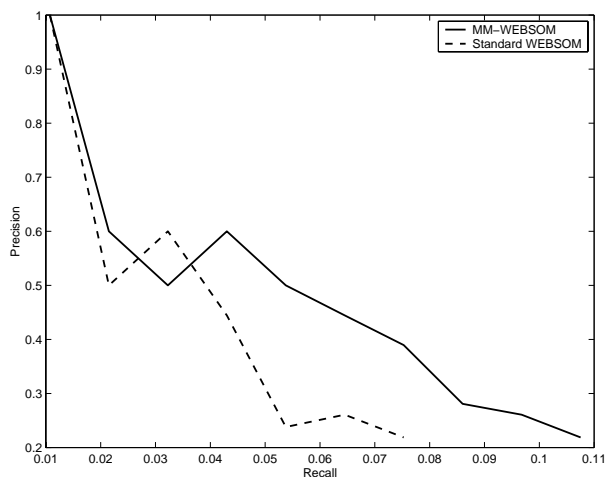
$$R = \frac{r}{n_1} \quad (16)$$

where  $r$  denotes the number of relevant documents which are retrieved,  $n_1$  is the total number of relevant documents in the corpus and  $n_2$  denotes the number of retrieved documents.

Figure 5 depicts the precision-recall curves for both techniques. It is seen that the marginal median variant outperforms the standard WEBSOM by clustering more relevant documents together.

	Retrieved	Not-Retrieved	
Relevant	$r$	$x$	$n_1 = r + x$
Not Relevant	$y$	$z$	
	$n_2 = r + y$		

**Table 1.** Contingency table for evaluating retrieval.



**Fig. 5.** The precision-recall curves for both techniques. The same test document was classified into the ‘history’ category.

## 5. CONCLUSIONS

The inherent drawbacks of SOMs used in the standard WEBSOM algorithm motivated us to develop a variant where multivariate median operators are employed. The first experimental results obtained indicate that the novel marginal median variant outperforms the standard algorithm.

## 6. REFERENCES

- [1] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Upper Saddle River: Prentice-Hall, 1999.
- [2] P. K. Simpson, *Artificial Neural Systems*, Pergamon Press, 1990.
- [3] T. Kohonen, *Self Organizing Maps*, Germany: Springer-Verlag, 1997.
- [4] T. Kohonen, “Self-organization of very large document collections: State of the art,” in *Proc. of ICANN*, 1998, vol. 1, pp. 65–74.
- [5] T. Kohonen, S. Kaski, K. Lagus, J. Salojrvi, J. Honkela, V. Paatero, and A. Saarela, “Self organization of a massive text document collection,” in *Kohonen Maps*, 1999, pp. 171–182, Elsevier.
- [6] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, V. Paatero, and A. Saarela, “Organization of a massive document collection,” *IEEE Trans. on Neural Networks*, vol. 11, no. 3, pp. 574–585, May 2000.
- [7] P. J. Huber, *Robust Statistics*, New York: J. Wiley, 1981.

- [8] E. L. Lehmann, *Theory of Point Estimation*, New York: J. Wiley, 1983.
- [9] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, New York: J. Wiley, 1986.
- [10] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*, MA: Kluwer Academic Publishers, 1990.
- [11] C. Kotropoulos, I. Pitas, and M. Gabbouj, “Marginal median learning vector quantizer,” in *Proc. European Signal Processing Conference*, August 1994, pp. 1496–1499.
- [12] I. Pitas, C. Kotropoulos, N. Nikolaidis, R. Yang, and M. Gabbouj, “Order statistics learning vector quantizer,” *IEEE Trans. on Image Processing*, vol. 5, no. 6, pp. 1048–1053, June 1996.
- [13] V. Barnett, “The ordering of multivariate data,” *J. R. Statist. Soc. A*, vol. 139, no. 3, pp. 318–354, 1976.
- [14] A. Georgakis, C. Kotropoulos, N. Bassiou, and I. Pitas, “Hypergeo: A data organization and retrieval system for tourist information,” in *IASTED Int. Conf. on Applied Informatics*, February 2001, pp. 719–724.
- [15] D. Manning and H. Schütze, *Foundation of Statistical Natural Language Processing*, Cambridge, MA: MIT Press, 1999.
- [16] W. B. Frakes and R. Baeza-Yates, *Information Retrieval: Data Structures and Algorithms*, Upper Saddle River: Prentice-Hall, 1992.
- [17] R. Krovetz, “Viewing morphology as an inference process,” in *Proc. SIGIR’93*, 1993, pp. 191–203.
- [18] M.F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.
- [19] P. Clarkson and R. Rosenfeld, “Statistical language modeling using the CMU-Cambridge toolkit,” in *Proc. of Eurospeech’97*, 1997, pp. 2707–2710.
- [20] M. P. Oak, *Statistics for Corpus Linguistics*, Edinburgh: Univ. Press, 1998.
- [21] C. Becchetti and L. P. Ricotti, *Speech Recognition: Theory and C++ Implementation*, New York: J. Wiley, 1999.
- [22] S. Kaski, “Dimensionality reduction by random mapping: Fast similarity computation for clustering,” in *IJCNN*, IEEE, 1998, vol. 1, pp. 413–418.
- [23] R. R. Korfhage, *Information Storage and Retrieval*, New York: J. Wiley, 1997.