Authors:	Saarikoski Jyri, Järvelin Kalervo, Laurikkala Jorma, Juhola Martti
Name of article:	On Document Classification with Self-Organising Maps
Name of work:	Adaptive and Natural Computing Algorithms: 9th International Conference, ICANNGA 2009, Kuopio, Finland, April 23-25, 2009, Revised Selected Papers
Editors of work:	Kolehmainen Mikko, Toivanen Pekka and Beliczynski Bartlomiej
Year of publication:	2009
ISBN:	978-3-642-04920-0
Publisher:	Springer
Pages:	140-149
Series name and number:	Lecture Notes in Computer Science 5495
ISSN:	0302-9743
Discipline:	Natural sciences / Computer and information sciences
Language:	en
School/Other Unit:	School of Information Sciences

URL: http://www.springerlink.com/content/421846m64qh75062/fulltext.pdf
URN: http://urn.fi/urn:nbn:uta-3-867
DOI: http://dx.doi.org/10.1007/978-3-642-04921-7

Additional infromation:

The original publication is available at www.springerlink.com.

Prepring from: Saarikoski, J, & Järvelin, K. & Laurikkala, J. & Juhola, M (2009). On Document Classification with Self-Organising Maps. In M. Kolehmainen et al. (Eds.): Proceedings of ICANNGA 2009. Heidelberg: Springer, Lecture Notes in Computer Science vol. 5495, pp. 140–149.

On document classification with self-organising maps

Jyri Saarikoski[1] • Kalervo Järvelin[2] • Jorma Laurikkala[1] • Martti Juhola[1]

[1]Department of Computer Sciences and [2]Department of Information Studies,
33014 University of Tampere, Finland
{Jyri.Saarikoski, Kalervo.Jarvelin, Jorma.Laurikkala, Martti.Juhola}@uta.fi

Abstract This research deals with the use of self-organising maps for the classification of text documents. The aim was to classify documents to separate classes according to their topics. We therefore constructed self-organising maps that were effective for this task and tested them with German newspaper documents. We compared the results gained to those of *k* nearest neighbour searching and *k*-means clustering. For five and ten classes, the self-organising maps were better yielding as high average classification accuracies as 88-89%, whereas nearest neighbour searching gave 74-83% and *k*-means clustering 72-79% as their highest accuracies.

1. Introduction

The growth of digital documents and information stored as text in the Internet has been rapid in the recent years. Searching and grouping such documents in various ways have become an important and challenging function. A myriad of documents are daily accessed in the Internet to find interesting and applicable information. Distinguishing in some way interesting documents from the uninteresting ones is, even if a self-evident goal, crucial. For this purpose, computational methods are of paramount importance. We are interested in researching the classification of text documents, also those written in languages other than English.

There are known methods for constructing groups, clusters or models of documents, see for instance (Doan et. al., 2003; Sebastiani, 2002; Serrano and del Castillo, 2007). These machine learning methods have included *k* nearest neighbour searching, probabilistic methods such as Naïve Bayesian classifiers (Duda et al., 2001) and evolutionary learning with genetic algorithms (Serrano and del Castillo, 2007). The methods were of the supervised category. We investigated the use of unsupervised Kohonen self-organising maps (Kohonen, 1995) that seemed to be seldom used in this field. They have been, however, applied to constructing visual maps of text document clusters, in which documents were clustered based on the features they contain. WEBSOM (Honkela, 1997; Lagus et al., 2004) was employed to organize large document collections, but it did not include document classification in the sense to be compared with the current research. Chowdhury and Saha (2005) classified 400, 500 and 600 sport articles, whereas Guerro-Bote et al. (2002) employed 202 documents from a bibliographic database. Moya-Anegón et al. (2006) made domain analysis of documents with self-organising maps, clustering and multidimensional scaling. Instead of document clustering, we were interested in investigating how accurately and reliably self-organising maps are able to classify documents. Therefore, we constructed self-organising maps on document sets belonging to different known classes and used them to classify new documents. We employed ten-fold crossvalidation runs on our test document collection to assess classification accuracy in the document collection. We also performed comparable tests with *k* nearest neighbour searching and *k*-means clustering which employ supervised learning to find a baseline level for the classification of the document data used. In principle, the use of self-organising maps is reasonable, because outside laboratory tests there is not necessarily a reliably classified learning set available.

In the present research, we extend our previous research of using self-organising maps for information retrieval in the same German document collection as in (Saarikoski et al., 2008). In the prior work, we studied retrieval from the document collection, the topics of which were associated with some of its documents, and we used both relevant and non-relevant documents in the document sample extracted from the collection. In the present work, our interest was in the classification, in other words separation between document classes. We therefore used only documents relevant to the classes examined.

2. The data and its preprocessing

We used a German document set which was taken from an original collection of 294809 documents (Airio, 2006) from CLEF 2003 of the years 1994 and 1995 (http://www.clef-campaign.org/). The articles were from newspapers such as Frankfurter Allgemeine and Der Spiegel. There were 60 test topics associated with the collection. In every topic there was a relatively small subset of relevant documents. Relevant topics were included in our tests. At first, 20 topics were taken from the 60 topics otherwise randomly. From those 20 selected the smallest classes (topics) were still left out which included 6-25 relevant documents in the collection. Such small document classes would not have been quite reasonable for 10-fold crossvalidation tests, because their average numbers of test documents in test sets would only have been from 0.6 to 2.5, which might have resulted in considerable random influence. Thus, we attained 10 topics (classes) and 425 relevant documents (observations) so that the numbers of the relevant documents of the topics were 27, 28, 29, 29, 34, 39, 44, 53, 55 and 87.

The concept of relevance means here that the association of the documents to the topics had been manually ensured in advance by independent evaluators who had nothing to do with the present research.

To transform pertinent document data into the input variable form for a self-organising map, some preprocessing was required. At first, the German stemmer called SNOWBALL was run to detect word stems like ‘gegenteil’ from ‘Gegenteil’ or ‘gegenteilig’ from all documents and topics chosen. In addition, a list of 1320 German stopwords was used to sieve semantically useless words from them. Stopwords are prepositions like ‘ab’, articles like ‘ein’ and ‘eine’ or pronouns like ‘alle’, adverbs or other uninteresting “small words”, which are mostly uninflected words. They were removed from the documents and topic texts. Thereafter, short words, shorter than four letters, were removed, because they are typically, after stemming, as word prefices rather useless as term words. The last preprocessing phase included the computation of the frequencies of remaining word stems.

The documents and topics were of SGML format. In the following, the first example presents an (abbreviated) SGML document and the second example depicts a topic connected to some other documents. Classification variables were formed on the basis of words occurring in the actual text parts of the documents and topics.

A document:

```
<DOC>
<TITLE>Ahornblatt nach 33 Jahren vergoldet</TITLE>
<TITLE>Zum 20. Mal Eishockey-Weltmeister</TITLE>
<TITLE>Sieg im Penaltyschießen</TITLE>
<TITLE>Finnland </TITLE>
<TEXT>Das Eishockey-Mutterland Kanada ist nach 33 Jahren wieder die Nummer eins in der Welt.
Durch einen 3:2-Erfolg im Penaltyschießen gegen Finnland lösten die Ahornblätter im WM-Finale in
Mailand den einst übermächtigen Rivalen und Titelverteidiger Rußland ab, der bereits im Viertelfinale
gegen die USA (1:3) ausgeschieden war. Nach regulärer Spielzeit und Verlängerung hatte es 1:1 (0:0,
0:0, 1:1, 0:0) gestanden. Zuvor hatte Brind'Amour (56.) die Führung der Finnen durch Keskinen (47.)
ausgeglichen. Im Penaltyschießen zeigten die Kanadier die besseren Nerven, die Finnen verschossen
viermal in sechs Versuchen. Robitaille verwandelte den sechsten Penalty für Kanada. Die Kanadier,
zuletzt 1961 bei der WM in Genf und Lausanne auf dem Thron, feierten bei den 58. Titelkämpfen ihren
20. WM-Titel und machten damit...
</TEXT>
</DOC>
```

A topic:

```
<DE-title> Rechte des Kindes </DE-title>
<DE-desc> Finde Informationen über die UN-Kinderrechtskonvention. </DE-desc>
```

We computed document vectors for all documents by applying the common vector space model with *tf·idf* weighting for all remaining word stems. Thus, a document is presented in the following form

$$D_i = (w_{i1}, w_{i2}, w_{i3}, ..., w_{it}) \tag{1}$$

where w_{ik} is the weight of word k in document D_i, $1 \le i \le n$, $1 \le k \le t$, where n is the number of the documents and t is the number of the remaining word stems in all documents. Weights are given in *tf·idf* form as the product of term frequency (*tf*) and inverse document frequency (*idf*). The former for word k in document D_i is computed with

$$tf_{ik} = \frac{freq_{ik}}{\max_l \{freq_{il}\}} \tag{2}$$

where $freq_{ik}$ equals the number of the occurrences of word k in document D_i and l is for all words of D_i, l=1,2,3,..., t-1, t. The latter is computed for word k in the document set with

$$idf_k = \log \frac{N}{n_k} \tag{3}$$

where N is equal to the number of the documents in the set and n_k is the number of the documents, which contain word k at least once. Combining equations (2) and (3) we obtain a weight for word k in document D_i

$$w_{ik} = tf_{ik} \cdot idf_k \tag{4}$$

Based on this computation all 425 documents were mapped as document vectors weighted with the *tf·idf* form.

Finally, the length of each document vector was shortened only to include 500 or alternatively 1000 middle (around median) word stems from the total word frequency distribution increasingly sorted. Very often the most and least frequent words are pruned in information retrieval applications, because their capacity to distinguish relevant and non-relevant documents (to a topic) is known to be poor. We chose either 500 or 1000 words, since from several values we found these as good choices for this data in our earlier research (Saarikoski et al., 2008).

It is worth noticing that document vectors were only computed from a learning set in crossvalidation. Information about its corresponding test set was not used in order to create as a realistic situation as possible, where the system knows an existing learning set and its words in advance, but not those of a test set. Thus, each learning set included its own word set, somewhat different from those of the other learning sets, and the document vectors of its corresponding test set were prepared according to the words of the learning set.

3. Classification with self-organising maps

Kohonen self-organising maps are neural networks that apply unsupervised learning and they have been exploited for numerous visualisation and categorisation tasks (Duda et al., 2001). We employed them to study their applicability to divide the test documents into different classes on the basis of document vectors computed. We used the SOM_PAK program written in C (http://www.cis.hut.fi/projects/somtoolbox/) in Helsinki University of Technology, Finland.

In our previous research on the same German document collection (Saarikoski et al., 2008), we observed that random initialisation, bubble neighbourhood and up to 17×17 nodes were good choices. Different numbers of learning epochs were tested. Finally, as few as 3 coarse and 15 tuning epochs were applied.

The following procedure was implemented.

1. Create a self-organising map using a learning data set.
2. Form the model vector of a node during the learning process of the network. Its dimension is equal to that of the input vectors.
3. Determine a class for a node of the map according the numbers of documents of different classes in the current node. The most frequent document class determines

the class of the node. If there are more than one class with the same maximum, label the node according to the class of the document (from the maximum classes) closest to the model vector (learnt during the process) of the node. Consider all nodes in this manner.

After this procedure each node corresponded to some document class. Some node could also remain empty, which would be bypassed during the later process.

Next the classification of a test document set was performed where a test document was compared to the model vector of each node to find which node was the closest (the best fit), on the basis of Euclidean distance, to the test document.

After computing all document vectors of a test set, classification accuracy was computed by checking for every document of a test set *j* whether it was classified into its correct class.

$$a_j = \frac{c_j}{n_j} 100\% \tag{5}$$

Here c_j (j=1,..,10) is equal to the number of the correctly classified documents in test set *j* and n_j is the number of all documents in that test set. Accuracy a_j was obtained for each test set. Since a random element is involved in the initialisations of neural networks, we repeated 10 tests for every learning and test set pair. For each such crossvalidation pair about 90% of documents were put to a learning set and the rest 10% to its corresponding test set. Documents were selected into learning sets and test sets so that the relative proportions of various kinds of documents were similar in both sets. Thus, 10-fold crossvalidation was applied, which produced 10 times 10 test runs for a test document set. Average classification accuracies were finally calculated from those 100 runs.

4. Nearest neighbour searching and *k*-means clustering

In order to compare results obtained by self-organising maps, we tested with nearest neighbour searching and *k*-means clustering by using exactly the same crossvalidation document selections as above for the documents.

Classification with nearest neighbour searching was performed with the following procedure.

1. Search for *k* nearest neighbours of a test document from a learning set.
2. Compute the majority class from those *k* documents, i.e. the most frequent document class among the neighbours.
3. Determine the class of the text document on the basis of the preceding step. If there are two or more classes including the same maximum number of documents, select the class randomly from those majority classes.
4. Repeat the former steps for all documents of a test set.

After the nearest neighbour searching, the classification results were assessed for correctness. Values of *k* were 1, 3, 5, 7 and 9. The Euclidean distance measure was applied. The procedure was run for all 10 pairs of the learning and test sets, for which average classification accuracies were calculated. We employed the Matlab program. Nearest neighbour searching included no such an initialisation property of random character as self-organising maps and clustering. Consequently, the nearest neighbour searching was run only once for every learning and test set pair.

Clustering was accomplished with the Matlab program according to the test protocol similar to that of nearest neighbour searching. The documents of a learning set were clustered into *k* clusters in the Euclidean space of the document vector variables, when *k* was equal to 2, 5, 10 and 20. The class of each cluster was determined similarly to the above "voting" principle of nearest neighbour searching. A test set was then dealt with and results computed. This was done 10 times for all 10 learning and test sets to obtain the average results.

5. Results

We tested with the two input vector lengths, 500 and 1000 word stems, either 2, 5 or 10 classes (topics), which respectively included 142, 278 or 425 relevant documents in total. Less than 10 classes (5 or 2 largest classes) were tested in order to see what may happen when we merely restricted ourselves to the largest document classes, i.e. discarded the classes smaller than with 39 or 55 documents. In the following, we present the means and standard deviations of 100 crossvalidation test runs of the self-organising maps and *k*-means clustering and those of 10 crossvalidation runs of nearest neighbour searching. The crossvalidation division into test and learning sets was identical between all three machine learning methods used.

Table 1 shows the results computed with the self-organising maps. The highest result at each row is written in bold in Tables 1-3. The best 2-class and 5-class situations in Table 1 were with the smallest network of the 25 nodes. Instead, the 10-class condition gave its best results with the networks of 7×7 nodes. The vector lengths used did not yield so unambiguous an outcome. For the self-organising maps, 4.8% of all nodes as minimum were empty with the size of 5×5 nodes and 5 classes. As maximum 66.9% were empty with the size of 13×13 and 2 classes. These empty nodes obtained hits (incorrect classifications) from 0.8% (10 classes) to 5.0% (2 classes) both with the size of 5×5.

Table 2 presents the results of nearest neighbour searching. Its results of all 2-class test alternatives were exceptionally high. This was at least partly due to very different topics of the two classes one being 'children theme' and the other 'nuclear power theme'. The 5-class and 10-class situations were at their best with nearest neighbour searching of *k* equal to 1. For the 2-class alternatives the longer vector length of 1000 word stems produced better results than the shorter length of 500, but for the 5-class and 10-class alternatives it was vice versa.

The numbers of 2, 5, 10, 20, 40, 60, 80, 100 and 120 clusters were tested for clustering. Table 3 describes most clustering results excluding those of 40, 60, 100 and 120 clusters since these were poorer than the results of 80 clusters. The best results were gained by using the cluster number of 80, except for the 2-class condition. The shorter vectors were better than the longer ones.

Running times of individual learning and test pairs were moderate while using a computer with a 1.6 GHz processor and 1 GB memory. They varied from 1.6 s to 13 s for the self-organising maps. The Matlab implementation of nearest neighbour searching took from 0.4 s to 1.1 s and that of *k*-means clustering from 1.8 s to 34 s. These do not contain the short time of the preprocessing common to all three.

Table 1. Means and standard deviations of classification accuracies (%) of self-organising maps for 100 test runs

Number of classes	Vector length	Number of nodes				
		5×5	7×7	9×9	11×11	13×13
2	500	**93.2±8.2**	88.4±10.2	77.4±11.1	68.8±11.5	60.6±13.8
	1000	**90.5±8.0**	86.4±9.5	75.7±11.5	67.2±13.1	62.5±14.3
5	500	**87.8±6.2**	86.0±6.9	84.3±7.1	77.5±6.6	73.4±7.9
	1000	**89.0±6.8**	87.0±5.9	83.2±7.3	78.1±7.5	72.2±8.2
10	500	79.2±7.3	**88.1±5.6**	86.7±5.8	82.6±6.6	79.6±6.3
	1000	76.5±5.1	**89.2±5.4**	88.0±5.4	84.2±4.8	80.8±5.6

Table 2. Means and standard deviations of classification accuracies (%) of nearest neighbour searching for 10 test runs

Number of classes	Vector length	Number *k* of nearest neighbours				
		1	3	5	7	9
2	500	95.1±4.6	97.1±3.7	95.8±3.6	97.9±3.4	**99.2±2.4**
	1000	**99.3±2.1**	**99.3±2.1**	**99.3±2.1**	98.7±4.2	98.7±2.8
5	500	**83.4±6.2**	76.3±7.1	69.0±6.7	70.5±5.4	69.7±8.4
	1000	**74.4±5.5**	60.8±7.7	56.5±9.9	59.0±7.7	59.4±10.9
10	500	**83.3±7.0**	81.8±6.1	80.2±6.8	78.9±5.7	78.5±5.7
	1000	**80.7±6.0**	72.6±6.2	69.7±5.7	67.9±6.9	71.1±5.7

Table 3. Means and standard deviations of classification accuracies (%) of *k*-means clustering for 100 test runs

Number of classes	Vector length	Number *k* of clusters				
		2	5	10	20	80
2	500	62.1±5.7	73.7±17.7	92.4±14.7	**97.9±7.0**	95.9±5.9
	1000	61.3±1.6	65.0±11.2	76.9±18.3	83.6±17.7	**91.9±9.7**
5	500		52.0±6.9	59.2±7.4	65.5±6.0	**78.5±7.3**
	1000		44.6±9.8	54.4±7.1	58.8±6.6	**72.3±6.9**
10	500			48.1±5.7	56.9±7.3	**73.6±8.3**
	1000			42.7±5.7	52.1±5.7	**71.9±6.2**

Fig. 1 shows an example of the self-organising maps. It includes 10 classes with 383 documents of a learning set when the size of the map was 7×7, the input vector length was 1000 and a random test run was chosen. Its average classification accuracy was 88.8%.

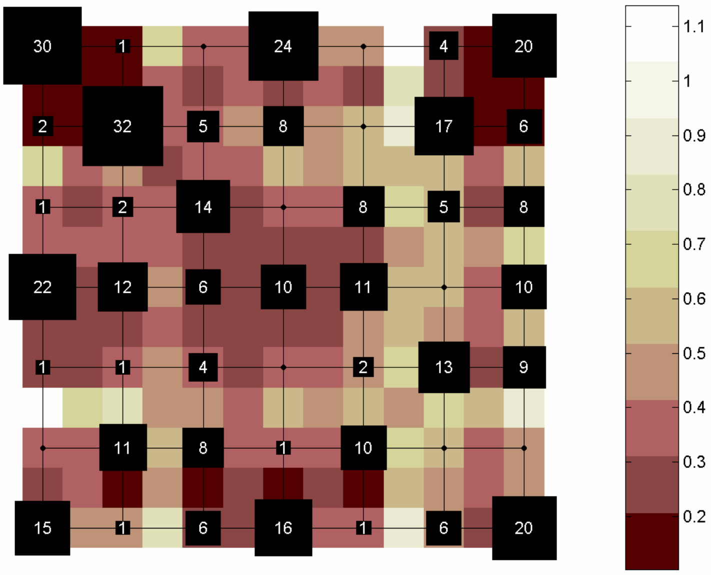

Fig. 1. The numbers of relevant documents of a learning set hit each node are counted in the map. The darker the node, the more compact the concentration of the document group is. The larger the node, the greater the number of documents. The other 42 documents of all 425 documents were not here, but allocated to the test set.

Fig. 2 depicts the same map as Fig. 1, but the nodes are marked with the class identifiers computed. The following list gives the class identifiers, numbers of documents and class titles occurring in Fig 2.

#186 : 24 : Holländische Regierungskoalition
#156 : 25 : Gewerkschaften in Europa
#147 : 26 : Ölunfälle und Vögel
#195 : 26 : Streik italienischer Flugbegleiter
#193 : 31 : EU und baltische Länder
#184 : 35 : Mutterschaftsurlaub in Europa
#150 : 40 : AI gegen Todesstrafe
#152 : 48 : Rechte des Kindes
#190 : 50 : Kinderarbeit in Asien
#187 : 78 : Atomtransporte in Deutschland

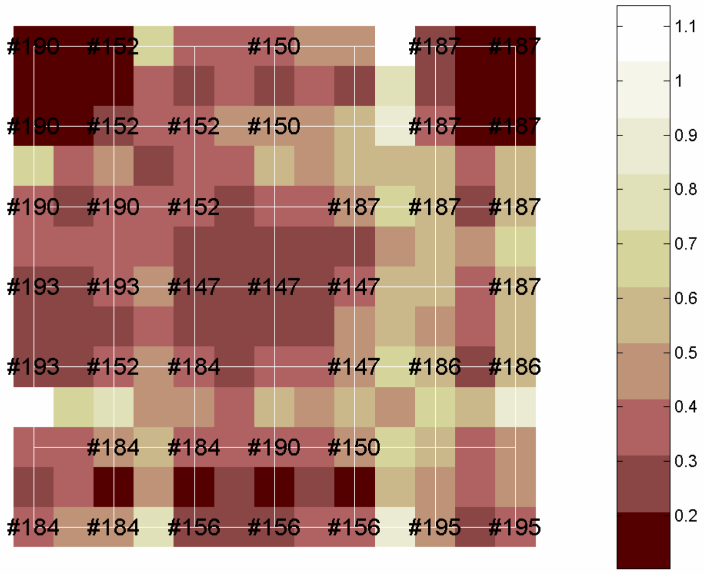

Fig. 2. The class identifiers are attached to the nodes where they beat voting as "majority" classes. Notice that we cannot sum up the numbers of documents from this figure and the preceding list and to compare them directly to those of Fig.1, because the nodes also include some probably incorrect (non-relevant) classifications from "minority" classes.

To statistically compare the results, the Friedman test (Conover, 1999) was conducted. Since nearest neighbour searching included 10, but the others 100 test runs, the means of the 10 crossvalidations of the latter two methods were first calculated. For the 2-class condition nearest neighbour searching and clustering obtained significantly (p = 0.004) better results than the self-organising maps for the vector length of 500. For the length of 1000, nearest neighbour searching was significantly (p = 0.00005) better. For the 5-class and 10-class conditions, the self-organising maps outperformed significantly (p < 0.001) the other methods with both vector lengths.

6. Conclusions

We tested self-organising maps, nearest neighbour searching and *k*-means clustering with documents from a German newspaper article collection. Except the 2-class alternative which favoured nearest neighbour searching, self-organising maps gave the best results. Table 1 suggests that if more classes are involved, the number of the nodes in a network should increase. On the other hand, for nearest neighbour searching the dispersion of documents to several classes supports the idea to keep to the number *k* of neighbours equal to 1. Table 3 (*k*-means) suggests that the number of the cluster is best to set high. Differences caused by the vector lengths were not consistent, but the self-organising maps were mostly somewhat better with the length of 1000 word stems, meanwhile nearest neighbour searching and *k*-means clustering favoured the length of 500. Doubtless the self-organising maps were effective classifiers for the current data. Excluding the 2-class condition, they outperformed the other two methods, when the self-organising maps gave the average classification accuracies of 88-89%, nearest neighbour searching reached 74-83% and clustering 72-79%. A 2-class condition is an extreme situation. A more realistic alternative contains a greater number of classes. Nearest neighbour searching was the fastest method.

We can continue our research with the current document data and larger document sets. We are going to perform an extensive analysis with additional learning methods.

Acknowledgements

The research was partially funded by Alfred Kordelin Fund and Academy of Finland, projects 120264, 115609 and 124131. SNOWBALL stemmer was by Martin Porter.

References

Airio, E. (2006). Word normalization and decompounding in mono- and bilingual IR. *Information Retrieval*, *9*(3), 249-271.

Chowdhury, N., & Saha, D. (2005). Unsupervised text classification using Kohonen's self organizing network. In *Computational Linguistics and Intelligent Text Processing* (pp. 715-718). Springer-Verlag, Lecture Notes in Computer Science 3406.

Conover, W. J. (1999). *Practical Nonparametric Statistics*, John Wiley & Sons, New York.

Doan, A., Domingos, P. Halevy, A. (2003). Learning to match the schemas of data sources: a multistrategy approach. *Machine Learning, 50,* 279-301.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern Classification,* second ed. John Wiley & Sons, New York.

Guerro-Bote, V. P., Moya-Anegón, F., & Herrero-Solana, V. (2002). Document organization using Kohonen's algorithm. *Information Processing and Management, 38,* 79-89.

Honkela, T. (1997). *Self-Organizing Maps in Natural Language Processing*, Academic Dissertation, Helsinki University of Technology, Finland.

Kohonen, T. (1995), *Self-Organizing Maps,* Springer-Verlag, Berlin.

Lagus, K., Kaski, S., & Kohonen, T. (2004). Mining massive document collections by the WEBSOM method. *Information Sciences, 163*(1-3), 135-156.

Moya-Anegón, F., Herrero-Solana, V., & Jiménez-Contreras, E. (2006). A connectionist and multivariate approach to science maps: the SOM, clustering and MDS applied to library and information science research. *Journal of Information Science, 32*(1), 63-77.

Saarikoski, J., Laurikkala, J., Järvelin, K., & Juhola, M. (2008). A study on the use of self-organising maps in information retrieval. To appear in *Journal of Documentation.*

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1-47.

Serrano, J. I., & del Castillo, M. D. (2007). Evolutionary learning of document categories. *Information Retrieval, 10,* 69-83.