Neurocomputing 71 (2008) 3353–3359

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A quickly trainable hybrid SOM-based document organization system

Renato Fernandes Corrêa, Teresa Bernarda Ludermir *

Center of Informatics, Federal University of Pernambuco, P.O. Box 7851, Cidade Universitária, 50.732-970 Recife-PE, Brazil

ARTICLE INFO

Available online 12 July 2008

Keywords:
Hybrid system
Document organization
Self-organizing map
k-Means
Modified leader algorithm

ABSTRACT

The large volume of nowadays document collections has increased the need of fast trainable document organization systems. This paper presents and evaluates a hybrid system to self-organization of massive document collections based on self-organizing map (SOM). The hybrid system uses prototypes generated by a clustering algorithm to train the document maps, thus reducing the training time of large maps. We test the system with k-means and modified leader clustering algorithms. The experiments are carried out with the Reuters-21758 v1.0 and 20 Newsgroup collections. The performance of the system is measured in terms of text categorization effectiveness on test set and training time. Experimental results show that the proposed system generates effective document maps in less time than SOM. However, the hybrid system using k-means generates better document maps than the one using modified leader at the cost of more long training time.

1. Introduction

Large digital document collections have become even more common on all sectors of the modern society. Thus it is necessary that even more fast trainable and effective systems to automatically organize and allow knowledge discovery over document collections.

Self-organizing map (SOM)-based document organization system [16] can be defined as a system that automatically organizes a collection of documents in groups of similarity using SOM [15], generating a document map.

The document map and their graphical representation provide means to explore large collections of texts by enabling an alternation between visualization, zooming in interesting information, browsing, and searching for a specific item [16].

A problem that can make difficult the application of SOM to document organization of large documents collections is their computation time complexity.

In the especial case of SOM algorithm, this problem has been addressed by WEBSOM project [16]. The WEBSOM methodology does scale up well even to very large datasets due to: (i) the use of random mapping, a fast dimensionality reduction method, (ii) the use of shortcuts in computation of SOM algorithm, and (iii) the use of a method to estimate large maps from trained small ones, progressively increasing the size of the SOM archive [16]. However, as in all SOM-based systems, WEBSOM's major drawback is the huge amount of training time and resources required for training of the document map.

In WEBSOM, size reduction efforts on dataset have been concentrated mainly on the number of dimensions. Azcarraga and Yap demonstrated in [2] that the volume (number of documents vectors) could also generate a drastic reduction. They improved the WEBSOM methodology by adding a size reduction phase, so-called volume reduction phase. The SOM archive is initially trained using representative vectors, called prototypes, and the whole (large) set of document vectors are loaded only, once the SOM training is completed. This makes for drastically reduced training time and computer memory requirement of SOM training. However, Azcarraga and Yap only suggest the use of a proposed prototype generator method, here called AY Method, and do not report a training time analysis and correct evaluation of the quality of the document maps generated by their system.

We generalize the methodology proposed by Azcarraga and Yap and propose and evaluate a hybrid SOM-based document organization system architecture [5]. We found that: (i) the proposed system with the k-means clustering is more efficient and faster than the same system with the AY method, (ii) the upper bound of the number of prototypes generated by the prototype generator method is the number of nodes desired in the document map, and (iii) the use of prototypes generated by AY method reduces the training time of the SOM, but the AY method requires a long training time, making the training of the hybrid system more time-consuming than the training of the similar SOM system.

The objectives of this paper are to extend the specification of the proposed hybrid system architecture present in [5] and to report the hybrid system behavior with volume reduction

* Corresponding author. Tel.: +55 81 21268430; fax: +55 81 21268438.
E-mail address: tbl@cin.ufpe.br (T.B. Ludermir).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.02.021

performed by k-means and modified leader clustering algorithms. The experiments are carried out with the Reuters-21758 v1.0 and 20 Newsgroup text collections. Text categorization effectiveness on test set and training time are used to measure the performance of the system.

2. Hybrid self-organization of document collections

The proposed hybrid SOM-based document organization system performs the five steps presented in Fig. 1.

The document indexing step consists in preprocessing the text documents to represent them statistically. Generally, non-informative words are removed from initial vocabulary and word affixes are removed using a stemmer algorithm [24]. The isolated words without affixes are called terms. The documents are represented using the vector space model [24]. Vectors represent documents where terms are the indexes and the corresponding values are the importance of a term to the meaning of a document. A function of the term's frequency of occurrence in a document is used to approximate the importance of a term in a document.

Dimensionality reduction step receives the document vectors generated in the document indexing step and applies some algorithms to reduce the number of dimensions or terms. There are many methods to dimensionality reduction, see [5] for more details.

The volume reduction step consists of training a clustering algorithm with the reduced document vectors obtained from the dimensionality reduction step. The vectors representing each cluster are taken as prototypes and represent samples or patterns mapped to the cluster. Clustering algorithms used to prototype generator must have linear time complexity in the size of the document collection. The clustering algorithms used in the experiments, k-means and modified leader, have this property and are described in the next section. Other examples of algorithms with this property are [10]: BIRCH, CURE, STING, and CLIQUE.

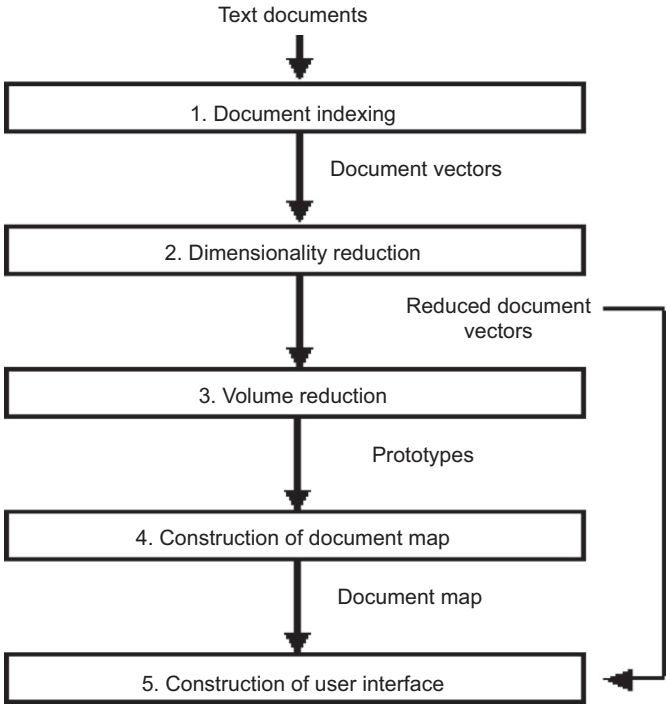


Fig. 1. Overview of a hybrid SOM-based document organization system construction.

Let the time complexity of the clustering algorithm be $O(ndk)$, where n is the number of training documents vectors, d is the dimensionality of the vectors, and k is the number of prototypes. Let M be the number of nodes in the SOM map. The time complexity of the hybrid system is $O(ndk)+O(kdM)$, where the terms are the time complexity of the clustering algorithm and SOM training, respectively. To obtain a hybrid system with significantly smaller training time than the analogous $O(ndM)$ SOM system, the number of prototypes must be less than or equal to the number of nodes in the SOM map.

Prototype vectors are used as input to the step of the construction of the document map. Document map construction step consists in training SOM map with the input vectors. The training may be done in one stage, or multiple stages. The one-stage training consists of training a random initialized map with SOM until it reaches stationary state. The multiple-stage training consists initially in one-stage training of a small map and after that performs multiple stages of estimation of initial state of a large map based on stationary state of a small one followed by fine-tuning of the estimated large map. The goal of fine-tuning is to the large maps reaching stationary state. The WEBSOM method presents a multiple-stage training where the computational time complexity of each estimation-fine-tuning stage is $O(dn)+O(n^2)$ [16], where n is the number of training documents vectors, d is the dimensionality of the vectors, and M is the number of nodes in the large map that is assumed to be about one-tenth of n. The time complexity of one-stage training is $O(ndM)$ and in multiple-stage training is $O(ndm)+O(dn)+O(n^2)$, where m is the number of nodes in the small map (a few hundred nodes in WEBSOM). For simplicity of experiment's execution, we perform one-stage training of SOM map.

Construction of the user interface is the last step. The user interface must allow interactive browsing, content-addressable and keyword searching and visualizing the searches' results over the document map. Details about the interface construction can be viewed in [16].

The construction of document map is the main step, due to the great influence on overall performance of the document organization system and consequently of the SOM-based information retrieval system. The map must organize the documents by generating consistent clusters. In good quality document maps, similar document vectors must be mapped to the same node or neighboring nodes.

Classification accuracy and text categorization effectiveness measures [24] are recommended to evaluate the quality of the document map because they express how the map captures the document similarity in a close way to the human being. The use of these measures presupposes the use of a manual categorized text collection. The quality of the document map is a consequence of all the steps performed before the interface construction of the document map.

3. Prototype generator algorithms

This section describes the clustering algorithms used to prototype generation in the experiments. We choose the modified leader algorithm because it is the simplest clustering algorithm, and the k-means because it is a well-known and efficient algorithm. Both prototype generator algorithms have linear time complexity in the size of the training set [14].

3.1. k-Means

k-Means [11] is an iterative algorithm that minimizes a dissimilarity criterion function. The original k-means algorithm

uses Euclidean distance between vectors, thus it minimizes the least-squares error criterion [14]. We employ a variation of k-means algorithm using cosine similarity measure (cosine of the angle between two vectors). The cosine variation of k-means minimizes the sum of least one complement of the cosine criterion. We use the cosine variation of k-means because it has generated better results in text clustering than the original algorithm [25]. The cosine measure seems to capture better the similarity of content between documents represented by vectors than Euclidean distance.

In k-means, each cluster is represented by its center (the mean of all input patterns mapped to it). The centers are initialized with a random selection of k patterns. Each input pattern is then labeled with the index j of the nearest or the most similar center. Subsequent recomputing of the mean for each cluster and re-assignment of cluster labels are iterated until convergence to a fixed labeling after t iterations or epochs. The complexity of this algorithm is $O(ndk)$, where n is the number of patterns in the training set, d is the number of features for each pattern and k is the desired number of clusters.

k-Means is the most popular clustering algorithm. The reasons behind the popularity are manifold: (i) it is easy to implement, (ii) its linear time complexity in the size of the training set, and (iii) it is order-insensitive—for a given initial seed set of cluster centers, it generates the same partition of the data irrespective of the order in which the patterns are presented to the algorithm.

The drawbacks of this algorithm are: (i) it is sensitive to the selection of the initial partition or initial seed selection, (ii) it may converge to a local minimum of the criterion function value if the initial partition is not properly chosen, and (iii) even in the best case, it can produce only hiperspherical clusters.

3.2. Modified leader algorithm

The leader algorithm [11] is a very fast method for clustering data, the simplest in terms of training time. It requires one pass through the data to put each input pattern in a particular cluster or group of patterns. Associated with each cluster is a ‘‘leader’’, which is one pattern against which new patterns will be compared to determine whether the new pattern belongs to this particular cluster.

Essentially, the leader algorithm starts off with zero prototypes and adds a prototype whenever none of the existing prototypes is close enough to a current input pattern. The newly created prototype is an exact copy of the current input pattern and is called ‘‘leader’’ of that cluster. The cosine of the angle between the input vector and each prototype is used as a similarity measure. The influence threshold is a parameter of the system and determines how similar the best matching prototype should be for it to be considered ‘‘close enough’’, its value ranges from 0 to 1. In cases when some existing prototype is sufficiently close to the current input pattern, the input pattern is mapped to that cluster.

The first pattern presented will always be the leader of the first cluster. The second pattern will be compared to the leader of the first cluster. If the second pattern is close enough to this leader (as determined by a supplied threshold), the second pattern is mapped to the first cluster. If the second pattern is not close enough to the first leader (again, as determined by a supplied threshold) then the second pattern will become the leader of the second cluster. The next pattern will then be compared to the leader of the first cluster, and if close enough, is mapped to the first cluster, and if not close enough, compared to the leader of the next cluster. Each pattern will be mapped to a cluster or, after having been compared to the existing cluster leaders and found to be not close enough to any of them, will become the leader of a new cluster. The next pattern goes through the same process and so on, until each pattern has been mapped to some cluster.

This algorithm requires one pass through the data for every pattern to be mapped to a cluster and is, thus, a very fast algorithm. On the other hand, the algorithm is sensitive to the presentation order of patterns. For example, the first pattern presented will always be a cluster leader. In addition, the clusters that are created first will tend to be very large since a pattern will always be compared to them first and be mapped to the first cluster to which it is close enough.

The modified leader algorithm [11] attempts to deal with one of the major problems of the leader algorithm described above. To determine the cluster that a new pattern will be mapped to, the algorithm will search for the closest cluster leader to the pattern (according to some user-defined distance metric). If that closest cluster leader is close enough to the new pattern (as determined by a user-supplied threshold), then the new pattern belongs to that cluster. If the cluster leader is not close enough to the new pattern (again, as determined by the user-supplied threshold), the new pattern becomes the leader for a new cluster. In this way, each cluster has an equal chance at having the new pattern fit into it rather than clusters that are created earlier having an undue advantage. Of course, this algorithm is still not invariant to the presentation order, since the first pattern will again always be the leader of the first cluster. This algorithm is also slower than the simple leader algorithm since distances between the pattern and every cluster leader must be calculated before the pattern can be mapped to any cluster.

In both versions of the leader algorithm, the user must guess the correct threshold level and various threshold values may yield completely different results. It may not be completely clear which is an optimal threshold level and often measures of goodness of clusters require some sort of trade off to be made. The choice of this threshold is critical. A very large threshold will result in all patterns assigned to one cluster. A very small threshold will result in each pattern assigned to its own individual cluster. To find the natural clusters inherent in the data, the threshold must be larger than the typical within cluster distance and smaller than the typical between cluster distances.

The complexity of both versions of the leader algorithm is $O(ndk)$, where n is the number of patterns in the training set, d is the number of features for each pattern, k is the desired number of clusters. The training time of both versions is lower than the training time of k-means. The drawbacks of the modified leader algorithm are the same of k-means, and additionally it is order-sensitive.

4. Methodology

The experiments consist in evaluating and comparing the performance of the proposed hybrid system and the correspondent SOM system for organization of Reuters-21578 v1.0 and 20 Newsgroup document collections.

The hybrid system is tested with two different prototype generator methods: k-means and modified leader.

The performances of the systems were measured by means of the quality of document maps (categorization effectiveness) and the training time of the systems (training efficiency). All the systems use the same randomly initialized map for training the document maps. For each hybrid system, the volume reduction step was performed 10 times, generating 10 training sets of prototypes, and the performance of the system was taken as the average over the 10 runs. For the SOM system, the randomly initialized map was trained with the entire training set of document vectors.

We also test the influence of the use of the weighting schema during the SOM map training in the hybrid system performance. The weighting schema consists in weighting each prototype with the number of document vectors in the training set that it represents. Prototypes with major weights are more important than the ones with minor weights in SOM training.

4.1. Document collections and preprocessing

Reuters-21578 v1.0 collection [19] is a benchmark in text categorization literature [6]. It consists of 21,578 news stories that appeared in the Reuters newswire in 1987, which are classified according to 135 thematic categories mostly concerning business and economy. This collection has the following characteristics [6]: (i) each document may belong to none, one, or more than one category, (ii) some categories have very few documents classified under them while others have thousands, and (iii) there are several semantic relations among the categories. We use the subset R90 of this collection and the ModApté split to define the documents used as training and testing examples. Text categorization practitioners have been adopting these subset and partition [6]. The R90 subset contains only documents categorized in at least one of 90 categories (categories with at least one positive training example and one positive test example). After preprocessing, the training set has 7770 document vectors, and the test set has 3019 document vectors.

The 20 Newsgroup collection [17] is also a benchmark in text categorization literature [3]. It consists of approximately 20,000 e-mail messages captured for 20 categories taken from the Usenet newsgroups collection. The e-mail messages mostly deal with computer, business, religion, politics, science, and recreation topics. This collection has the following characteristics: (i) each document may belong to one category, (ii) the categories are balanced i.e., they have almost the same number of documents classified under them, and (iii) there are highly related categories as well as not related categories. We use standard "By Date" split, where documents are ordered by date and the first two-thirds are used for training and the remaining third for testing. After preprocessing, the training set has 11,293 document vectors, and the test set has 7528 document vectors.

The document vectors of the collections are constructed using the vector space model with term frequency. In this process, a standard list of stop words http://www.research.att.com/~lewis is used to remove irrelevant words and remaining words are reduced to base forms using the Porter stemmer algorithm [23].

We reduce the dimensionality of the document vectors by eliminating generic and non-informative terms. The final dimensionalities of document vectors are 5180 and 8165 terms for Reuters-21578 and 20 Newsgroup, respectively.

The document vectors are mounted using tfidf document representation with normalization to unitary length [5].

4.2. Prototype generation

For the k-means and the modified leader algorithms the number of prototypes is 900, equal to the desired number of nodes in the document map as discussed in Section 2. Each algorithm is run 10 times, with randomized sample order.

For modified leader algorithm, a threshold of 0.70 is used and the number of clusters is limited to 900.

4.3. Document map construction

In the experiments, we use a SOM map with rectangular structure and hexagonal neighborhood. The dimensions of the map are 30×30 nodes. The map is randomly initialized using som_randinit function of the somtoolbox [26].

The map randomly initialized was trained with the 10 sets of prototypes generated by each clustering method (10 runs of hybrid system), and with the entire training set (correspondent SOM run).

We perform the SOM training in one stage. For training the SOM map, we use the batch-type SOM algorithm [11] because it is faster to converge than the sequential SOM algorithm. During training, we use truncated Gaussian neighborhood function and neighborhood size linearly decreasing with the number of epochs. The number of epochs of the training step is 10 epochs to the ordering phase and 20 epochs to the fine-tuning phase. The initial neighborhood size is set to half of the number of units in the biggest dimension plus one and the final neighborhood size is set to one in the ordering phase. In fine-tuning phase, the initial and final neighborhood sizes are always equal to one. During fine-tuning phase a convergence condition is also used as a stop criterion (no changes in best matching correspondence between documents and nodes or improvement in mean quantization error below 0.01% between epochs).

4.4. Performance evaluation

After SOM training, each document in training and test set is mapped to the SOM map node with the closest model vector in terms of cosine distance. The nodes are labeled with the category of the document vectors in training set that dominated the node (the category that has the major number of documents in the node) or the category of the best matching document vector for dead neurons. The document vectors of the test set receive the categories of the node where they are mapped.

The classification accuracy for the SOM maps is measured as the percentage of documents mapped to a node labeled with one of its category (correctly classified).

We measure the effectiveness in text categorization for the SOM maps by micro-averaged and macro-averaged F1 [5]. The F1 classifier performance on a category is a combination of precision and recall obtained to the category. When effectiveness is computed for several categories, the results for individual categories must be averaged. In micro-averaged F1 computation the categories count proportionally to the number of their positive examples, while in macro-averaged F1 computation all categories count the same. Micro-averaged F1 is dominated by F1 on common categories while macro-averaged F1 is dominated by F1 on rare categories.

The training time necessary to generate each document map is measured in seconds. The training time for the hybrid systems consists of the time spent in the volume reduction and the construction of the document map steps. For the SOM system, the training time consists of the time spent in the step of the construction of the document map.

The *t*-test of combined variance [27] is used to compare the performances of the system with different clustering methods. The *t*-test is applied on the average and the standard deviation of the performance measures over 10 runs.

5. Results

Tables 1 and 2 show the systems' performance on text categorization of Reuters-21578 and 20 Newsgroup, respectively. The performance of the systems is measured in terms of accuracy, micro-averaged F1, macro-averaging F1, and training time. In this table, KM and ML are short descriptions to k-means and modified leader algorithms, respectively, and "W." means the use of

Table 1
Systems' performance on Reuters-21578 collection

System	Accuracy	Micro-averaged F1	Macro-averaged F1	Training time
SOM	0.8278 ± 0.0000	0.7390 ± 0.0000	0.21581 ± 0.0000	141.0 ± 00.0
W. KM+SOM	0.8021 ± 0.0108	0.7163 ± 0.0097	0.2039 ± 0.0184	115.8 ± 11.1
KM+SOM	0.8057 ± 0.0107	0.7193 ± 0.0096	0.2106 ± 0.0155	127.1 ± 13.0
W. ML+SOM	0.7865 ± 0.0091	0.7022 ± 0.0081	0.1797 ± 0.0145	83.5 ± 9.7
ML+SOM	0.7851 ± 0.0103	0.7009 ± 0.0092	0.1772 ± 0.0132	81.8 ± 8.3

Table 2
Systems' performance on 20 Newsgroup collection

System	Accuracy	Micro-averaged F1	Macro-averaged F1	Training time
SOM	0.6823 ± 0.0000	0.6823 ± 0.0000	0.6738 ± 0.0000	281.0 ± 00.0
W. KM+SOM	0.6657 ± 0.0088	0.6657 ± 0.0088	0.6584 ± 0.0094	202.1 ± 16.5
KM+SOM	0.6735 ± 0.0088	0.6735 ± 0.0088	0.6651 ± 0.0091	217.0 ± 31.1
W. ML+SOM	0.5809 ± 0.0184	0.5809 ± 0.0184	0.5714 ± 0.0183	131.3 ± 13.1
ML+SOM	0.5782 ± 0.0130	0.5782 ± 0.0130	0.5679 ± 0.0134	136.4 ± 06.3

weighting prototypes schema in SOM training. The numbers are, respectively, averages and standard deviations from 10 test runs.

We can see in Table 1 that for the Reuters-21578, the SOM system produces significantly better document maps than the hybrid systems in terms of accuracy and micro-averaged F1; however, the differences in the performance are about 2% and 4% for hybrid system with k-means and modified leader, respectively. Comparing the macro averaging F1 of the generated document maps, there is not significant difference in performance of the hybrid system with k-means and SOM, and the hybrid systems with modified leader algorithm generates document maps with smaller macro averaging F1 than SOM and the hybrid system with k-means (the differences in the performance are about 3%). These facts suggest that the hybrid system have inferior, but close effectiveness to the SOM system.

Analyzing the performance of the hybrid system in Table 1, we can observe that the use of the k-means algorithm as prototype generator produces significantly better document maps than the use of modified leader (better accuracy, micro-averaged F1 and macro-averaged F1). Thus, in terms of effectiveness the hybrid system with the k-means algorithm is better than the hybrid system with the modified leader algorithm. Observing the impact of the use of weighting of the prototypes in the hybrid system effectiveness in Table 1, we conclude that this use does not improve the performance of the hybrid system.

In Table 1, the training time of the hybrid system is significantly smaller than SOM system. The hybrid system with k-means is about 10–18% faster than SOM system. The hybrid system with the modified leader algorithm is about 41–42% faster than the SOM system. Observing the impact of the use of weighting of the prototypes in the hybrid system efficiency in Table 1, we conclude that this use improves the efficiency of hybrid system with k-means, but does not improve the efficiency of the hybrid system with the modified leader.

Table 2 shows the performance of the systems on the 20 Newsgroup document collection. As it is pointed out in [3], the values of accuracy are equal to micro-averaged F1 values because the documents are single-labeled and we perform a single-label categorization task. The SOM system produces significantly better document maps than the hybrid systems in terms of accuracy, micro-averaged F1 and macro-averaged F1; however, the differences in the performance are about 2% and 10% for hybrid system with k-means and modified leader algorithms, respectively. These facts suggest that the hybrid system has inferior, but close effectiveness to the SOM system.

Analyzing the performance of the hybrid system in Table 2, we can observe that the use of the k-means algorithm as prototype generator method yields significantly better document maps than the use of modified leader (better accuracy, micro-averaged F1 and macro-averaged F1). Thus, in terms of effectiveness, the hybrid system with k-means is better than the hybrid system with the modified leader algorithm. Observing the impact of use of weighting of the prototypes in the hybrid system effectiveness in Table 1, we conclude that its use do not improve the performance of the hybrid system.

In Table 2, the training time of the hybrid system is significantly smaller than the SOM system. The hybrid system with k-means is approximately 23–28% faster than the SOM system. The hybrid system with the modified leader algorithm is approximately 51–53% faster than the SOM system. Analyzing the impact of the use of weighting of the prototypes in the hybrid system efficiency in Table 2, we conclude that this use does not improve the efficiency of the hybrid systems.

Based on the observed facts in the categorization of Reuters-21,578 and 20 Newsgroup collection, we conclude that: (i) the hybrid systems generate document maps with performance similar (but inferior) to the SOM system, (ii) k-means is a better prototype generator than the modified leader algorithm when effectiveness is the main goal, (iii) the modified leader algorithm is a better prototype generator than k-means when small training time is preferred, and (iv) prototype weighting schema does not significantly improve the performance and efficiency of the hybrid system.

There are some studies that evaluate SOM on the document categorization of Reuters-21578 or 20 Newsgroup collections. However, only few studies report standard text categorization effectiveness and efficiency measures on well-known subsets of these collections.

Among the studies on document categorization of Reuters-21,578 (for instance [2,7–9,12,13,21]), the unique that can be, at certain level, comparable with ours is [21]. In that study the performance of SOM and hierarchically growing hyperbolic self-organizing map (H^2SOM) are compared. The size of the SOM was

48×48 nodes (2304 nodes), the reported training time was 13 h and 25 min, the micro-averaged F1 was 0.628 and the macro-averaged F1 was 0.633 over the top 20 categories. The long training time is typical of a non-sparse data manipulation implementation of SOM and contrast with the 149 s achieved by our implementation of SOM, besides the difference in the size of the map (we use 30×30 nodes). The micro-averaged and macro-averaged F1 are different mainly because of the set of topics considered (20 of 90 topics) and the different manner of calculating F1 (using retrieval set rather than labeling of nodes). Due to the use of the top 20 of the 90 topics and the use of more nodes, the values of micro- and macro-averaged F1 must be superior to the achieved by the SOM and hybrid systems; however, the micro-averaged F1 reported is inferior to 10% to the achieved by SOM and hybrid systems. Additionally, the effectiveness and efficiency of SOM and hybrid systems are comparable with of H^2SOM (2281 nodes, 829 s of training time, 0.705 of micro-averaged F1 and 0.674 of macro-averaged F1), revealing great quality and efficiency of the SOM and hybrid systems presented.

To the best of our knowledge the unique work using SOM on document categorization of 20 Newsgroup is [9] that it cannot be comparable with ours. However, in [3] it is reported the k-NN accuracy of 0.7593 on this collection, the k-NN effectiveness is about 9% superior to the achieved by SOM and hybrid system with k-means. We conclude that SOM and the hybrid system are very important unsupervised systems, because they have performance close to k-NN. k-NN is one of the best-supervised classifiers to text categorization [24].

We use SOM as the base for our system, because it is computationally the lightest of all its variants [16], decisive aspect in very large document collection organization. However, the proposed hybrid system can be adapted to use SOM variants, with the motivation to reduce the training time of these models e.g., ViSOM [28] and kernel self-organizing map (KSOM) (k-means-based KSOM [4,20], normalized gradient descendent-based KSOM [1,22], and energy function-based KSOM [18]). ViSOM provides a direct visualization of both the structure and distribution of the data by preserving the map the inter-node distances, as well as the topology as faithfully as possible. KSOMs try to improve the classification performance by kernelizing SOM. Both models are more computationally consuming than the SOM. SOM training is about a fraction (approximately one-fortieth) of computation cost of KSOMs [18]. In terms of classification performance, ViSOM is similar to SOM when the number of nodes becomes larger [28] and KSOMs' performances are not superior to the SOM performance [18,29].

6. Conclusions

The proposed hybrid SOM-based document organization system showed to be an effective and fast alternative to the SOM-based document organization system. The hybrid system may be applied to the construction of document maps of large collections, allowing the construction of more intuitive and useful information retrieval systems with less time.

In this article, we characterize how the hybrid system must be constructed to allow fast construction of the document maps: (i) the prototype generator methods must not be more computationally consuming than the k-means, (ii) the upper bound of the number of prototypes generated by the prototype generator method is the number of nodes desired in the document map, and (iii) the k-means and the modified leader algorithms are good alternatives to the prototype generator methods in hybrid systems, the k-means is preferred when effectiveness is the main goal and the modified leader algorithm is preferred when efficiency is more important.

Future works involve: (i) test other clustering algorithms with linear time complexity in the construction of hybrid systems, (ii) determination of the lower bound of the number of prototypes generated to obtain document maps of good quality aiming to minimize the training time of the hybrid system, and (iii) research for methodologies to construct hybrid systems with multiple-stage training of the SOM map.

Acknowledgments

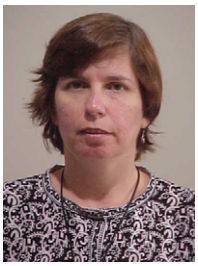
The authors would like to thank CNPq and CAPES (Brazilian research agencies) for their financial support.

References

[1] P. Andras, Kernel–Kohonen networks, Int. J. Neural Systems 12 (2002) 117–135.
[2] A. Azcarraga, T. Yap Jr., SOM-based methodology for building large text archives, Proc. DASFAA (2001) 66–73.
[3] A. Cardoso-Cachopo, A.L. Oliveira, Semi-supervised single-label text categorization using centroid-based classifiers. In: Proceedings of 2007 ACM Symposium on Applied Computing, ACM Press, New York, USA, 2007, pp. 844–851.
[4] E. Corchado, C. Fyfe, Relevance and kernel self-organising maps, in: O. Kaynak, E. Alpaydin, E. Oja, L. Xu (Eds.), Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003, Lecture Notes in Computer Science, vol. 2714, Springer, Berlin, Heidelberg, 2003, pp. 280–287.
[5] R.F. Correa, T.B. Ludermir, A hybrid SOM-based document organization system, in: Proceedings of IXth Brazilian Symposium on Neural Networks, vol. 1, IEEE Computer Society Press, Silver Spring, MD, 2006.
[6] F. Debole, F. Sebastiani, An analysis of the relative hardness of Reuters-21578 subsets, J. Am. Soc. Inf. Sci. Technol. 56 (6) (2005) 584–596.
[7] R.T. Freeman, H. Yin, Adaptive topological tree structure for document organisation and visualisation, Neural Networks 17 (2004) 1255–1271.
[8] A. Georgakis, C. Kotropoulos, A. Xafopoulos, I. Pitas, Marginal median SOM for document organization and retrieval, Neural Networks 17 (2004) 365–377.
[9] J. Ghosh, A. Strehl, Similarity-based text clustering: a comparative study, in: J. Kogan, C. Nicholas, M. Teboulle (Eds.), Grouping Multidimensional Data, Springer, Berlin, Heidelberg, 2006, pp. 73–97.
[10] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, Los Altos, CA, 2000.
[11] J.A. Hartigan, Clustering Algorithms, Wiley, New York, USA, 1975.
[12] J. He, A. Tan, C. Tan, Modified ART 2A growing network capable of generating a fixed number of nodes, IEEE Trans. Neural Networks 15 (3) (2004) 728–737.
[13] M. Hussin, M. Kamel, Document clustering using hierarchical SOMART neural network, in: Proceedings of the International Joint Conference on Neural Networks 2003 (IJCNN 2003), vol. 3, 2003, pp. 2238–2242.
[14] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. 31 (3) (1999) 264–323.
[15] T. Kohonen, Self-organizing Maps, second ed., Springer, Berlin, 1997.
[16] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, A. Saarela, Self organization of a massive document collection, IEEE Trans. Neural Networks 11 (3) (2000) 574–585.
[17] K. Lang, Newsweeder learning to filter netnews, Int. Conf. Mach. Learn. (1995) 331–339.
[18] K.W. Lau, H. Yin, S. Hubbard, Kernel self-organising maps for classification, Neurocomputing 69 (2006) 2033–2040.
[19] D.D. Lewis, Reuters-21578 Text Categorization Test Collection, AT&T Labs Research, 1997. Available: 〈http://www.research.att.com/~lewis〉.
[20] D. MacDonald, C. Fyfe, The kernel self-organising map, in: Proceedings of the Fourth International Conference on Knowledge-based Intelligent Engineering Systems and Allied Technologies, vol. 1, 2000, pp. 317–320.
[21] J. Ontrup, H. Ritter, Large-scale data exploration with the hierarchically growing hyperbolic SOM, Neural Networks 19 (2006) 751–761.
[22] Z.S. Pan, S.C. Chen, D.Q. Zhang, A kernel-base SOM classifier in input space, Acta Electron. Sin. 32 (2004) 227–231 (in Chinese).
[23] M. Porter, An algorithm for suffix stripping, Program 14 (3) (1980) 130–137.
[24] F. Sebastiani, Machine learning in automated text categorization, ACM Comput. Surv. 34 (1) (2002) 1–47.
[25] A. Strehl, J. Ghosh, R. Mooney, Impact of similarity measures on web-page clustering, Proc. AAAI Workshop AI Web Search. (2000) 58–64.
[26] J. Vesanto, Neural network tool for data mining: SOM toolbox, Proc. TOOLMET 2000 (2000) 184–196.
[27] R.R. Wilcox, Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy, Springer, New York, 2001.
[28] H. Yin, Data visualisation and manifold mapping using the ViSOM, Neural Networks 15 (2002) 1005–1016.

[29] H. Yin, On the equivalence between kernel self-organising maps and self-organising mixture density networks, Neural Networks 19 (2006) 780–784.

Renato Fernandes Corrêa received the B.S. degree in Computer Science in 2000 from Federal University of Viçosa, Brazil. He received the M.S. degree in Computer Science in 2002 from Federal University of Pernambuco, Brazil. From 2002 to 2007, he was a Visiting Professor at Polytechnic School at Pernambuco University, Brazil. Currently, he is a Ph.D. student of the Center of Informatics at the Federal University of Pernambuco. His research interests include neural networks and information retrieval systems.



Teresa Bernarda Ludermir received the Ph.D. degree in Artificial Neural Networks in 1990 from Imperial College, University of London, UK. From 1991 to 1992, she was a lecturer at Kings College London. She joined the Center of Informatics at the Federal University of Pernambuco, Brazil, in September 1992, where she is currently a Professor and the head of the Computational Intelligence Group. She has published over 150 articles in scientific journals and conferences, three books in NN and organized two of the Brazilian Symposium on Neural Networks. She is one of the editors-in-Chief of the International Journal of Computation Intelligence and Applications. Her research interests include weightless NN, hybrid neural systems and applications of NNs.