EVALUATION OF THREE DIMENSIONALITY REDUCTION TECHNIQUES FOR DOCUMENT CLASSIFICATION

Xiao Luo, A. Nur Zincir-Heywood
Dalhousie University, Faculty of Computer Science, NS, Canada
luo@cs.dal.ca, zincir@cs.dal.ca

Abstract

High dimensional document collections restrict the choice of data processing methods especially machine learning methods which need to calculate the inter-vector distances. This paper describes the development and evaluation of three different dimensionality reduction methods for document representation. Specifically, these methods are: Latent Semantic Indexing, Random Mapping and the two combined together. In this work, we are interested in how far these dimensionality reduction methods affect accurate measurement of document categorization. The results show that LSI performs better in terms of F1-measure, however RM+LSI has a very close performance record with a much less computational cost.

Keywords: *Dimensionality reduction; document classification; neural networks*

1. INTRODUCTION

In information retrieval, a typical data representation phase uses the Vector Space Model (VSM). There are two main parts to the VSM: Parsing and Indexing [2]. Parsing converts documents into a succession of words, whereas indexing represents documents by a vector and the number of dimensions of the vector is the number of different words in the collection. The values of each entry of the vector are then generally weighted using the term frequency multiplied by the inverse document frequency. With rapid increase of the complexity and variety of document set, this original VSM will face two main challenges: information overload and vocabulary differences [3]. Moreover, the availability of electronic documents has created high demand of automated text analysis tools. Thus, many machine-learning algorithms are widely used as automated text analysis tools. However, when the VSM representation is used as input to the machine learning algorithms, it may generate a high dimensional term space, which becomes problematic for machine learning algorithms when applied to large document collections [8]. The higher dimensional space will extremely slow down the training process if the training is based on inter-point distances. Thus, a dimensionality reduction phase becomes necessary to reduce the size of the vector space.

Hence in this paper, our goal is to analyze three different techniques for dimensionality reduction and to evaluate their performance for a multi-labeled data set by minimizing *a priori* assumptions regarding suitable word features. These techniques are: Latent Semantic Indexing (LSI), Random Mapping (RM) and a combination of the two (LSI+RM). Specifically, the work only uses the VSM with a simple stop list and basic stemming. After which one of the three methods is applied before finalizing the representation phase. Then, a hierarchical Self Organizing Feature Map (SOM) architecture is employed for each data reduction method for pattern discovery in the document space. After which, the similarity between a document and a category is measured using the cosine distance function. Finally, performances of the three different systems are measured. Based on the results obtained, LSI+RM gives the best trade-off between computational cost and classification performance.

The remainder of the paper is organized as follows. Section 2 provides the methodology and the details of data reduction algorithms. The learning algorithm used is introduced in Section-3, and results are given in Section-4. Finally, conclusions are drawn in Section 5.

2. METHODOLOGY

To achieve the above objectives, the framework of Figure 1 is followed, for all of the techniques assessed.

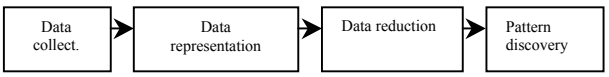

Fig.1. An overview of the architecture

2.1 Data Pre-Processing

First, the pre-processing is employed to the data set, before constructing the document vectors and employing any of the dimensionality reduction algorithms, Figure 2.

CCECE 2004- CCGEI 2004, Niagara Falls, May/mai 2004
0-7803-8253-6/04/$17.00 ©2004 IEEE

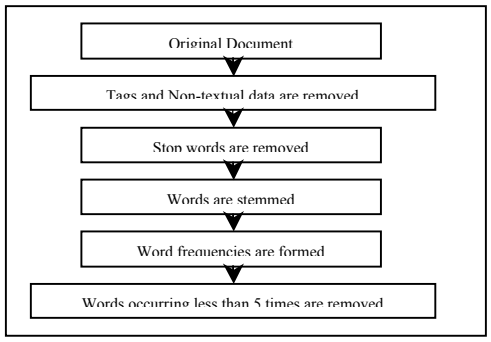

Fig. 2 . An overview of pre-processing

2.2 Dimensionality Reduction

After data pre-processing, VSM is used to identify the vectors to represent the documents. Then three dimension reduction algorithms described in the following sections are employed to reduce the original dimension.

2.2.1 Random Mapping. In the random mapping method, the data vector in the original matrix is reduced by replacing, each of its dimensions by a random non-orthogonal direction. The mapping [5] is as follows:

$$A'_{k\times m} = R_{k\times n} A_{n\times m} \qquad (1)$$

$A_{n\times m}$ – The original term-document matrix, n terms and m documents

$R_{k\times n}$ – A matrix consisting of random values, where the Euclidean length of each column has been normalized to unity

$A'_{k\times m}$ – Reduced-dimensional matrix

The reason why random direction works is based on the fact that there exists a much larger number of almost orthogonal directions than orthogonal directions in a high dimensional space [5]. Hence, even data vectors having random directions may be sufficiently close to being orthogonal. This in return provides resulting vectors that are close enough to representing original similarities in the data. Thus, the capabilities of Random Mapping depend fundamentally on how it affects the mutual similarities of data vectors.

2.2.2 Latent Semantic Indexing. Latent Semantic Indexing (LSI) is an information retrieval method for dimensionality reduction, where the emphasis is on capturing the underlying semantics or "latent" association in the pattern of terms or keywords used across documents. The mapping of original vectors into new vectors is based on the Singular Value Decomposition (SVD) applied to the original data vectors [1, 6].

Hence, let $A_{n\times m}$ be a $n\times m$ matrix of rank r, whose rows represent terms and columns represent documents in the corpus. Let the Eigen values of AA^T be $\partial_1 \geq \partial_2 \geq \ldots \geq \partial_r$. Then, the SVD of A becomes [6]:

$$A_{n\times m} = U_{n\times r} S_{r\times r} V_{m\times r}^{\ T} \qquad (2)$$

$U_{n\times r}$ – $(u_1, u_2, \ldots u_r)$ is the term-concept matrix, n terms, and r concepts, the columns are orthonormal

$S_{r\times r}$ – $diag(\partial_1, \partial_2, \ldots \partial_r)$, S_{ii} is the strength of concept i

$V_{m\times r}$ – $(v_1, v_2 \ldots, v_r)$ is the document-concept matrix, m documents, and r concepts, the columns are orthonormal

LSI works by omitting all but the k largest singular values in (3). Here, k is an appropriate value to represent the dimension of the low-dimensional space for the corpus. Hence, the approximation of $A_{n\times m}$ becomes:

$$A_{k\times m} = U_{n\times k} S_{k\times k} V_{m\times k}^{\ T} \qquad (3)$$

where the column vectors of $A_{n\times m}$ are projected to the k-dimensional space spanned by the column vectors of $U_{n\times k}$, and the rows of $V_{m\times k} S_{k\times k}$ are used to represent the documents. Thus, LSI preserves the relative distances in the original data set while projecting it to a lower dimensional space using techniques from linear algebra.

2.2.3 Random Mapping together with Latent Semantic Indexing. In this approach, we combined the two techniques in order to use their benefits and minimize their drawbacks. Random mapping can reduce the dimension of the document space but it does not necessarily guarantee to bring together semantically related data. On the other hand, LSI aims to achieve the latter, but its computation time is a bottleneck.

It should be noted that the computational costs for different approaches are quite different. Let $A_{n\times m}$ be a sparse data matrix with about c non-zero entries per column (i.e., c is the average number of terms in a document). The computational complexity of LSI is $\Omega(mnc)$, whereas the computational complexity of RM is $\Omega(mc\log n)$. Considering the combination of the two methods, if LSI were applied after RM, then the computational complexity would be $\Omega(m\log^2 n)$. Thus, if two techniques are combined together the total cost is $\Omega(m(\log^2 n + c\log n))$. Hence, the computational complexity of the combination of the two techniques is asymptotically superior to LSI: $\Omega(m(\log^2 n + c\log n))$ compared to $\Omega(mnc)$ [6].

Figure 3 shows the computational time under the same hardware environment of these three techniques. Obviously, the computational complexity of LSI is much higher than RM and the combination of them two. This naturally suggests first using Random Mapping to reduce the input space so that the computational complexity is

reduced. Then applying LSI to bring together semantically related documents, such an approach has already been shown theoretically in [6].

3. LEARNING ALGORITHM

Once the dimension of the input space is reduced, a two level hierarchical SOM architecture is developed to discover pattern similarities among the input space. The hierarchical nature of the architecture is shown in Figure 4.

3.1 Training the SOMs:

The algorithm responsible for the formation of the SOM involves three basic steps after initialization: sampling, similarity matching, and updating. These three steps are repeated until formation of the feature map has completed [5]. The algorithm is summarized as follows:

- Initialization: Choose random values for the initial weight vectors $w_j(0)$, $j=1,2, \ldots,l$, where l is the number of neurons in the map.
- Sampling: Draw a sample x from the input space with a certain probability.
- Similarity Matching: Find the best matching neuron $i(x)$ at time step n by using the minimum distance Euclidean criterion:
  $$i(x) = \arg\min_j \|x(n) - w_j\|, \quad j=1,2, \ldots,l \qquad (4)$$
- Updating: Adjust the weight vectors of all neurons by using the update formula:
  $$w_j(n+1) = w_j(n) + \eta(n)H(j,n)[x(n)-w_j(n)] \qquad (5)$$

where $\eta(n)$ is the learning rate at epoch n; and $H(j, n)$ is a suitable neighborhood function.

- Continuation: Continue with sampling until
  $$\Delta w = \text{average } |w_j(n+1) - w_j(n)| \leq \varepsilon \qquad (6)$$

where Δw is the average weight change of all neurons, and $\varepsilon > 0$ is a suitable constant. The sizes of the maps (25x20 for level-1, 4x4 or 6x6 for level-2) are chosen empirically according to the weight changes of neurons.

3.2 Input for the SOMs:

3.2.1 First-level SOMs. The input to the first level SOM is the reduced dimensional vector by one of the aforementioned three techniques. The assumption at this level is that similar document characteristics, i.e. words, will appear in the densely populated regions of the SOM for the patterns in words that occur in a specific document cluster.

3.2.2 Second-level SOMs. When a reduced dimensionality vector representing a document is run through a trained first-level SOM, certain neurons are affected more than others, depending on the variations in the statistics of the input distribution. However, due to the self-organizing characteristics of the map, some of the neurons may become more crowded with words (representing documents) as compared to others. Enlarging the size of the SOM solves the problem to some degree, but after a certain size the number of the crowded neurons remains the same [4]. Moreover, as the size of a map increases, also the computational cost increases. Therefore, instead of increasing the size of the SOM within the level, adding one more level for those crowded neurons can solve the problem by a divide-and-conquer technique [4]. Hence, each additional SOM network in the second-level is trained whenever a region in the first-level SOM was excited by more than fifty documents from 2 or more categories.

4. PERFORMANCE EVALUATION

In this work, a subset of the Reuters-21578 document collection [7] is used to evaluate the three different approaches. There are a total of 12612 news stories in this collection, where 9603 of them are in the training data set, and 3299 are in the test set. Among the test documents, 2526 documents belong to the top 25 categories. In our experiments, the training set consists of 9603 documents, whereas test set has 2526 documents.

As described in the previous sections, after the construction of document vectors, each document vector resulted with a dimension of 5612. Then, LSI and RM are employed respectively to reduce the dimension from 5612 to k, where k changes from 15 to 350. Finally, a combination of the two is employed together. To this end, first random mapping is applied with the reduced dimension k, where it has the best performance, and then the LSI is applied with the best k value.

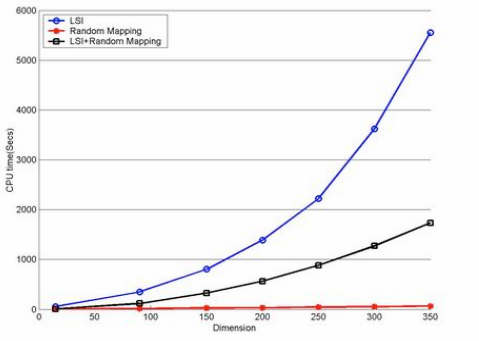

Fig.3. Computational cost of LSI, RM and LSI+RM

Once the reduced dimensions were obtained, SOM architecture is developed (for each approach) to discover the similarities among the documents. Then, typical classification quality measures such as F1-measure [9] are computed for each approach. In order to compute these,

the hit histograms are built for each document (*d*) and each category (*C)*, where these histograms represent the hit weights for each neuron on the map. The similarity between *d* and *C* is commonly measured using the cosine distance function. Thresholds from 0.01 to 0.3 increased by 0.01 are employed for the experiments. The threshold values represent the noise that is introduced to the system.

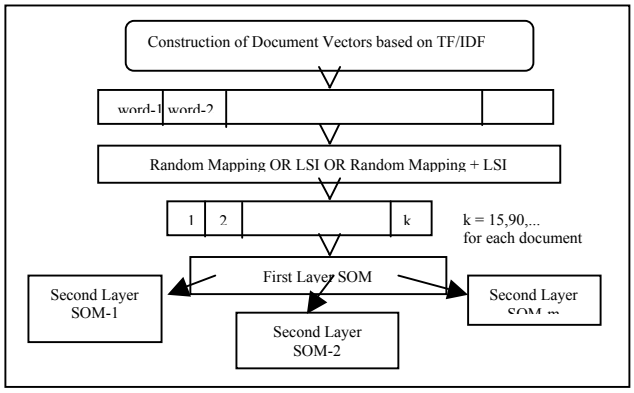

Fig. 4. The hierarchical SOM architecture

Both LSI and RM techniques seem to be giving the best performance when *k=350*. The best F1-measure for LSI (0.65) is better than the best F1-measure for RM (0.58). However, the difference in performance is only 0.07, yet the computational cost of *k=350* for RM is many magnitudes better than the computational cost of LSI where *k=350*, Figure 3.

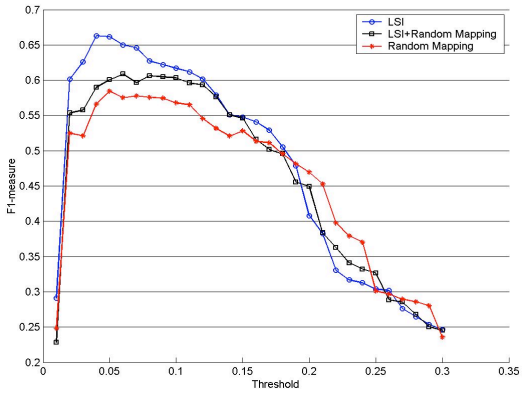

Fig.5. F1-measure for LSI, RM and LSI+RM where k=350

Therefore, to be able to decrease the computational cost of LSI but also keep the performance at a reasonable level, we combined the two techniques where k=350. Figure 5 shows the results for F1-measure for different thresholds (noise). According to the results of Figure 9, the combination of the two techniques increased the performance of the RM and had a competitive performance with the LSI. On the other hand, the computational cost of LSI+RM, where k=350, is much better than that of LSI only, Figure 3.

5. CONCLUSIONS

Three different techniques on data reduction have been developed for categorization of document collections. The first approach emphasizes RM to lower the computational cost of the training of the map and to decrease the dimension of the input space. The second approach emphasizes LSI to lower the dimensions and to bring together semantically related documents, whereas the third approach aims to combine the advantages of the two different approaches. The results show that LSI performs better than RM by approximately 13% in terms of F1-measure. However, RM outperforms the other two approaches in terms of computational cost, Figure 3, whereas the combination of the both (LSI+RM) gives a good performance in either of the evaluation methods. LSI+RM saves from the computational cost (by approximately 50%) compared to LSI. Furthermore, its performance is very similar to LSI under classification metrics, F1-measure. Finally, independent of the technique used, if the *k* value is not chosen properly, the documents will not be differentiated enough to be classified.

References

[1] R. K. Ando, L. Lee, "Iterative Residual Rescaling: An Analysis and Generalization of LSI," Proceedings of SIGIR'01, pp.154-162, 2001.

[2] G. Boone, "Concept features in Re: Agent, and intelligent e-mail agent," Proceedings of the 2nd International Conference on Autonomous Agents, ACM Press, pp.141–148. 1998.

[3] H. Chen, C. Schuffels, R. Orwig, "Internet categorization and search: A self-organizing approach," Journal of Visual Communication and Image Representation, Vol. 7, No.1, pp.88-102, 1996.

[4] U. Halici, G. Ongun, "Fingerprint classification through self-organizing feature maps modified to treat uncertainties", Proceedings of the IEEE, vol.84, no.10, pp.1497-1512, 1996.

[5] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," Proceedings of the IJCNN'98 Int. Joint Conf. Neural Networks, pp.413-418, 1998.

[6] C. H. Papadimitriou, P. Raghavan, H. Tamaki, S. Vempala, "Latent Semantic Indexing: a probabilistic analysis," Proceedings of the 17th ACM Symposium on the principles of Database Systems – PODS'98, pp.159-168, 1998.

[7] Reuters,www.daviddlewis.com/resources/testcollections/reuters21578/

[8] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys, vol.34, no.1, 2002, pp.1-47.

[9] Y. Yang, "An Evaluation of statistical approaches to text categorization," Information Retrieval, 1(1-2), pp.69-90, 1999.