Self-Organising Maps in Document Classification: A Comparison with Six Machine Learning Methods

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

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

Abstract. This paper focuses on the use of self-organising maps, also known as Kohonen maps, for the classification task of text documents. The aim is to effectively and automatically classify documents to separate classes based on their topics. The classification with self-organising map was tested with three data sets and the results were then compared to those of six well known baseline methods: *k*-means clustering, Ward's clustering, *k* nearest neighbour searching, discriminant analysis, Naïve Bayes classifier and classification tree. The self-organising map proved to be yielding the highest accuracies of tested unsupervised methods in classification of the Reuters news collection and the Spanish CLEF 2003 news collection, and comparable accuracies against some of the supervised methods in all three data sets.

Keywords: machine learning, neural networks, self-organising map, document classification.

1 Introduction

Finding relevant information about something is of highest importance, particularly in electronic documents. We need to seek information in our everyday lives, both at home and work while the amount of information available is getting enormous. The Internet is full of digital documents covering almost every topic one can imagine. How can one find relevant information in this massive collection of documents? One cannot really do it manually, so one needs some automatic methods. These methods can help in the search for useful information by clustering, classifying and labelling the documents. When the documents are ordered and preclassified, one can effectively browse through them. This is why we need document classification methods.

The machine learning solutions [21] for the text document classification task mostly use the supervised learning procedure. These include, for example, classic methods such as *k* nearest neighbour searching and Naïve Bayes classifier [8]. However, we are interested in the unsupervised self-organising map method [13], also known as Kohonen map or SOM. It is an artificial neural network originally designed for the clustering of high-dimensional data samples to a low-dimensional map. It is widely used in the clustering and classification of text documents. WEBSOM [12, 15]

A. Dobnikar, U. Lotrič, and B. Šter (Eds.): ICANNGA 2011, Part I, LNCS 6593, pp. 260–269, 2011.

is a self-organising map based method for effective text mining and clustering of massive document collections. However, it is not really designed for the classification of documents. ChandraShekar and Shoba [2] classified 7000 documents with self-organising maps and Chowdhury and Saha [4], Eyassu and Gambäck [9] and Guerro-Bote et al. [11] used smaller collections of a few hundred documents. Recently, Chumwatana et al. [5] used maps in clustering task of news collection of 50 Thai news and Chen et al. [3] compared self-organising maps and *k*-means clustering method in clustering of 420 documents collection. The method has also been used in information retrieval, see for instance [10] and [14].

We have used self-organising maps earlier in information retrieval [18] and document classification [19] tasks of a German news document collection. In classification self-organising map showed some potential by beating *k* nearest neighbour searching and *k*-means clustering in the five (278 documents) and ten class (425 documents) cases with accuracies as high as 88-89%. Encouraged by that performance, we decided to test its classification ability in this research with multiple reasonably large data sets and against well known baseline methods, something that is seldom seen in research papers of this field. We were also interested in testing the self-organising map classification method with documents of different languages. Therefore, one of our present data sets is in Spanish.

The research proved us that self-organising map is an effective method in document classification even when there are thousands of documents in the data set. Self-organising map yielded over 90% micro-averaged accuracy in Data Sets 1 (Reuters, Mod Apte Split collection) and 3 (Spanish CLEF 2003 collection) and competed very well against unsupervised methods and comparably against some of the supervised methods.

2 The Data Sets

2.1 Data Set 1: Reuters-21578, Mod Apte Split

The first data set is a subset of the well known Reuters-21578 collection [17, 21]. The complete collection includes 21578 English Reuters news documents from the year 1987. We chose the widely used Mod Apte split [1, 21] subset, which contains 10789 documents and 90 classes. Some of these documents have multiple class labels. To make things simpler, we discarded those and took only the documents with one label. Then, we selected 10 largest classes and finally obtained our collection of 8008 documents, consisting of 5754 training samples and 2254 test samples. The class labels are words, for example 'earn', 'coffee' and 'ship'.

2.2 Data Set 2: 20 Newsgroups, Matlab/Octave

The second data set is also a widely used collection of 20 Internet newsgroups [21, 23]. We selected its Matlab/Octave version, which provides 18774 English documents, 12690 in the training set and 7505 in the test set. The class labels are names of newsgroups, for instance 'rec.sport.hockey', 'soc.religion.christian' and 'sci.space', and each document has only one class label.

2.3 Data Set 3: Spanish CLEF 2003

The third data set is the Spanish collection of CLEF 2003 news documents [6]. The collection contains news articles from the years 1994 and 1995. There are 454045 documents in the complete collection. Here test topics form the classes, and the relevant documents for each topic the class members. From the 60 available classes we selected 20 largest classes. There were, in all, 1901 documents for the top-20 classes. Finally, we constructed 10 test sets using a 10-fold cross-validation procedure. In this data set each document has only one class label. The labels are news topics, such as 'Epidemia de ébola en Zaire', 'Los Juegos Olímpicos y la paz', 'El Shoemaker-Levy y Júpiter'.

3 Preprocessing

Conventional preprocessing was performed to all three data sets. Firstly, the SNOWBALL stemmer was used to transform words to their stems, for instance word 'continued' became 'continu'. Then, stopwords, useless "little" words, such as 'a', 'about' and 'are', were removed. At this point also words shorter than three letters, numbers and special characters were discarded. This is because short words generally have little information value for the classification of documents.

Next, we calculated the frequencies of remaining words (word stems). Then we computed document vectors for all documents by applying the common vector space model [20] with *tf·idf* weighting for all remaining word stems. Thus, a document was 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, $1 \leq i \leq n$, $1 \leq k \leq$, t, where n is the number of documents and t is the number of 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 to the number of the occurrences of word k in document D_i and l is for all words of D_i, l=1,2,..., 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 the number of the documents in the set and n_k is the number of documents, which contain word k at least once. Combining equations (2) and (3) we obtain a weight for word k in document D_i

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

After this procedure every document has its own document vector. Finally, the length of each document vector was shortened only to 1000 of middle frequency (around median) word stems from the total word frequency distribution sorted in ascending order. 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. Only 1000 stems were chosen to ease the computational burden and based on the fact that it had proven quite effective choice in a previous study [19]. The same vectors were then used for all of the methods used, except for the Naïve Bayes method, which needed frequency weighted vectors.

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

4 Document Classification with Self-Organising Map

In order to use a self-organising map in the document classification task we needed to label the map nodes with class labels of the training data set in some meaningful way. The labelled nodes then represent the document classes and the map is able to classify new documents (test set samples) by mapping them. The following simple procedure was implemented to label the self-organising map with class labels:

- Create a self-organising map using a training data set.
- Map each training set sample to the map.
- Determine a class for each node of the map according to the numbers of training documents of different classes mapped on that 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 of the node.

After this procedure the map is labelled with class labels. Fig. 1 shows an example of a labelled map. The data on the map is the training set of Data Set 1 and the labels are: #1 earn, #2 acq, #3 crude, #4 trade, #5 money-fx, #6 interest, #7 money-supply, #8 ship, #9 sugar and #10 coffee. Most of the classes seem to form one or two clusters on the map.

After giving the labels to the map nodes, the classification of the test set was done by mapping each test sample and comparing the classification result given by the map with the known class label of the sample.

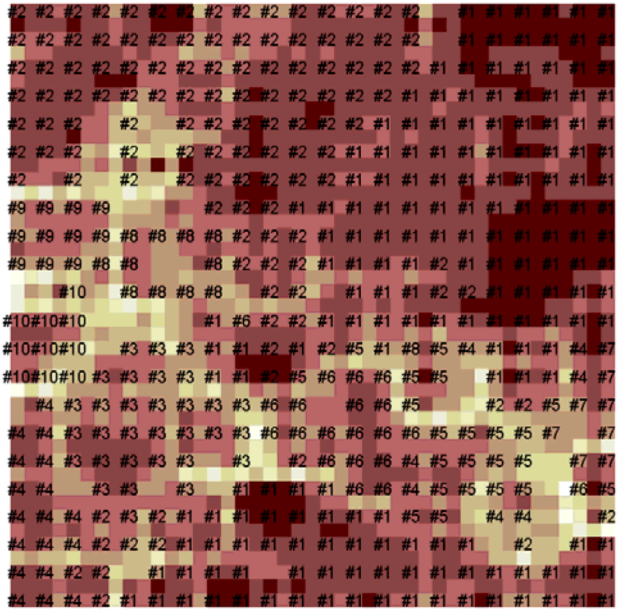

Fig. 1. Labelled self-organising map. The numbers on the map surface are class labels. The darker the colour on the map, the closer the neighbouring nodes are to each other.

The self-organising map was implemented with the SOM_PAK [22] program written in C in Helsinki University of Technology, Finland. We programmed supporting software tools in Java.

5 The Baseline Methods

To evaluate the classification performance of self-organising maps, six classic baseline methods were used in comparison. The idea was to take some unsupervised methods as well as some supervised. Being unsupervised a self-organising map is in disadvantage against the supervised methods as it does not use the class information of the training samples at all. On the other hand, unsupervised methods can be used even without the class labels. In real life the labels are rarely available.

The selected unsupervised methods were ***k*-means clustering** and **Ward's clustering**. For these methods a similar labelling procedure as described earlier for self-organising map had to be implemented for the classification task. The chosen supervised methods were ***k* nearest neighbour searching**, **discriminant analysis**, **naïve Bayes classifier** and **classification tree**. All these baseline methods were implemented with Matlab software.

More information about these baseline methods can be found from numerous sources, for example in [8].

6 Results

In the classification of Data Sets 1 and 2 we used a single test set, while with Data Set 3 we used the 10-fold cross-validation procedure. Because of the randomness in the self-organising map method initialization phase, we built 10 maps for each test set and calculated the average outcome. The results of the baseline methods were calculated with the same test sets, but they were run once for each set, because there was no randomness in these methods. The same preprocessed document vector data was used for all methods, except for the Bayes method, which needed frequency weighted data. No information of the test set vocabulary was used in the word selection of the vectorization.

Two measures of classification performance were used: micro- and macro-averaged accuracy [21]. Micro-averaged accuracy for a given test set *j* is

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

where c_j 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. Macro-averaged accuracy for test set *j* is computed with

$$a_j^{macro} = \frac{\sum_{k=1}^{nc_j} d_{jk}}{nc_j} \tag{6}$$

where nc_j is the number classes in the test set *j* and the d_{jk} (k=1,...,nc_j) is of form

$$d_{jk} = \frac{c_{jk}}{n_{jk}} 100\%, \tag{7}$$

where c_{jk} is the number of correctly classified documents in class *k* of test set *j* and n_{jk} is the number of documents in class *k* of test set *j*. The micro-averaged accuracy tells how well all the documents were classified, but it does not take the class differences into account, and is, therefore, very much influenced by the largest classes, when class sizes are imbalanced. The macro-averaged measure addresses the importance of all classes and it lessens the influence of large classes.

The preprocessed vectors of 1000 features were used and the free parameters of the methods were tested for optimal results in each data set. The free parameters were the map size for self-organising map, the number of nearest neighbours searched for *k* nearest neighbour method, the number of clusters for *k*-means clustering and the maximum number of clusters for Ward's clustering. Based on the obtained accuracies, the best result for every method was selected to be compared

Table 1 shows the results of the Data Set 1, the Reuters news collection (Mod Apte Split). Overall the results are good with most of the methods performing over 90% accuracies (micro-averaged). Naïve Bayes, discriminant analysis and self-organising map yielded the top results. Self-organising map was the best unsupervised method.

Table 2 shows the results of Data Set 2, which was more difficult to classify than the other two data sets. Only the top two methods (Bayes and discriminant analysis) gave over 50% of correct answers (micro-averaged). Self-organising map was the best of the unsupervised methods.

Table 1. Micro- and macro-averaged classification accuracies (%) and the significant differences (Friedman test) of the methods for the Data Set 1. Significant statistical differences are here notated with '>' and '<' characters. For example, A > {B, C} means that A is significantly better than B and A is significantly better than C

	Accuracy (%)		Significant
Method	**micro**	**macro**	**differences (macro)**
Self-organising map (som)	92.3	83.5	>war , <nba
k-means clustering (kme)	90.7	79.2	>war , <{nba, dis}
Ward's clustering (war)	81.1	55.8	< {knn, dis, clt, nba, kme, som}
k nearest neighbour search (knn)	83.0	76.7	>war , < {dis, nba}
Discriminant analysis (dis)	95.0	87.0	> {knn, kme, war}
Naive Bayes (nba)	95.2	90.4	> {som, kme, war, knn, clt}
Classification tree (clt)	91.1	81.7	> war , < nba

Table 2. Micro- and macro-averaged classification accuracies (%) and the significant differences (Friedman test) of the methods for the Data Set 2. Significant statistical differences are here notated with '>' and '<' characters.

	Accuracy (%)		Significant
Method	**micro**	**macro**	**differences (macro)**
Self-organising map (som)	42.3	41.4	>kme , <{dis, nba}
k-means clustering (kme)	30.6	29.9	<{som, war, dis, clt, nba}
Ward's clustering (war)	41.9	40.8	>{kme, knn} <{clt, nba}
k nearest neighbour search (knn)	38.9	38.6	<{war, dis, nba, clt}
Discriminant analysis (dis)	60.1	59.4	>{som, kme, war, knn, clt}
Naive Bayes (nba)	62.0	61.1	>{som, kme, war, knn, clt}
Classification tree (clt)	46.2	45.5	>{kme, knn} , <{dis, nba}

Table 3. Micro- and macro-averaged classification accuracies (%) and the significant differences (Friedman test) of the methods for the Data Set 3. Significant statistical differences are here notated with '>' and '<' characters.

	Accuracy (%)		Significant
Method	**micro**	**macro**	**differences (micro)**
Self-organising map (som)	95.6	91.7	>kme , <{war, knn, dis, nba}
k-means clustering (kme)	90.8	86.7	<{war, knn, dis, clt, nba, som}
Ward's clustering (war)	97.2	96.0	>{dis, clt, kme, som} , <nba
k nearest neighbour search (knn)	97.0	95.6	>{som, kme, dis, clt} , <nba
Discriminant analysis (dis)	96.3	94.9	>{som, kme, clt} , <{war, knn, nba}
Naive Bayes (nba)	98.1	97.5	>{som, kme, war, knn, dis, clt}
Classification tree (clt)	94.5	92.3	>{kme} , <{war, knn, dis, nba}

The results of Data Set 3, the Spanish CLEF news collection, are in Table 3. It turned out to be the easiest case of the three data sets. All methods gave over 90% accuracies (micro-averaged) and Naïve Bayes outperformed the others with very good 98.1% result. Self-organising map performed well with 95.6% and were the second best of the unsupervised methods. In this set also the macro-averaged accuracies were also reasonably high.

Naïve Bayes proved to be the most effective in all cases and discriminant analysis performed almost at the same level. Self-organising maps were at least average compared to others in all cases, and among the unsupervised methods it was the most consistent. Another interesting outcome was that in *k* nearest neighbour classification the best results was always, with all three data sets, obtained with k=1, although we tried with *k* values up to 20.

We conducted the Friedman test [7] to compare the results. For the Data Set 3 we used the micro-averaged accuracies of 10 test sets. For Data Sets 1 and 2 we had to use the macro-averaged accuracies to get enough data, because there was only one test set in these data sets. All the significant differences ($p < 0.05$) between methods are shown in the Tables 1-3. For example, self-organising map was significantly better than *k*-means clustering in Data Sets 2 and 3, and also significantly better than Ward's clustering in Data Set 1. On the other hand, Naïve Bayes was significantly better than self-organising map in all three data sets.

7 Conclusions and Discussion

We tested self-organising map in text document classification task with three different kinds of collections and compared the results to those of the standard text classification methods. Naïve Bayes turned out to be the most effective of all, but self-organising map performed well in its own category of the unsupervised methods. Overall, the results of self-organising maps were encouraging with over 90% classification accuracy (micro-averaged) in Data Sets 1 and 3. This suggests that it is an effective method for the document classification tasks. Futhermore, self-organising maps performed comparably against some of the supervised classification methods tested. The intuitive visual map (see Fig. 1) and the unsupervised learning phase are also a benefit of using self-organising map in document classification, because the map enables data visualization, and in some applications browsing. Additionally, labelled data is rarely available.

Even the costly learning procedure has its benefits. If we compare self-organising map with *k* nearest neighbour, it is easy to see that the learning phase of the map takes much more time than the learning of *k* nearest neighbour. In actual classification it is quite the opposite. The map has usually by an order of magnitude less nodes than there are documents in the training set and this actually leads to 10 times faster classification compared to *k* nearest neighbour with the same data. One does not have to construct a new map every time when a new document is added to the collection, one can just map it and do the learning later when there is more new data available. The slow learning is done rarely and the fast classification often.

The self-organising map method could give even better results, if some more advanced features would be implemented. For example, it is possible to calculate classification based on multiple class hits on the map, for instance using three nearest could be classes. Another approach is the use of multiple maps. In the future, we consider these options and focus on the dimensionality reduction and feature selection problem associated with document vectors.

Acknowledgements

Jyri Saarikoski was supported by the Tampere Graduate Programme in Information Science and Engineering (TISE). SNOWBALL stemmer was by Martin Porter.

References

1. Apte, C., Damerau, F.J., Weiss, S.M.: Automated learning of decision rules for text categorization. ACM Transactions on Information Systems 12, 233–251 (1994)
2. ChandraShekar, B.H., Shobha, G.: Classification of Documents Using Kohonen's Self-Organizing Map. International Journal of Computer Theory and Engineering 5(1), 610–613 (2009)
3. Chen, Y., Qin, B., Liu, T., Liu, Y., Li, S.: The Comparison of SOM and K-means for Text Clustering. Computer and Information Science 2(3), 268–274 (2010)
4. Chowdhury, N., Saha, D.: Unsupervised text classification using kohonen's self organizing network. In: Gelbukh, A. (ed.) CICLing 2005. LNCS, vol. 3406, pp. 715–718. Springer, Heidelberg (2005)
5. Chumwatana, T., Wong, K., Xie, H.: A SOM-Based Document Clustering Using Frequent Max Substring for Non-Segmented Texts. Journal of Intelligent Learning Systems & Applications 2, 117–125 (2010)
6. CLEF: The Cross-Language Evaluation Forum, `http://www.clef-campaign.org/`
7. Conover, W.J.: Practical Nonparametric Statistics. John Wiley & Sons, New York (1999)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, New York (2001)
9. Eyassu, S., Gambäck, B.: Classifying Amharic News Text Using Self-Organizing Maps. Proceeding of the ACL Workshop on Computational Approaches to Semitic Languages, Ann Arbor, Michigan, USA, pp. 71–78 (2005)
10. Fernández, J., Mones, R., Díaz, I., Ranilla, J., Combarro, E.F.: Experiments with Self Organizing Maps in CLEF 2003. In: Peters, C., Gonzalo, J., Braschler, M., Kluck, M. (eds.) CLEF 2003. LNCS, vol. 3237, pp. 358–366. Springer, Heidelberg (2004)
11. Guerro-Bote, V.P., Moya-Anegón, F., Herrero-Solana, V.: Document organization using Kohonen's algorithm. Information Processing and Management 38, 79–89 (2002)
12. Honkela, T.: Self-Organizing Maps in Natural Language Processing, Academic Dissertation. Helsinki University of Technology, Finland (1997)
13. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (1995)
14. Lagus, K.: Text retrieval using self-organized document maps. Neural Processing Letters 15, 21–29 (2002)
15. Lagus, K., Kaski, S., Kohonen, T.: Mining massive document collections by the WEBSOM method. Information Sciences 163(1-3), 135–156 (2004)

16. Moya-Anegón, F., Herrero-Solana, V., Jiménez-Contreras, E.: 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 (2006)
17. Reuters-21578 collection,
    `http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`
18. Saarikoski, J., Laurikkala, J., Järvelin, K., Juhola, M.: A study of the use of self-organising maps in information retrieval. Journal of Documentation 65(2), 304–322 (2009)
19. Saarikoski, J., Järvelin, K., Laurikkala, J., Juhola, M.: On Document Classification with Self-Organising Maps. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 140–149. Springer, Heidelberg (2009)
20. Salton, G.: Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer. Addison-Wesley, Reading (1989)
21. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
22. SOM_PAK,
    `http://www.cis.hut.fi/research/som-research/nnrc-programs.shtml`
23. 20 newsgroups collection,
    `http://people.csail.mit.edu/jrennie/20Newsgroups/`