Evaluation of Two Systems on Multi-class Multi-label Document Classification

Xiao Luo, A. Nur Zincir-Heywood

Faculty of Computer Science, Dalhousie University
6050 University Avenue, Halifax, NS, Canada B3H 1W5
{luo, zincir}@cs.dal.ca

Abstract. In the world of text document classification, the most general case is that in which a document can be classified into more than one category, the multi-label problem. This paper investigates the performance of two document classification systems applied to the task of multi-class multi-label document classification. Both systems consider the pattern of co-occurrences in documents of multiple categories. One system is based on a novel sequential data representation combined with a kNN classifier designed to make use of sequence information. The other is based on the "Latent Semantic Indexing" analysis combined with the traditional kNN classifier. The experimental results show that the first system performs better than the second on multi-labeled documents, while the second performs better on uni-labeled documents. Performance therefore depends on the dataset applied and the objective of the application.

1 Introduction

Text documents classification is one of the tasks of content-based document management. It is a problem of assigning a document into one or more classes. In multi-class document classification, there are more than two classes, however documents can be uni-labeled. On the other hand, in multi-label classification, a document may fall into more than one class, thus a multi-label classifier should be employed. For example, given classes "North America", "Asia", "Europe", a news article about the trade relationship between U.S.A and France may be labeled both to the "North America" and the "Europe" classes.

In this paper, we investigate the performance of two document classification systems regarding the task of multi-class multi-label document classification. Both of the systems consider the patterns of co-occurrences in documents or categories, and solve the high dimensionality problem of the traditional vector space model. The first system is based on a novel document representation technique developed by the authors – a hierarchical Self Organizing Feature Map (SOM) architecture – to encode the documents by first considering the relationships between characters, then words, and then word co-occurrences. After which, word and word co-occurrence sequences were constructed to represent each document. This system automates the identification of typi-

cal document characteristics, while considering the significance of the order of words in a document. In the second system, documents are first represented using the traditional vector space model (VSM), and then Latent Semantic Indexing is used to encode the original vectors into a lower dimensional space. It constructs the association within documents and terms from a "Semantic" view. The *k*-Nearest Neighbour (kNN) classification algorithm is employed at the stage categorization of both systems. However, to make use of the sequence information that the first system captures, new similarity function designed by the authors is employed, whereas kNN is applied using cosine similarity function to the second system. To evaluate both of the systems, experiments are performed on a subset of Reuters-21578 data set. The results show that the first system performs better than the second one on the multi-labeled documents; while the second performs better on the uni-labeled documents. The overall performances of both systems are very competitive, however the first system emphasizes the significant order of the words within a document, whereas the second system emphasizes the semantic association within the words and documents.

The rest of the paper is organized as follows: Section 2 presents the system framework and the data preprocessing applied to both systems. Section 3 describes the encoded data representation model designed by the authors, whereas section 4 presents the alternative Latent Semantic Indexing algorithm. The classification algorithm employed is described in section 5. Experiments performed and results obtained are presented in section 6. Finally, conclusions are drawn and future work is discussed in section 7.

2 System Framework and Document Pre-processing

The principle interest in this work is the scheme used to express co-occurrences consequently; the pre-processing and classification stages remain the same for both systems. Pre-processing removes all tags and non-textual data from the original document. Then a simple part-of-speech (POS) tagging algorithm [3] is used to select nouns from the document after which special nouns are removed. What is left is used as the input for both of the representation systems.

3 The Encoded Sequential Document Representation

In this document representation, we make use of the ability of an unsupervised learning system – Self Organizing Feature Map (SOM) – to provide approximations of a high dimensional input in a lower dimensional space [5, 6]. The SOM acts as an encoder to represent a large input space by finding a smaller set of prototypes.

Thus, in this work, a hierarchical SOM architecture is developed to understand the documents by first encoding the relationships between characters, words, and then words co-occurrences. The hierarchical nature of the model and the corresponding U-matrix of trained SOMs are shown in Figure 1. In this model, the system is defined by the following three steps:

1) Input for the First-Level SOMs: The assumption at this level is that the SOM forms a codebook for the patterns in characters that occur in a specific document category. In order to train an SOM to recognize patterns in characters, the document data must be formatted in such a way as to distinguish characters and highlight the relationships between them. Characters can easily be represented by their ASCII representation. However, for simplicity, we enumerated them by the numbers 1 to 26, i.e. no differentiation between upper and lower case. The relationships between characters are represented by a character's position, or time index, in a word. It should be noted that it is important to repeat these words as many times as they occur in the documents, so that the neurons on the SOM will be more excited by the characters of the more frequent words and their information will be more efficiently encoded.

2) Input for the Second-Level SOMs: The assumption at this level is that the 2nd level SOM forms a codebook for the patterns in words that occur in a specific document category. When a character and its index are run through a trained first-level SOM, the closest neurons (in the Euclidian sense), or *Best Matching Units* (*BMU*s), are used to represent the input space. A two-step process is used to create a vector for each word, k, which is input to the first-level SOM of each document:

- Form a vector of dimension equal to the number of neurons (r) in the first-level SOM, where each entry of the vector corresponds to a neuron on the first-level SOM, and initialize each entry of the vector to 0.
- For each character of word k,
    - Observe which neurons $n_1, n_2, \ldots, n_r$ are closest.
    - Increase entries in the vector corresponding to the 3 most affected *BMU*s by $1/j$, $1 \leq j \leq 3$.

Hence, each vector represents a word through the sum of its characters. The results given by the second-level SOM are clusters of words on the second-level SOM.

3) Input for the Third-Level SOMs: The assumption at this level is that the SOM forms a codebook for the patterns in word co-occurrence that occur in a specific document category. In the context of this architecture, word co-occurrence is simply a group of consecutive words in a document. The consecutive words are from a single document with a sliding window of size 3. The input space of the third-level SOM is formed in a similar manner to that in the second-level, except that each word in the word co-occurrences is encoded in terms of the indexes of the 3 closest *BMU*s resulting from word vectors passed through the second-level SOMs. Thus, the length of the input vector to the third-level SOM is 9. The results given by third-level SOM are clusters of word co-occurrence on the third-level SOM.

After training the SOMs, we analyzed the results of the trained second and third level SOMs. We observed that documents from the same category always excited the same or similar parts of the second and third level SOMs respectively, as well as sharing *BMU* sequences with each other, Figures 2-3. Moreover, we observed that the different categories have their own characteristic most frequent *BMU* sequences. Based on these observations, we propose a document representation system where each document has two sequence representations: The first one is based on the second-level SOM, i.e. the word based sequence representation. The other is based on the third-level SOM, i.e. the word co-occurrence based sequence representation. These two sequence

representations are combined together during the classification stage. The sequence representation (word/word co-occurrence) is defined as a two dimensional vector. The first dimension of the vector is the index of the *BMU*s on the second (third) level SOM; the second dimension is the Euclidean distance to the corresponding *BMU*s.

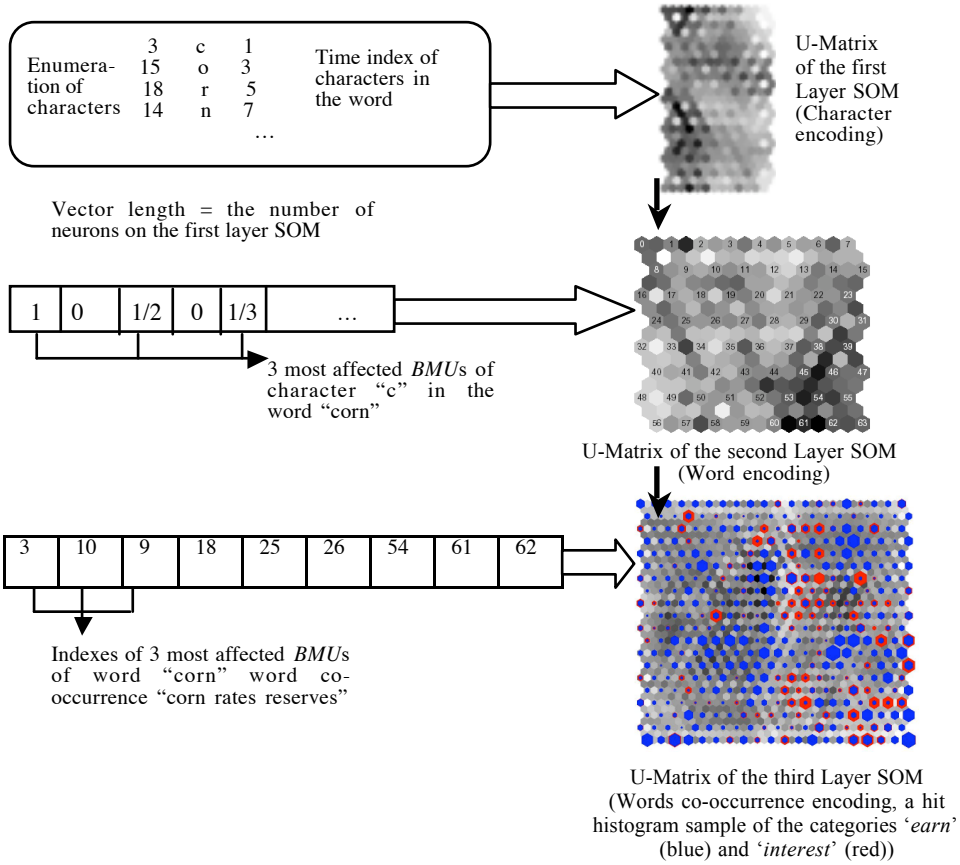

Fig. 1. An overview of the hierarchical SOMs encoding architecture

4 Latent Semantic Indexing (LSI)

A lot of research has been performed using Latent Semantic Indexing (LSI) for information retrieval with many good results for text retrieval, in particular [2, 4]. However, very little consideration has been given to the performance of LSI on multi-class multi-label document classification problems. LSI emphasizes capturing words or term co-occurrences based on the semantics or "latent" associations. The mapping of the original vectors into new vectors is based on Singular Value Decomposition (SVD) applied to the original data vectors [1, 7]. Usually, the original document vec-

tors are constructed using the so-called vector space model and the TF/IDF weighting schema. There are many TF/IDF weighting schemas. The one we used here is:

$$P_{ij} = tf_{ij} \cdot log\ (N/df_i) \qquad (1)$$

P_{ij}: Weight of term t_j in document d_i
tf_{ij}: Frequency of term t_j in document d_i
N: Total number of documents in corpus
df_i: Number of documents containing term t_j

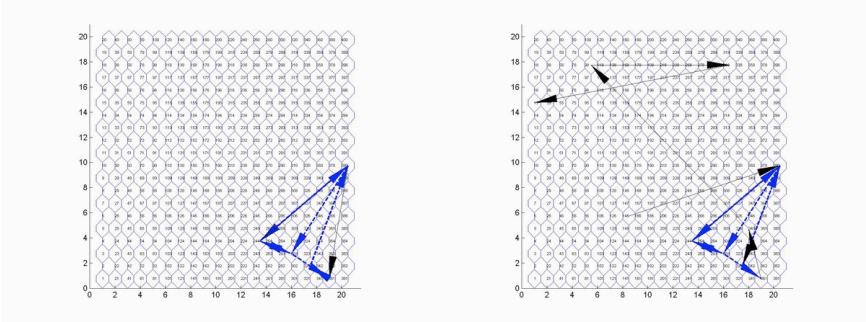

Fig. 2. BMU sequences of two documents from category "Earn" on the second-level SOM

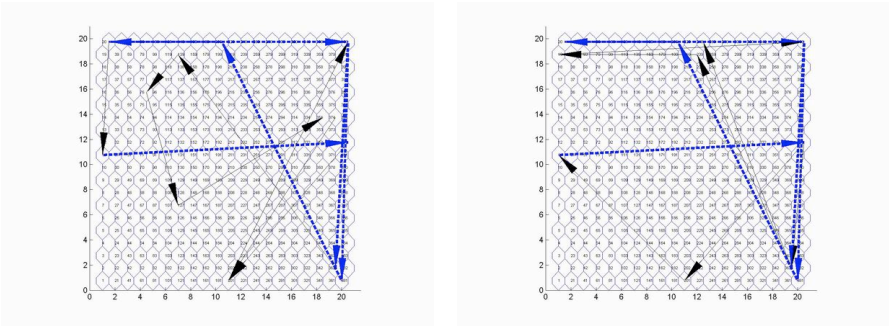

Fig. 3. BMU sequences of two documents from category "Earn" on the third-level SOM

Moreover, LSI omits all but the k largest singular values. Here, k is an appropriate value to represent the dimension of the low-dimensional space for the corpus. Hence, the approximation of $A_{n \square m}$ becomes [7]:

$$\hat{A}_{n \square m} = U_{n \square k} S_{k \square k} V_{m \square k}^{T} \qquad (2)$$

A_{nxm}- is a $n \square m$ matrix representing documents
$U_{n \square r} - (u_1, u_2, \ldots u_r)$ is the term-concept matrix
$S_{r \square r} - diag(\partial_1, \partial_2, \ldots \partial_r)$, S_{ii} is the strength of concept i
$V_{m \square r} - (v_1, v_2 \ldots, v_r)$ is the document-concept matrix

In the experiments performed here, a large range of k values (from 25 to 1125) was explored to fully investigate its efficiency on the multi-class multi-label documents classification.

5 The Document Classification Algorithm Employed

The k-Nearest Neighbour *(*kNN) classifier algorithm has been studied extensively for text categorization by Yang and Liu [8]. The kNN algorithm is quite simple: To classify a test document, the k-Nearest Neighbour classifier algorithm finds the k nearest neighbours among the training documents, and uses category labels of the k nearest training documents to predict the category of the test document. The similarity in score of each neighbour document to the test document is used as the weight of the categories of the neighbour document [8]. If there are several training documents in the k nearest neighbour, which share a category, the category gets a higher weight.

In this work, we used the Cosine distance (3) to calculate the similarity score for the document representation based on LSI. However, since the Cosine distance measurement does not fit the encoded sequential data representation, we designed a similarity measurement formula (4) and (5). Both (4) and (5) consider the length of the shared *BMU* sequences, in which (4) emphasizes the similarity degree of the *BMU*s in a sequence, while (5) emphasizes the length of the document to be measured. After weighting the categories by using (4) and (5), we normalize the weight and combine the normalized weight of each category by adding them together.

$$Sim(D_i, D_j) = \frac{D_i \cdot D_j}{\|D_i\|_2 \times \|D_j\|_2} \quad (3)$$

$$Sim(D_i, D_j) = \sum_{k=1}^{n} \frac{100}{1 + dist(W_{ik}, W_{jk})} \times n \quad (4)$$

$$Sim(D_i, D_j) = \frac{\sum_{k=1}^{n} \frac{1}{1 + dist(w_{ik} + w_{jk})}}{length(D_j)} \quad (5)$$

D_i : Test document to be categorized.
D_j : Document in the training set.
n : Total number of *BMU*s in common *BMU* sequences of D_i and D_j.
W: Euclidean distance of a word to the corresponding *BMU*
$dist(W_{ik}, W_{jk})$: The distance between W and the corresponding *BMU* in the common *BMU* sequences of D_i and D_j. This shows the similarity between words (word co-occurrences)
$length(D_j)$: Length of the document in the training set in terms of *BMU* sequences.

6 Experimental setup and results

In this work, we employed a well-known multi-class multi-label document data set - Reuters-21578[1]. There are a total of 12612 news stories in this collection. These stories are in English, where 9603 are pre-defined as the training set, and 3299 are

[1] Reuters data set, http://www.daviddlewis.com/resources/testcollections/reuters21578/

pre-defined as the test set. In our experiments, all 9603 training files are used, which belong to 118 categories. In order to fully analyze the performance of the systems for the multi-class multi-label document classification problem, we chose 6 of the top 10 categories of the Reuters-21578 data set to test the categorization performance. These categories are "Earn", "Money-fx", "Interest", "Grain", "Wheat" and "Corn". The relationships between these categories in the training set are shown in Figure 4, and are the same as their relationships in the test set. Moreover, there are 8231 nouns left after pre-processing; where it is these nouns that are used to build the SOM and LSI representations.

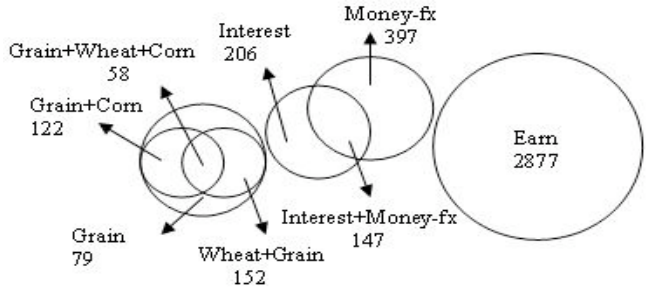

Fig. 4. Size and relationship of the six categories in the training set

Facing a multi-labeled (*N* labels) test document, we classify the test document to the top *N* weighted categories. The F1-measure is used to measure performance. We experiment with four kNN limits for the SOM representation — k = 3, 5, 10 and 15, whereas seven kNN limits (k = 25, 115, 225, 425, 525, 625, and 1125) for the LSI representation. In this case, our experiments show that "k = 5" gives the best performance in terms of the Macro F1-measure score.

$$R = \frac{TP}{TP + FN} \quad (6) \qquad P = \frac{TP}{TP + FP} \quad (7) \qquad F1\text{-}measure = \frac{2RP}{R + P} \quad (8)$$

TP : Positive examples, classified to be positive.
FN : Positive examples, classified to be negative.
FP: Negative examples, classified to be positive.

Tables 1 and 2 show most of the results on multi-labeled documents and uni-labeled documents, respectively. Table 3 shows the Macro F1-measure of the multi-labeled and uni-labeled documents. It is apparent that both systems have very competitive performance on the task of multi-class multi-label document classification. From the returned Macro-F1 value, the system based on the sequential document representation performs better than the system based on LSI. On the other hand, LSI works better, where each document is based on a single topic. The experiments show that LSI does work better on the uni-labeled documents on some *k* values (25 or 125). However, it works worse on the multi-labeled documents whatever the *k* value is.

The encoded sequential document representation can capture the characteristic sequences for documents and categories. Good performance is achieved by utilizing the sequence information for classification. The results show that it works better for the relatively larger categories, such as "Money+Interest", "Grain+Wheat" and "Earn". We conclude the reasons behind this is: This data representation is based on the machine-

learning algorithm to capture the characteristic word or word co-occurrences for categories, so the more frequent the word or word co-occurrences are, the more easily it can be caught and represented by the neurons of the hierarchical SOM based architecture developed here. In the experimental corpus, the larger categories present more frequent word or word co-occurrences to the architecture.

Table 1. Classification results of the multi-labeled documents[2,3]

Category	*Size in Test set*	*F1-Measure*							
		SDR	*LSI (25)*	*LSI (125)*	*LSI (225)*	*LSI (425)*	*LSI (525)*	*LSI (625)*	*LSI (1125)*
M+I	43	0.86	0.79	0.73	0.74	0.80	0.78	0.78	0.67
G+C	34	0.55	0.44	0.54	0.59	0.59	0.58	0.52	0.56
G+W	49	0.76	0.62	0.43	0.71	0.73	0.71	0.69	0.65
G+C+W	22	0.75	0.98	0.88	0.90	0.85	0.80	0.86	0.76
Micro-F1 Measure		0.74	0.68	0.61	0.72	0.73	0.71	0.70	0.65

Table 2. Classification results of the uni-labeled documents[3]

Category	*Size in Test set*	*F1-Measure*							
		SDR	*LSI (25)*	*LSI (125)*	*LSI (225)*	*LSI (425)*	*LSI (525)*	*LSI (625)*	*LSI (1125)*
Earn	1087	0.97	0.97	0.97	0.96	0.96	0.96	0.95	0.94
Money	136	0.61	0.68	0.67	0.64	0.58	0.58	0.56	0.56
Interest	88	0.44	0.54	0.61	0.56	0.57	0.58	0.58	0.51
Grain	44	0.35	0.40	0.35	0.34	0.30	0.29	0.23	0.29
Micro-F1 Measure		0.88	0.89	0.89	0.88	0.87	0.87	0.86	0.85

Table 3. The overall classification results of multi-labeled and uni-labeled documents[2]

	SDR	*LSI (25)*	*LSI (125)*	*LSI (225)*	*LSI (425)*	*LSI (525)*	*LSI (625)*	*LSI (1125)*
Macro-F1	0.81	0.78	0.75	0.80	0.80	0.75	0.75	0.75

7 Conclusion and Future Work

Through this work, we explored the performance of two document classification systems regarding the task of multi-class multi-label document classification. Both of the systems consider the patterns of co-occurrences in documents or categories, and solve the high dimensionality problem associated with the traditional vector space model. However, the first system considers the term co-occurrences from the aspect of the order of the terms within documents, while the Latent Semantic Indexing considers the

[2] M+I: "Money-fx+Interest"; G+C: "Grain+Corn"; G+W: "Grain+Wheat"; G+C+W: "Grain+Corn+Wheat"

[3] SDR: Sequential Document Representation; LSI (n): Latent Semantic Indexing with *k* value=n.

term co-occurrences from the aspect of semantic association within them and documents.

The experimental results show that the SOM representation performs better than the LSI on the multi-labeled documents; while LSI performs better on the uni-labeled documents. The over all performances of both systems are very competitive. The encoding mechanism of the SOM system can efficiently work on both textual and non-textual data. Since the order and the frequency of patterns are maintained as they appear before input to the encoding architecture, it is expected to perform better for other data classification applications such as medical or business information systems where sequence information is more significant and important. Moreover, it can be used for online data classification without seeing all the terms or patterns in the whole corpus. The methodology of the LSI system is easy to use and discovered the semantic associations between the terms within the documents. It is very accurate when each document is on a single topic and the terms are partitioned among the topics such that each topic distribution has high probability on its own terms.

The idea of sequential data representation for document classification is new, more improvements such as looking into the integration of different sizes of the SOMs for the encoding architecture, investigating other styles of representations on top of the encoding architecture, will be done in the future. In addition, other classifiers, which explicitly support sequence classification, will also be analyzed and utilized in the later research.

References

1. Ando, R. K., Lee L.: Iterative Residual Rescaling: An Analysis and Generalization of LSI. Proceedings of the ACM SIGIR'01 (2001) 154-162
2. Berry, M. W., Dumais, S. T., O'Brien G. W.: Using Linear Algebra for Intelligent Information Retrieval. SIAM Review, 37(4) (1995) 573-595
3. Brill, E.: A Simple Rule Based Part of Speech Tagger. Proceedings of the 3rd Conference on Applied Natural Language Processing (1992) 152-155
4. Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A.: Indexing by Latent Semantic Analysis. Journal of the Society for Information Science 41(6) (1990) 391-407
5. Haykin, S.: Neural Networks - A Comprehensive Foundation, Chapter-9: Self-Organizing Maps. Second Edition, Prentice Hall (1999) ISBN 0-13-273350-1
6. Luo, X., Zincir-Heywood, A. N.: A Comparison of SOM Based Document Categorization Systems. Proceedings of the IEEE International Joint Conference on Neural Networks (2003) 1786-1791
7. Papadimitriou, C. H., Raghavan, P., Tamaki, H., Vempala, S.: Latent Semantic Indexing: A probabilistic Analysis. Proceedings of the 17th ACM Symposium on the Principles of Database Systems (1998) 159-168
8. Yang Y., Liu X.: A Re-examination of Text Categorization Methods. Proceedings of the ACM SIGIR'99 (1999) 42-49