KOHONEN NETWORKS WITH GRAPH-BASED AUGMENTED METRICS

Peter Andras and Olusola Idowu
School of Computing Science
University of Newcastle
Newcastle upon Tyne, NE1 7RU, UK
{peter.andras,o.c.idowu}@ncl.ac.uk

Abstract – *Correct and efficient text classification is a major challenge in today's world of rapidly increasing amount of accessible electronic text data. Kohonen networks have been applied to document classification with comparable success to other document clustering methods. An important challenge is to devise text similarity metrics that can improve the performance of text classification Kohonen networks by integrating more semantic information into the metric. Here we propose an augmented metric for text similarity that is based on the comparison of word consecutiveness graphs of documents. We show that using the proposed augmented similarity metric Kohonen networks perform better than Kohonen networks using usual Euclidean distance metric comparison of word frequency vectors. Our results indicate that word consecutiveness graph comparison includes more semantic information into the text similarity measure improving text classification performance.*

Key words – augmented metric, Kohonen network, text classification, word consecutiveness graph

1 Introduction

Analysing structured and semi-structured data is a major challenge today in the world of fast increasing volume of accessible electronic data [1]. Text data is perhaps the most common type of such data. Text mining, text search and retrieval, text classification and other related tasks constitute key functional components of many popular Internet services (e.g., search engines) and other computer-based applications (e.g., electronic library catalogues). At the foundation of these tasks resides the similarity comparison of text data.

Classification of text data can be done in various ways, commonly used methods include: bisecting k-means clustering [2], principal component analysis [3], independent component analysis [4], classification with probabilistic mixture models [5], and self-organizing maps [6-8]. Self-organizing maps, and in particular Kohonen networks are one of these methods, which can be used to classify text data efficiently, also supporting the relatively straightforward visualisation of classification results [6-8].

Text similarity is measured usually by comparing the word frequency vectors of two texts [2]. Word frequency vectors contain the list of words occurring within the text together with the relative frequency of these words within the text. Comparison of these vectors is done in a metric space having a separate dimension for each possible word. This approach considers texts as bags of words. Recent works on word consecutiveness graphs show that such graphs contain important information about the structure of the language and contribute to the determination of meaning [9, 10]. This suggests that considering the word consecutiveness graphs for the calculation of similarity between texts should improve the similarity metric used for comparison of text data.

We construct an augmented text similarity metric in this paper, considering the word consecutiveness graphs of texts. We build and analyse Kohonen networks using the augmented text similarity metric. Our analysis shows that in accordance with our expectation Kohonen networks with the augmented text similarity metric have better performance than Kohonen networks working with usual similarity metric calculated by comparing word frequency vectors only.

The rest of the paper is structured as follows. In Section 2 we give a brief introduction to Kohonen networks. In Section 3 we provide a brief overview of text similarity metrics and text classification. In Section 4 we introduce our augmented text similarity metric. Section 5 contains the description of the Kohonen networks using the augmented metric, the text classification experiments, and the results and their interpretation. The paper is closed by Section 6 containing conclusions.

2 Kohonen networks

The Kohonen networks were introduced in the early 80s by Kohonen [11] to model how the brain processes sensory stimuli. The key idea of Kohonen networks is that multi-dimensional data vectors can be represented by a few-dimensional feature vectors that contain characteristic features of the original data vectors. The Kohonen networks perform a topology-preserving dimension reduction in order to transform the high-dimensional original vectors into low-dimensional feature vectors.

Let us consider data vectors $x^t \in R^m, t = 1, \dots, n$. A Kohonen network is a set of neurons, $N_i, i = 1, \dots, p$, each of them containing a data prototype vector $w^i \in R^m$ and a feature vector $v^i \in R^q$ (i.e., v^i represents a position in the q-dimensional feature space). The neurons of the Kohonen network are initialized with random w^i and v^i vectors. By training the Kohonen network the w^i vectors are changed such that if $x, y \in R^m$ are two data vectors, and $i(x) = \arg\min_i \| x - w^i \|, i(y) = \arg\min_i \| y - w^i \|$, and $v^{i(x)}$ and $v^{i(y)}$ are close to each other in the feature space then x and y are close to each other in the data space. In other words topological neighbourhoods of the data space are preserved in the feature space. The invariant characteristic of data vectors represented by a given neuron of the Kohonen network having the feature vector v^i, is that these data vectors belong to the Voronoi cell of the data prototype vector w^i, where the Voronoi cell of the data prototype vector is determined by the Voronoi tesselation of the data space induced by the data prototype vectors of the Kohonen network [11].

Kohonen networks are trained by repeated presentation of a training data set. For each presentation of a data vector x, a neuron N_j of the network is selected in a winner-take-all manner, such that

$$j = i(x) = \arg\min_i \| x - w^i \| \tag{1}$$

The data prototype vectors of neighbouring neurons are updated according to the following equations:

$$w^k = w^k + c \cdot (x - w^k) \tag{2}$$

$$k \in Nb(j) = \{h \mid \| v^h - v^j \| < \rho\} \quad (3)$$

The parameters c, ρ change during the training such that they gradually approach zero (e.g., the update equations $c = \alpha \cdot c,\ \rho = \beta \cdot \rho,\ 0 < \alpha, \beta < 1$ are applied after each epoch of presentation of the full training data set).

After training the Kohonen network performs a topology-preserving transformation of the original data vectors into lower dimensional feature vectors, i.e., a data vector x is transformed into the feature vector $v^{i(x)}$). The Kohonen networks learn the distribution of the data in the data space. If the data can be classified in the data space on the basis of the space-specific similarity metric (e.g., norm of the difference vector), this classification will be preserved in the feature space [6, 12-14].

To use the trained Kohonen network for classification we need to label the neurons with class labels. To attach labels to Kohonen neurons we present the training data vectors to the trained network. We count for each neuron how many data vectors belonging to each class are attracted to this neuron (i.e., a vector x is attracted by neuron N_j if $j = i(x)$). The class label of the dominant class will be attached to the neuron (i.e., if the majority of the attracted data vectors for neuron N_j belong to class C than the class label C will be attached to the neuron N_j).

Other related methods used for classification are the learning vector quantization (LVQ) methods [11], in which case the data prototype vector update is restricted to the winner neurons, i.e., we do not update the weight vectors of those neurons that satisfy the neighbourhood criterion (see equation (3)). A significant difference between LVQ methods and Kohonen networks is that in the case of the former, class labels are fixed and the data prototype vectors are updated by taking into consideration their associated class labels (i.e., the vectors with correct class labels are moved towards the incoming data vectors, and those with different class labels are moved away from the incoming data vectors). The advantage of using Kohonen networks instead of LVQ methods for classification is that the neighbourhood-based adaptation can take into account the noise present in the data, leading to better classification performance in presence of noise, which is the usual case of real data [11].

3 Text similarity and classification

The amount of electronically accessible text data is growing very rapidly. The exchange of information in an increasing number of areas is based today on electronic text data (e.g., communications with government institutes, scientific publications). Consequently the need is growing for appropriate and efficient text processing applications.

Text data contains sequences of words organized in an order compatible with the syntax and semantics of the language and context in which the text was produced. Usual classification analysis of text data ignores a large part of the information contained in the text, typically considering the text as collection of words, ignoring their particular order in the text [3]. An important problem in increasing the information content of text data representation is that it is difficult to extract structured content related information from raw text data (e.g., grammatical structure) [15].

Text classification is based on measuring the similarity between text data. In principle, high text similarity implies the similarity of the content and meaning of text data. In practice, text similarity is measured by measuring the similarity of text representations [16], which necessarily distorts the similarity measure. Consequently, text classification based on text similarity analysis deals with highly noisy data, which increases the difficulty of solving the classification problem efficiently.

Similarity measures used in the context of text classification are usually based on the word frequency vector associated with text data. For a given text data the different words are considered and their frequencies are measured within the text. To reduce the noise in the vector representation of the text data, the words are usually filtered against the list of most common words (i.e., these would not help the discrimination between texts), and the remaining words are stemmed, i.e., reduced to their stem word using some stemming algorithm (e.g., Porter's stemming algorithm [17]). After such pre-processing the absolute frequencies of remaining words are counted and the relative frequencies are calculated, the text data being represented by the vector of relative frequencies of words (stemmed and filtered) occurring in the text.

The simplest way to calculate the measure of similarity between two text documents is to calculate the Euclidean distance between frequency vectors representing two documents [16], i.e., if f^1, f^2 are two frequency vectors, the similarity between the documents represented by these frequency vectors is given by $d = \| f^1 - f^2 \|$, if d is small (e.g., close to zero) the two documents are considered similar, while if the value of d is large the documents are considered dissimilar. To eliminate the problem caused by uncommon words, both frequency vectors are padded with zeros for all words which appear only in the other document. To improve the discriminative power of the frequency vector comparison, the components of the vectors can be weighted such that words having more discriminative power contribute more to the distance between frequency vectors. The common method for such weighting is to use the inverse document frequency (IDF) weights [16]. The IDF weight of words is calculated in the context of document collection, within the limits of which we aim to measure the similarity between documents. The IDF weight is calculated as

$$IDF(s) = \ln\left(\frac{M_{total}}{M(s)}\right) \tag{4}$$

where M_{total} is the total number of documents in the collection and $M(s)$ is the number of documents containing the word s. Words that are very common will have a low weight, while words, which are rare, will have high weight. An alternative commonly used way to calculate similarity between word frequency vectors is to use the cosine measure of vectors [18]. If f^1, f^2 are two frequency vectors, the cosine measure of the two vectors is one minus the value of the cosine of the angle between the two vectors:

$$d_{\cos}(f^1, f^2) = 1 - \cos(f^1, f^2) = 1 - \frac{< f^1, f^2 >}{\| f^1 \| \cdot \| f^2 \|} \tag{5}$$

The cosine measure is large (close to 1) if the vectors are close to orthogonal (i.e., there are very few common words between the two documents), and it is small (close to 0) if the vectors are very similar.

Classification of text documents can be done using various methods on the basis of a chosen similarity metric calculated by considering (possibly weighted) word frequency vectors. The most commonly used methods are the bisecting k-means clustering [2] and the variants of the latent semantic analysis method [4]. More recently proposed clustering methods include the use support vector machines to define document clusters [16], probabilistic mixture models [5], and self-organising maps applied to document classification [6-8].

4 Augmented metric for text similarity

Recent research results indicate an increasing interest in finding better metrics for measuring document similarity, aiming to include more semantic information in the measure [6, 15, 19]. Other recent works have shown that the word consecutiveness networks of languages are likely to be

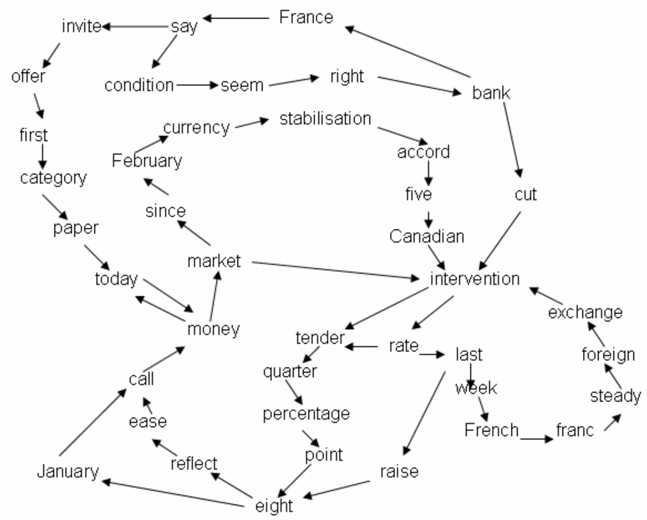

Figure 1. The word consecutiveness graph of a short document from the Reuters-21578 text database

related to the syntactic and semantic structure of the language [9]. Other results show that word semantics can be derived by analysing word co-occurrences in documents [10].

Inspired by the above mentioned results we propose to augment the word frequency vector based measure of text similarity by considering the similarity of word consecutiveness graphs of compared documents. The word consecutiveness graph is constructed by having as nodes the words in the document, and for each pair of consecutive words in the document we add to the graph the directed arc linking the nodes corresponding to these words (see Figure 1 for an example). Intuitively, adding more information in terms of the word consecutiveness graph should lead to better results in terms of correctness of document clustering. In our view the word consecutiveness graph of the document contains significant semantic information about the document, although in a relatively simplified form. In case of very short documents that contain a single sentence of unique words this is obvious, as the whole document can be reproduced using the word consecutiveness graph. In case of long documents the words are repeated many times, and the word graph will represent an extract of the syntactic / semantic structure of the document.

We define our augmented text similarity measure for two documents represented by the pairs of their word frequency vectors and word consecutiveness graphs, $D_1(f^1, G_1), D_2(f^2, G_2)$ as follows:

$$d(D_1, D_2) = \frac{1}{\gamma(G_1, G_2)} \cdot \| f^1 - f^2 \| \tag{6}$$

where

$$\gamma(G_1, G_2) = | E(G_1 \cap G_2) | \tag{7}$$

i.e., $\gamma(G_1, G_2)$ is the number of common edges between the two word consecutiveness graphs. In this way we apply a weighting to the similarity measure calculated as the Euclidean distance of the two word frequency vectors, the weighting representing the semantic similarity between the two documents. If two documents share a large number of words and also have a large common part between their word consecutiveness graphs the measure will be very close to zero, indicating high similarity between the documents. If only the two word frequency vectors are similar to the same extent as in the previous case, but the words are used many times in different ways, the augmented measure will produce a larger value for the documents, indicating less similarity between them. We expect that the augmented metric will include enough semantic information such that the clustering results of the network with augmented metric will be better than the clustering results obtained by networks using Euclidean distance metric defined for word frequency vectors to measure text similarity.

We note that our choice of augmenting the vector distance similarity metric with a similarity measure of word consecutiveness graphs is perhaps the simplest one. We expect that this simple version of augmented measure will work well for short documents (e.g., news items contained in the Reuters-21578 text database) and it should also work reasonably well for longer documents. In case of longer documents more elaborated measures may also be implemented to measure the similarity of word consecutiveness graphs with more precision. For example, the frequency of links (i.e., the frequency of pairs of consecutive words) can be taken into account to get a better measure of similarity of word graphs. However, we note than in case of short documents, like news items, the frequency of links is generally low, and taking into account such frequencies in measuring graph similarity does not have significant effect on the similarity measure. Another possibility is to compare the motif (i.e., highly connected small sub-graphs, like small size cliques) distribution of large word graphs [20], which has been suggested to give a good measure of functional similarity of the system represented by the network. Again, in case of small documents the word graphs contain no or at most very few highly connected clusters, so the motif-based measure would not lead to meaningful addition to the simple graph similarity measure that we introduced above.

5 Results

To evaluate the performance of Kohonen networks with the above proposed augmented text similarity measure we considered the Reuters-21578 text database containing short news items [21]. Many news items contained in the database are indexed with topic markers, defining the class of the item. A large part of the indexed items have more than one topic term, associating them with more than one document class. The news items are also indexed as training or test data, according to their use as such data in earlier experiments [22].

We used two types of Kohonen networks, one with Euclidean distance metric and another with graph-based augmented metric. Both types of networks were trained with the whole training dataset of the Reuters-21578 text database. In both cases each network had 1500 neurons having 3-dimensional feature vectors set by uniform random distribution over the unit cube. To initialize the data prototype vectors of the neurons we analyzed first the whole database to determine the list of all existing words (23501 distinct words in total, including various abbreviations and misspellings) together with their frequencies in the whole database. The prototype vectors were set by random sampling the calculated word distribution.

To compare the performance of the two types of networks we tested the networks using the whole database. The training data were used to set the training class labels associated with each neuron and the test data was used to calculate the validating set of class labels for each neuron. The class labels of attracted training data items were summed, giving a list of training class labels with frequencies of class labels. In the same way the test data was used to generate the validation class label list, again with frequencies of class labels (e.g., the class label set with absolute frequencies of a neuron was: ‘sugar – 2; interest – 5; money-fx – 1; acq – 1; grain – 1; wheat – 1; corn – 1’, while the validation class label set with absolute frequencies was: ‘grain – 2; wheat - 2’). We calculated for each neuron the match between the training and validation class label sets by calculating the Tchebishev distance between the two label frequency vectors (possibly padded with zeros for labels which were present in only one of the two label sets), i.e., if φ^1, φ^2 were the two frequency

Table 1. Comparison of class label frequency vector mismatches between Kohonen networks with augmented and Euclidean metrics

Network type	Average	Variance
Kohonen w. Euclidean metric	1.060355	0.577117
Kohonen w. augmented metric	0.981618	0.595463
t-value / significance	4.057931	<0.001

vectors, the distance was calculated as $\delta(\varphi^1, \varphi^2) = \sum_{k=1}^{r} | \varphi_k^1 - \varphi_k^1 |$, where r is the number labels in the union of the two label sets. The maximal distance is 2, i.e., when the labels are completely different, taking into account that $\sum_{k=1}^{r} \varphi_k = 1$. The performances of the two types of networks were compared in terms of average mismatch between training and validation class label frequency vectors of neurons.

In total we trained five networks with augmented metric and seven networks with Euclidean metric. In the case of networks with augmented metric we found 2179 neurons (435.8 neurons in average) that attracted at least one data item, while in the case of networks with Euclidean metric we found 5339 such neurons (762.7 neurons in average). We calculated for both network types the average mismatch between the training and validation class label frequency vectors of all neurons that attracted at least one data item, together with the variance of mismatch values. The results are presented in Table 1.

The comparison results show that Kohonen networks with augmented metric achieved significantly better performance in terms of mismatch between label frequency vectors than Kohonen networks with Euclidean metric (i.e., the t-value of the difference between the average mismatches is significant at a level p<0.001). This confirms our expectation that in the case of short text data the augmented similarity measure introduced in this paper increases the correctness of classification by Kohonen networks. This shows that indeed, considering word consecutiveness graphs of documents and measuring their similarity in a simple way captures sufficient amount of semantic similarity such that it can be used to improve significantly the similarity metric of documents. (We note that the computational costs are linear in terms of the number of data items in both cases, but in the case of augmented metric the similarity measure calculation costs are higher than in the case of the Euclidean metric.)

6 Conclusions

In this paper we discussed the application of Kohonen networks to text classification. Motivated by the need to include more semantics-based information in the similarity metric used for comparison of documents we proposed an augmented metric. The augmented metric is based on the comparison of word consecutiveness graphs of documents.

Our results show that indeed, the proposed augmented metric works significantly better than the simple Euclidean metric, leading to more correct classification of documents by trained Kohonen networks. This confirms our expectation that comparison of word consecutiveness graphs can capture a sufficient measure of semantic similarity of documents, to improve document classification performance.

The presented work opens the way for further research on developing more elaborated measures for semantic similarity on the basis of the word consecutiveness graphs of documents. We expect that in case of longer documents such more elaborated measure will contribute even more significantly to the increase of classification performance in case of Kohonen networks and possible also in the case of other document clustering methodologies.

References

[1] R.B. Kellog, M. Subhas (1996), Text to hypertext: can clustering solve the problem of digital libraries ?, *Proceedings of DL'96*, p. 144-150.

[2] F. Beil, M. Ester, X. Xu (2002), Frequent term-based text clustering, *Proceedings of SIGKDD'02*, p. 436-442.

[3] M., Steibach, G. Karypis, V. Kumar (2000), A comparison of document clustering techniques, Department of Computer Science and Engineering, University of Minnesota, *Technical Report #00-034*.

[4] A. Kaban, M.A. Girolami (2002), Fast extraction of semantic features from a latent semantic indexed text corpus, *Neural Processing Letters*, **vol. 15**, p. 31-43.

[5] E. Erosheva, S. Finberg, J. Lafferty (2004), Mixed-membership models of scientific publications, *PNAS*, **vol. 101**, p. 5220-5227.

[6] R.T. Freeman, H. Yin (2004), Adaptive topological tree structure for document organisation and visualisation, *Neural Networks*, **vol. 17**, p. 1255-1272.

[7] A. Skupin (2004), The world of geography: visualizing a knowledge domain with cartographic means, *PNAS*, **vol. 101**, p. 5274-5278.

[8] H.D. White, X. Lia, J.W. Buzdylowski, C. Chen (2004), User-controlled mapping of significant literatures, *PNAS*, **vol. 101**, p. 5297-5302.

[9] R. Ferrer y Cancho, R.V. Sole (2001), The small world of human language, *Proceedings of the Royal Society of London B*, **vol. 268**, p. 2261-2265.

[10] R. Cilibrasi, P. Vitanyi (2004), Automatic meaning discovery using Google, arXiv :cs.CL/0412098

[11] T. Kohonen (1995), *Self-organizing Maps*, Heidelberg, Springer-Verlag.

[12] T. Kohonen, P. Somervuo (2002), How to make large self-organising maps for nonvectorial data, *Neural Networks*, **vol. 15**, p. 945-952.

[13] J.T. Laaksonen, J.M. Koskela, E. Oja (2004), Class distributions on SOM surfaces for feature extraction and object retrieval, *Neural Networks*, **vol. 17**, p. 1121-1134.

[14] M. Cottrell, S. Ibbou, P. Letreny (2004), SOM-based algorithms for qualitative variables, *Neural Networks*, **vol. 17**, p. 1149-1168.

[15] D. Jimenez, E. Ferretti, V. Vidal, P. Rosso, C.F. Enguix (2003), The influence of semantics in IR using LSI and k-means clustering techniques, *ACM Transactions on Information Systems*, **vol. 22**, p. 279-284.

[16] P. Ginsparg, P. Houle, T. Joachims, J.-H. Sul (2004), Mapping subsets of scholarly information, *PNAS*, **vol. 101**, p. 5236-5240.

[17] M.F. Porter (1980) An algorithm for suffix stripping, *Program*, **vol. 14**, p. 130-137.

[18] J. Hopcroft, O. Khan, B. Kulis, B. Selman (2004), Tracking evolving communities in large linked networks, *PNAS*, **vol. 101**, p. 5249-5253.

[19] Q. Ma, K. Kanizaki, Y. Zhang, M. Murata, H. Isahara (2004), Self-organizing semantic maps and its application to word alignment in Japanese – Chinese parallel corpora, *Neural Networks*, **vol. 17**, p. 1241-1254.

[20] J. Berg, M. Lassig (2004), Local graph alignment and motif search in biological networks, *PNAS*, **vol. 101**, p. 14689-14694.

[21] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

[22] D.D. Lewis (1991), Representation and Learning in Information Retrieval, *Technical Report 91 – 93*, Computer Science Department, University of Massachusetts.