A Hybrid SOM-Based Document Organization System

Renato Fernandes Corrêa 1, 2, Teresa Bernarda Ludermir 2
1 Polytechnic School, Pernambuco University
Rua Benfica, 455, Madalena, Recife - PE, Brazil, 50.750-410
2 Center of Informatics, Federal University of Pernambuco
P.O. Box 7851, Cidade Universitaria, Recife - PE, Brazil, 50.732-970
{rfc, tbl}@cin.ufpe.br

Abstract

This paper presents and evaluates a hybrid system to self-organization of massive document collections based on Self-Organizing Maps. The hybrid system uses prototypes generated by a clustering algorithm to training the document maps, thus reducing the training time of large maps. We test the system with two clustering algorithms: k-means and the AY method. The experiments were carried out with the Reuters-21758 v1.0 collection. The performance of the system was measured in terms of text categorization effectiveness on test set and training time. The experimental results show that proposed system generate pretty good document maps and that the system had similar effectiveness performance with both clustering methods, however the use of k-means generated the smallest training time.

1. Introduction

A SOM-based document organization system [1] can be defined as a system that automatically organize a collection of documents in groups of similarity using self-organizing maps (SOM) [2], generating a document map.

The document map and their graphical representation provide a means to explore large collections of texts by enabling an alternation between visualization, zooming in on interesting information, browsing, and searching for a specific item [1].

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 [1]. The WEBSOM methodology does scale up well even to very large datasets due to the use of a fast dimensionality reduction method called random mapping, use of shortcuts in computation of SOM algorithm and a method to estimate large maps from trained small ones, progressively increasing the size of the SOM archive [1]. 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 [3] that the volume (number of documents vectors) could also stand 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 load 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 do not report a training time analysis and correct evaluation of the quality of the document maps generated by their system.

In this paper, we generalize the methodology proposed by Azcarraga and Yap and propose hybrid SOM-based document organization system architecture. The system proposed by Azcarraga and Yap can be viewed as an instance of the system proposed in this paper.

The objective of this paper is to presents and evaluates a hybrid system to self-organization of massive document collections based on SOM and to report a more deep analyses on how the hybrid system behave with use of different clustering algorithms. The experiments are carried out with the Reuters-21758 v1.0 collection. The performance of the system was
measured in terms of text categorization [4] effectiveness on test set and training time.

2. Hybrid self-organization of document collections

The proposed hybrid SOM-based document organization system performs the following five steps: document indexing, dimensionality reduction, volume reduction, construction of document map and construction of user interface (see Figure 1).

The document indexing step consists on preprocess the text documents and represent them statistically. Generally, non-informative words are removed from initial vocabulary and word affixes are removed using a stemmer algorithm [4]. The isolated words without affixes are called terms. The documents are represented using the vector space model [4], i.e. the documents are represented by vectors, where terms are the indexes and the corresponding values represent the importance of a term to the semantics of a document. The importance of a term in a document is approximated by a function of frequency of occurrence of the term in the document [4].

Dimensionality reduction step receive the document vectors generated in document indexing step and apply some algorithms to reduce the number of dimensions or terms. There are many methods to dimensionality reduction, see [4] 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 vector representing each cluster is taken as prototypes, i.e. representative samples or patterns. The clustering algorithms used to prototype generation must have linear time complexity in the size of the document collection. The clustering algorithms used in the experiments, k-means e AY method, have this property and are described in the next section. Another examples of algorithms with this property are [5]: Leader, CLARANS, and BIRCH.

The prototype vectors are used as input to the step of construction of the document map. This step consists in training a SOM map with the input vectors. The training may be done in one stage, or multiple stages. The training in one stage consists of training a random initialized map with SOM until it reach stationary state. The training with multiple states consists initially in train in one stage a small map and after this do multiple stages of estimation of initial state of a large map based on stationary state of a small one, and fine-tuning the large map to reach 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(n²) [1], where n is the number of training documents vectors, d is the dimensionality of the vectors, and the number of nodes in the large map, M, is assumed to be about one-tenth of n. Assuming that the time complexity of training in one stage is O(ndM), and in multiple stage is O(ndm) + O(dn) +O(n²), where m is the number of nodes in the small map (435 nodes in WEBSOM), we conclude that the multiple stage only is more advantageous than one stage if the collection have more than ten thousands documents.

The last step is the construction of the user interface. The user interface must allow interactive browsing, perform and visualize the results of content-addressable and keyword searches on the document map, details about the interface construction can be viewed in [1].

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 IRS. The map must organize the documents generating consistent clusters. In good quality document maps, similar document vectors must be mapped in the same node or neighboring nodes. Classification error and text categorization effectiveness measures [4] are recommended to evaluate the quality of the document map because they express how the map captures the document similarity.
3. Prototype generation algorithms

We describe in this section the clustering algorithms used to prototype generation in the experiments. We choose the AY method because of the use of a supervised variation of it in [3], and the k-means because is a well known and efficient algorithm. Both prototype generation algorithms are linear time complexity in the size of the training set.

3.1. K-means

We employ a variation of k-means algorithm using cosine similarity measure (cosine of the angle between two vectors). The k-means is an iterative algorithm to minimize a dissimilarity criterion function. The original k-means use Euclidean distance between vectors, thus it minimizes the least squares error criterion [5]. 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 [6], i.e. the cosine measure shows 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, i.e. the mean of all input patterns mapped in 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 most similar center. Subsequent re-computing of the mean for each cluster and re-assigning the cluster labels is 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 clusters.

K-means is the most popular clustering algorithm, the reasons behind the popularity are: it is easy to implement; its linear time complexity in the size of the training set; it is order-independent - 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.

A major problem with this algorithm is that it is sensitive to the selection of the initial partition (sensitive to initial seed selection) and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen. In addition, the k-means algorithm, even in the best case, it can produce only hyperspherical clusters.

3.2. AY method

The AY method is a prototype-based incremental learning method proposed by Azcarraga and Yap in [3]. A supervised version of this method was used to reduce the volume of a subcollection of Reuters-21578. In this work, we use the unsupervised version of this method because it is faster than supervised version and the proposed hybrid system is unsupervised.

Essentially, the method starts off with zero prototypes and adds a prototype whenever none of the existing prototypes is close enough to a current input pattern (random selected input pattern). The newly created prototype is an exact copy of the current input pattern.

The cosine of the angle between the input vector and each prototype is used as similarity measure. An influence threshold, whose value ranges from 0 to 1, is a parameter of the system and determines how similar the best matching prototype should be for it to be considered "close enough".

In cases when some existing prototype is sufficiently close to the current input pattern, this prototype's reference vector (codebook) is adjusted to assimilate the current pattern. In unsupervised mode, the reference vector of the nearest prototype is updated using the so-called "Grossberg learning rule".

Because of the updates made on the reference vectors of winning prototypes, it is possible for some of these vectors to move into locations in the input space where they will never ever win for any of the input patterns. To remedy this so-called "drift phenomenon", a decay mechanism of energy is employed. This mechanism causes the energy of each prototype to decrease by a small amount for every training input pattern. Whenever a prototype wins, its energy is boosted. Once the energy of a prototype goes below a pre-set threshold, the prototype is removed.

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 clusters. The number of epochs of the algorithm T is fixed before the training. The processing time of AY method is higher than k-means.

This algorithm has the same disadvantages of k-means, and additionally is order-dependent.
4. Methodology

The experiments consist in evaluating and comparing the performance of the proposed hybrid system with different prototype generation methods, and the correspondent SOM system (without the volume reduction step), for document organization of a document collection. The performances of the systems were measured by means of the quality and the training time of the generated document maps. 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.

For the hybrid systems, we test also the influence of use or not of weighting of the prototypes in training the SOM map. The weighting schema consists of to give a weight to each prototype equals to the number of document vectors in the training set that it represent. These weights influence the SOM algorithm to give more importance to prototypes with major weights than the ones with minor weights in the topological ordering of nodes.

4.1. Document collection preprocessing

The documents categorized belong to Reuters-21578 v1.0 collection. This collection is a benchmark in text categorization literature; see [7] for more detailed description. This collection 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 [7]: each document may belong to none, one, or more than one category; some categories have very few documents classified under them, while others have thousands of documents; there are several semantic relations among the categories.

The document vectors of the collection were constructed using the vector space model with term frequency. In this process, a standard list of stop words [8] was used to remove irrelevant words and the remaining words were reduced to base forms using the Porter stemmer algorithm [9].

We use the subset R90 of the collection and the ModApte split to define the documents used as training and testing examples. These subset and partition have been adopted by Text Categorization experimenters [7]. 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). The training set has 7770 document vectors, and the test set has 3019 document vectors.

The dimensionality of the document vectors was reduced eliminating generic and non-informative terms (terms occurring in more than half of the training set and less than five documents); the final dimension of the vectors was equal to 5180 terms.

4.2. Prototype generation

For k-means e AY method the number of prototypes generated was 900, equals to the desired number of nodes in the document map.

For AY method the incremental learning procedure was run with an influence threshold of 0.70 and gain parameter in the Grossberg learning rule of 0.1 as suggested in [3]. The number of training epochs was five, to limit the processing time.

4.3. Document map construction

In the experiments, the used SOM map had rectangular structure with hexagonal neighborhood. The dimensions of the map were 30x30 nodes. For each topology, a randomly generated configuration was generated using som_randinit function of somtoolbox [10]. The algorithm used for training SOM networks was the batch-type SOM [6] with truncated Gaussian neighborhood function and neighborhood size linearly decreasing with the number of epochs. The batch-type SOM algorithm was used because it is faster than sequential SOM algorithm.

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

In all the runs the parameters are determined as follows: the number of epochs of the training were 10 epochs to rough phase and 20 epochs to fine-tuning phase; for each topology the initial neighborhood size was equal to the half of the number of units in the biggest dimension plus one and the final neighborhood size was equals to one in rough phase; in fine-tuning phase, the initial and final neighborhood size was always equals to one.

We perform the training of the SOM map in one stage, because the number of document vectors in training set is less than ten thousand and the desired number of nodes in the document map is 900, making this kind of training more advantageous than multiple training stage, as discussed in section 2.
4.4 Performance Evaluation

For each document map generated: each document in training and test set was mapped to the SOM map node with the closest model vector in terms of cosine distance; each map node was labeled according to the category of the document vectors in training set that dominated the node (the category has more than 50% of the documents mapped in the node). The document vectors of the test set received the category assigned to the labeled node where they were mapped.

The classification accuracy for the SOM maps was 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 [4]. 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 averaging way the categories count proportionally to the number of their positive examples; in macro averaging way all categories count the same. Micro averaged F1 tend to be dominated by F1 on common categories, while macro averaged F1 tend to be dominated by F1 on rare categories.

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

The statistical t-test of combined variance [11] was used to compare the performances of the system with different clustering methods, it was applied on the average and the standard deviation of the performance measures achieved by each method in 10 runs.

5. Results

Table 1 shows the systems’ performances measured in terms of accuracy, microaveraged F1, macroaveraging F1 and training time. In this table, KM and AY are short descriptions to K-means and AY method respectively, and “W.” means use of weighting for prototypes obtained from each method.

We can see in Table 1 that SOM system produces significantly better document maps than the hybrid systems in terms of the accuracy and micro averaged F1. 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 AY method generates document maps with significantly better macro averaging F1 than SOM and hybrid systems with k-means. These facts suggest that the hybrid systems have inferior performance than the SOM system in representing the majority classes and superior performance than SOM system in representing the minority classes.

Table 1. Systems’ performance. The figures are averages from ten test runs, and the error margins are standard deviations.

<table>
<thead>
<tr>
<th>System</th>
<th>Accuracy</th>
<th>Micro F1</th>
<th>Macro F1</th>
<th>Trn Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOM</td>
<td>0.7867 ± 0.0000</td>
<td>0.7260 ± 0.0000</td>
<td>0.1656 ± 0.0000</td>
<td>590 ± 00</td>
</tr>
<tr>
<td>W. KM</td>
<td>0.7703 ± 0.0115</td>
<td>0.7148 ± 0.0072</td>
<td>0.1699 ± 0.0056</td>
<td>500 ± 18</td>
</tr>
<tr>
<td>+ SOM</td>
<td>0.7636 ± 0.0131</td>
<td>0.7114 ± 0.0070</td>
<td>0.1707 ± 0.0057</td>
<td>514 ± 27</td>
</tr>
<tr>
<td>KM +</td>
<td>0.0131 ± 0.0115</td>
<td>0.7065 ± 0.0072</td>
<td>0.1925 ± 0.0056</td>
<td>928 ± 39</td>
</tr>
<tr>
<td>W. AY</td>
<td>0.7545 ± 0.0090</td>
<td>0.7025 ± 0.0056</td>
<td>0.1933 ± 0.0097</td>
<td>590 ± 34</td>
</tr>
<tr>
<td>+ SOM</td>
<td>0.7490 ± 0.0131</td>
<td>0.7025 ± 0.0056</td>
<td>0.1933 ± 0.0097</td>
<td>922 ± 34</td>
</tr>
<tr>
<td>SOM</td>
<td>0.0000 ± 0.0000</td>
<td>0.0056 ± 0.0000</td>
<td>0.0097 ± 0.0000</td>
<td>34 ± 00</td>
</tr>
</tbody>
</table>

Comparing the performance of the hybrid systems in Table 1, we can observe that the use of the k-means algorithm as prototype generation generates document maps with significantly better accuracy and micro averaged F1 measure than the ones using AY method. In terms of macro averaged F1, hybrid systems with AY method have better performance than hybrid systems with k-means, although the differences in performance are close to two percent in mean. Thus, the hybrid systems with k-means represent better the majority classes and worst the minority classes than hybrid systems with AY method, but the general performance of the hybrid systems with k-means are better than the hybrid systems with AY method.

Observing the impact of use of weighting of the prototypes in the hybrid systems in Table 1, we conclude that the use of weighting tend to improve the representation of the majority classes and reduce the representation of the minority classes in the generated document maps. There are not significantly differences between the performance of the hybrid systems with the same prototype generation method using or not weighting, but the use of weighting generates significantly differences comparing the hybrid systems with different prototype generation methods. For
instance, the hybrid system with k-means and weighting of the prototypes have significantly better performance than the hybrid system with AY method and weighting of the prototypes, but there is not significantly difference in the performance between the hybrid system with k-means without weighting of the prototypes and hybrid system with AY method and weighting of the prototypes.

Table 1 shows that the training time of the hybrid systems using k-means was significant smaller than SOM system and the hybrid systems using the AY method. Additionally, the hybrid systems with AY method has training time superior to SOM system. Thus k-means, as a prototype generation is the only viable alternative to SOM system. The AY method reduces the training time of the SOM (as observed by Azcarraga and Yap [3]) but it is computational expensive making the construction of the document map by the hybrid system more time expensive than the construction of the document map by a SOM system.

Based on the observed facts we conclude that: the hybrid systems generate document maps with performance similar (but inferior) to a SOM system; k-means is a better prototype generation than AY method; to be a viable alternative to k-means in a hybrid system, a prototype generation can not be more computational expensive than k-means; and that to methods with the same time complexity that k-means, the number of prototypes must be small or equals to the number of nodes in the SOM map to obtain significantly slower training time of a hybrid system than the training time of a analogous SOM system.

6. Conclusion

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 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: the prototypes generations methods (clustering algorithms) must have time complexity lesser or equal to k-means algorithm, and must not be more computational expensive than k-means; the upper bound of the number of prototypes generated by the prototype generation method is the number of nodes desired in the document map.

Future works involves: test other clustering algorithms with small time complexity than k-means in the construction of hybrid system; determination of the lower bound of the number of prototypes generated to obtain document maps of good quality minimizing the training time of the hybrid system; research for methodologies to construct hybrid systems with multiple stage training of the SOM map, useful to organize large document collections.

7. References


