Multi-Label Text Classification using VG-RAM Weightless Neural Networks

Claudine S. B. Gonçalves, Felipe T. Pedroni, and Alberto F. De Souza

Departamento de Informática, Universidade Federal do Espírito Santo, Brazil
claudinebadue@dcc.ufmg.br, felthomaz@gmail.com, albertodesouza@gmail.com

Abstract. In automated multi-label text classification, each text document is associated with a set of classes and an automatic classification system should output a class set, whose size is unknown a priori, for each document under classification. Many machine learning techniques have been used for building such automatic text classification systems. Virtual generalizing random access memory weightless neural networks (VG-RAM WNN) is an effective machine learning technique which offers simple implementation, and fast training and test. In this work, we have trained VG-RAM WNN with a multi-label Web database and evaluated its classification performance using multi-label classification metrics. We compared our results with that of the boosting-style algorithm BOOSTEXTER, the multi-label kernel method RANK-SVM, the multi-label decision tree ADTBOOST.MH, and the multi-label lazy learning approach ML-KNN. Our experimental comparative analysis showed that, on average, VG-RAM WNN either outperforms the other mentioned techniques or show similar classification performance.

1 Introduction

Automatic text classification is still a very challenging computational problem to the information retrieval communities both in academic and industrial contexts. Most works on text classification in the literature are focused in single-label text classification [1]. However, in real-world problems, multi-label classification is frequently necessary [2–11].

From a theoretical point of view, single-label classification is more general than multi-label, since an algorithm for single-label classification can also be used for multi-label classification: one needs only to transform the multi-label classification problem into n independent single-label problems, where n is number of possible classes (labels) [1]. However, this equivalence between single-label and multi-label classification only holds if the n classes are stochastically independent, that is, the association of a class c_k to a document is independent of the association of another class, c_l, to the same document, but this frequently is not the case. Multi-label classification systems can take advantage of the correlation between classes in order to improve their performance.

Several approaches specially designed for multi-label classification have been proposed, such as multi-label decision trees [4, 6], multi-label kernel methods [5, 8, 9], and multi-label text classification [2, 3, 7, 12]. In this work, we present an experimental evaluation of the performance of virtual generalizing random access memory weightless neural networks (VG-RAM WNN [13]) on multi-label text classification. VG-RAM WNN is an effective machine learning technique which offers simple implementation, and fast training and test [14]. We have trained VG-RAM WNN with a multi-label Web database (Web pages categorized by "yahoo.com") and evaluated its classification performance using the *hamming loss*, *one-error*, *coverage*, *ranking loss*, and *average precision* multi-label classification metrics [3]. We compared the VG-RAM WNN classification performance, according to these metrics, with that of the boosting-style algorithm BOOSTEXTER [3], the multi-label kernel method RANK-SVM [5], the multi-label decision tree ADTBOOST.MH [6], and the multi-label lazy learning techniques ML-KNN [11]. Our results showed that, on average, VG-RAM WNN either outperforms these techniques or show similar classification performance.

This paper is organized as follows. Section 2 introduces the multi-label text classification problem and the metrics used to evaluate the performance of the multi-label classifiers examined. Section 3 briefly introduces VG-RAM WNN and describes how we have used it for multi-label text classification. Section 4 presents our experimental methodology and analyzes our experimental results. Our conclusions and directions for future work follow in Section 5.

2 Multi-Label Text Classification

Text classification may be defined as the task of assigning documents to a pre-defined set of classes [1]. Let $\mathcal{D}$ be the domain of documents, $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ a set of pre-defined classes, and $\Omega = \{d_1, \ldots, d_{|\Omega|}\}$ an initial corpus of documents previously classified under the set of classes $\mathcal{C}$. In multi-label learning, the training(-and-validation) set $TV = \{d_1, \ldots, d_{|TV|}\}$ is composed of a number documents, each associated with a set of classes. TV is used to train a classifier that associates the characteristics of each document in the TV to the appropriate combination of classes. The test set $Te = \{d_{|TV|+1}, \ldots, d_{|\Omega|}\}$, on the other hand, consists of documents for which the classes are unknown to the classifier. After being trained, the classifier is used to predict the set of classes for such test documents.

A multi-label classifier typically implements a real-valued function of the form $f : \mathcal{D} \times \mathcal{C} \rightarrow \mathbb{R}$ that returns a value to each pair $\langle d_j, c_i \rangle \in \mathcal{D} \times \mathcal{C}$ that, roughly speaking, represents the evidence for the fact that the test document d_j should be classified under the class c_i. The real-valued function $f(.,.)$ can be transformed to a ranking function $r(.,.)$ that is a one-to-one mapping onto $\{1, 2, \ldots, |\mathcal{C}|\}$, such that if $f(d_j, c_1) > f(d_j, c_2)$ then $r(d_j, c_1) < r(d_j, c_2)$. If C_j is the set of proper classes for the test document d_j, then a successful classifier will tend to rank classes in C_j higher than those not in C_j. Those classes that rank above a threshold τ (i.e., $c_k | f(d_j, c_k) \geq \tau$) are then assigned to the test document.

The evaluation of multi-label classifiers is more complex than the evaluation of the traditional single-label ones. The most popular metrics in the literature for single-label learning systems are precision, recall and the F-measure [1]. For evaluating the classification performance of VG-RAM WNN, we have used five multi-label evaluation metrics proposed in [3]:

Hamming Loss ($\mathbf{hloss}_j$): evaluates how many times the test document d_j is misclassified, i.e., a class not belonging to the document is predicted or a class belonging to the document is not predicted.

$$\text{hloss}_j = \frac{1}{|\mathcal{C}|}|P_j \Delta C_j| \tag{1}$$

where $|\mathcal{C}|$ is the number of classes and Δ is the symmetric difference between the set of predicted classes P_j and the set of appropriate classes C_j of the test document d_j.

One-error ($\mathbf{one\text{-}error}_j$): evaluates if the top ranked class is present in the set of proper classes C_j of the test document d_j.

$$\text{one-error}_j = \begin{cases} 0 \text{ if } [\arg\max_{c \in \mathcal{C}} f(d_j, c)] \in \mathcal{C} \\ 1 \text{ otherwise.} \end{cases} \tag{2}$$

where $[\arg\max_{c \in \mathcal{C}} f(d_j, c)]$ returns the top ranked class for the test document d_j.

Coverage ($\mathbf{coverage}_j$): measures how far we need to go down the rank of classes in order to cover all the possible classes assigned to a test document. It is loosely related to precision at the level of perfect recall.

$$\text{coverage}_j = \max_{c \in C_j} r(d_j, c) - 1 \tag{3}$$

where $\max_{c \in C_j} r(d_j, c)$ returns the maximum rank for the set of appropriate classes of the test document d_j.

Ranking Loss ($\mathbf{rloss}_j$): evaluates the fraction of class pairs $\langle c_k, c_l \rangle$, for which $c_k \in C_j$ and $c_l \notin C_j$, that are reversely ordered (i.e., $r(d_j, c_l) > r(d_j, c_k)$) for the test document d_j.

$$\text{rloss}_j = \frac{|\{(c_1, c_2)|f(c_i, y_1) \leq f(c_i, y_2), (y_1, y_2) \in C_j \times \bar{C}_j\}|}{|C_j||\bar{C}_j|} \tag{4}$$

where $\bar{C}_j$ is the complementary set of C_j in $\mathcal{C}$.

Average Precision ($\mathbf{avgprec}_j$): evaluates the average fraction of classes ranked above a particular class $c \in C_j$ which actually are in C_j. This is the non-interpolated average precision, a metric frequently used for evaluation of information retrieval systems [15]. We note that the non-interpolated average precision is typically used in information retrieval systems to evaluate the document ranking for query retrieval. In contrast, in our experiments we use average precision for evaluating class rankings.

$$\text{avgprec}_j = \frac{1}{|C_j|} \sum_{c \in C_j} \frac{|\{c' | r(d_j, c') \leq r(d_j, c), c' \in C_j\}|}{r(d_j, c)} \tag{5}$$

For p test documents, the overall performance is obtained by averaging each metric, that is $\text{hloss} = \frac{1}{p}\sum_{j=1}^{p}\text{hloss}_j$, $\text{one-error} = \frac{1}{p}\sum_{j=1}^{p}\text{one-error}_j$, $\text{coverage} = \frac{1}{p}\sum_{j=1}^{p}\text{coverage}_j$, $\text{rloss} = \frac{1}{p}\sum_{j=1}^{p}\text{rloss}_j$, and $\text{avgprec} = \frac{1}{p}\sum_{j=1}^{p}\text{avgprec}_j$. The smaller the value of *hamming loss*, *one-error*, *coverage* and *ranking loss*, and the larger the value of *average precision*, the better the performance of the classifier. The performance is perfect when $\text{hloss} = \text{one-error} = \text{rloss} = 0$ and $\text{avgprec} = 1$.

3 Weightless Neural Network

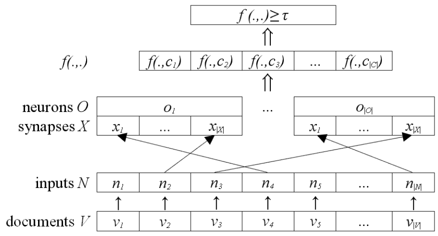
RAM-based neural networks, also known as n-tuple classifiers or weightless neural networks (WNNs), do not store knowledge in their connections but in Random Access Memories (RAMs) inside the network's nodes, or neurons. These neurons operate with binary input values and use RAMs as lookup tables: the synapses of each neuron collect a vector of bits from the network's inputs that is used as the RAM address, and the value stored at this address is the neuron's output. Training can be made in one shot and basically consists of storing the desired output in the address associated with the input vector of the neuron [16].

In spite of their remarkable simplicity, RAM-based neural networks are very effective as pattern recognition tools, offering fast training and easy implementation [17]. However, if the network input is too large, the memory size becomes prohibitive, since it must be equal to 2 to the power of the input size. Virtual Generalizing RAM (VG-RAM) networks are RAM-based neural networks that only require memory capacity to store the data related to the training set [13]. In the neurons of these networks, the memory stores the input-output pairs shown during training, instead of only the output. In the test phase, the memory of VG-RAM neurons is searched associatively by comparing the input presented to the network with all inputs in the input-output pairs learned. The output of each VG-RAM neuron is taken from the pair whose input is nearest to the input presented—the distance function employed by VG-RAM neurons is the *hamming distance*. If there is more than one pair at the same minimum distance from the input presented, the neuron's output is chosen randomly among these pairs.

Figure 1(a) shows the lookup table of a VG-RAM neuron with three synapses (X_1, X_2 and X_3). This lookup table contains three entries (input-output pairs), which were stored during the training phase (entry #1, entry #2 and entry #3). During the test phase, when an input vector (input) is presented to the network, the VG-RAM test algorithm calculates the distance between this input vector and each input of the input-output pairs stored in the lookup table. In the example of Figure 1(a), the *hamming distance* from the input to entry #1 is two, because both X_2 and X_3 bits do not match the input vector. The distance to entry #2 is one, because X_1 is the only non-matching bit. The distance to entry #3 is three, as the reader may easily verify. Hence, for this input vector, the algorithm evaluates the neuron's output, Y, as class 2, since it is the output value stored in entry #2.

lookup table	X_1	X_2	X_3	Y
entry #1	1	1	0	label 1
entry #2	0	0	1	label 2
entry #3	0	1	0	label 3
input	1	0	1	**label 2**

(a)



(b)

Fig. 1. (a) VG-RAM neuron lookup table; (b) VG-RAM WNN architecture employed.

To classify text documents using VG-RAM WNN, we represent a document as a multidimensional vector $V = \{v_1, \ldots, v_{|V|}\}$, where each element v_i corresponds to the number of times a term in the vocabulary of interest appears in this document. We use single layer VG-RAM WNN (Figure 1(b)) whose neurons' synapses $X = \{x_1, \ldots, x_{|X|}\}$ are randomly connected to the network's inputs $N = \{n_1, \ldots, n_{|N|}\}$, which has the same size of the vectors representing the documents, i.e., $|N| = |V|$. Note that $|X| < |V|$ (our experiments have shown that $|X| < |V|$ provides better performance). Each neuron's synapse x_i forms a minchinton cell with the next, x_{i+1} ($x_{|X|}$ forms a minchinton cell with x_1) [18]. The type of the minchinton cell we have used returns 1 if the synapse x_i of the cell is connected to an input element n_j whose value is larger than that of the element n_k to which the synapse x_{i+1} is connected (i.e. $n_j > n_k$); otherwise, it returns zero. During training, for each document in the training set, the corresponding vector V is connected to the VG-RAM WNN' inputs N and the neurons outputs $O = \{o_1, \ldots, o_{|O|}\}$ to one of its classes. All neurons of the VG-RAM WNN are then trained to output this class with this input vector. The training for this input vector is repeated for each class associated with the corresponding document. During test, for each test document, the inputs are connected to the corresponding vector and the number of neurons outputting each class is counted. The network output is computed by dividing the count of each class by the number of neurons of the network. This output is organized as a vector whose size is equal to the number of classes. The value of each vector element varies from 0 to 1 and represents the percentage of neurons which presented the corresponding class as output (the sum of the values of all elements of this vector is always equal to 1). This way, the output of the network implements the function $f(.,.)$, defined in Section 2. A threshold τ may be used with the function $f(.,.)$ to define the set of classes to be assigned to the test document.

4 Experimental Evaluation

In this section, we present our experimental methodology and analyze our experimental results.

Table 1. Characteristics of the Web page data sets (after term selection). NC denotes the number of classes, SV denotes the size of the vocabulary, PMC denotes the percentage of documents belonging to more than one class, ANL denotes the average number of classes for each document, and PRC denotes the percentage of rare classes, i.e. the kind of class where only less than 1% documents in the data set belong to it.

Data set	NC	SV	Training(-and-validation) set			Test set		
			PMC	ANL	PRC	PMC	ANL	PRC
Arts&Humanities	26	462	44.50%	1.63	19.23%	43.63%	1.64	19.23%
Business&Economy	30	438	42.20%	1.59	50.00%	41.93%	1.59	43.33%
Computers&Internet	33	681	29.60%	1.49	39.39%	31.27%	1.52	36.36%
Education	33	550	33.50%	1.47	57.58%	33.73%	1.46	57.58%
Entertainment	21	640	29.30%	1.43	28.57%	28.20%	1.42	33.33%
Health	32	612	48.05%	1.67	53.13%	47.20%	1.66	53.13%
Recreation&Sports	22	606	30.20%	1.41	18.18%	31.20%	1.43	18.18%
Reference	33	793	13.75%	1.16	51.52%	14.60%	1.18	54.55%
Science	40	743	34.85%	1.49	35.00%	30.57%	1.43	40.00%
Social&Science	39	1 047	20.95%	1.27	56.41%	22.83%	1.29	58.97%
Society&Culture	27	636	41.90%	1.71	25.93%	39.97%	1.68	22.22%

4.1 Data Set

Ueda and Saito [7] evaluated the performance of probabilistic generative models on the classification of real World Wide Web pages. They tried to classify real Web pages linked from the “yahoo.com” directory. The Yahoo directory consists of 14 top-level classes (i.e., "Arts & Humanities", "Business & Economy", "Computers & Internet", and so on), and each class is categorized into a number of second-level subclasses. By focusing on these sub-classes, one can devise 14 independent text classification problems. Ueda and Saito studied 11 of these 14 problems. Zhang and Zhou [11] proposed a lazy learning approach to multi-label learning and used the same 11 text classification problems (of Ueda and Saito [7]) to evaluate the performance of their multi-label learning algorithm. To reduce the dimensionality of each data set, they used a simple term selection method based on document frequency (the number of documents containing a specific term). Only the top 2% terms with highest document frequency were retained in the final vocabulary. After term selection, each document in the data set was described as a multidimensional vector using the “Bag-of-Words” representation [19], i.e., each dimension of the vector corresponds to the number of times a term in the vocabulary appears in the corresponding document. We used Zhang and Zhou’s data sets to evaluate the performance of VG-RAM WNN. Table 1 summarizes the characteristics of the Web page data sets[1]. For each problem, the training(-and-validation) set contains 2000 documents while the test set contains 3000 documents.

[1] The characteristics of the Web page data sets were obtained from the work presented in [11].

4.2 Results

To implement VG-RAM WNN, we used the Event Associative Machine (MAE) [20], an open source framework for modeling VG-RAM neural networks developed at the Universidade Federal do Espírito Santo.

Table 2. Parameters of VG-RAM WNN that yield the best performance.

Data set	Number of neurons	Number of synapses	Threshold τ
Arts&Humanities	1024	64	0.2
Business&Economy	1024	64	0.2
Computers&Internet	1024	64	0.4
Education	1024	128	0.4
Entertainment	1024	128	0.3
Health	1024	128	0.2
Recreation&Sports	1024	64	0.2
Reference	1024	64	0.5
Science	1024	64	0.2
Social&Science	1024	128	0.4
Society&Culture	1024	64	0.3

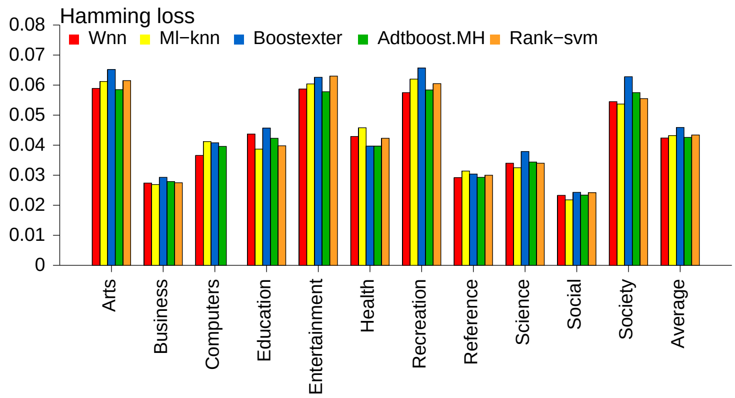


Fig. 2. Experimental results of each multi-label learning algorithm on the Web page data sets in terms of *hamming loss*. The smaller the value of *hamming loss*, the better the performance of the classifier.

In order to optimize a VG-RAM WNN classifier, its parameters, i.e. number of neurons, number of synapses per neuron and threshold τ (Section 3), must be tuned by testing which values yield the best performance. To avoid taking advantage of the test set to optimize the classifier used on each of the 11 problems, we divided the 2000 documents training(-and-validation) set $TV = \{d_1, \ldots, d_{|TV|}\}$ of each problem into a 1500 documents training set $Tr = \{d_1, \ldots, d_{|Tr|}\}$, from which the classifier were inductively built, and a 500 documents validation set $Va = \{d_{|Tr|+1}, \ldots, d_{|TV|}\}$, on which the repeated evaluations of the classifier aimed at parameter optimization were performed. Table 2 shows, for each one of the 11 text classification problems, the parameters that yield the best performance in terms of the five multi-label evaluation metrics adopted (Section 2).

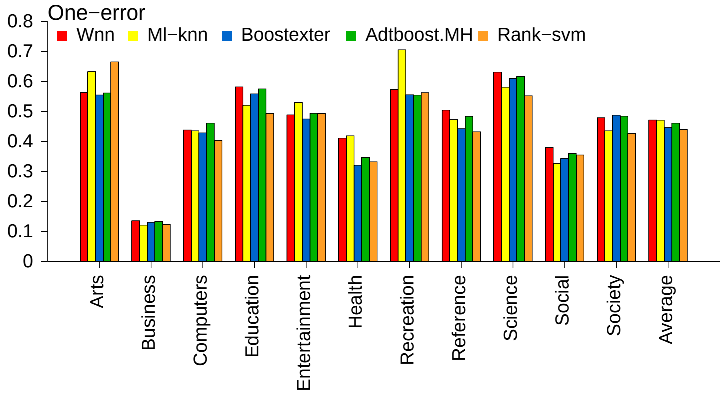


Fig. 3. Experimental results of each multi-label learning algorithm on the Web page data sets in terms of *one-error*. The smaller the value, the better the performance.

Once its parameters are estimated, we can use VG-RAM WNN to predict the set of classes of the test documents. We compared VG-RAM WNN classification performance with that of: the boosting-style algorithm BOOSTEXTER [3], the multi-label kernel method RANK-SVM [5], the multi-label decision tree ADTBOOST.MH [6][2], and the multi-label lazy learning approach ML-KNN [11]. We believe that these classifiers are representative of some of the most effective multi-label text classification methods available.

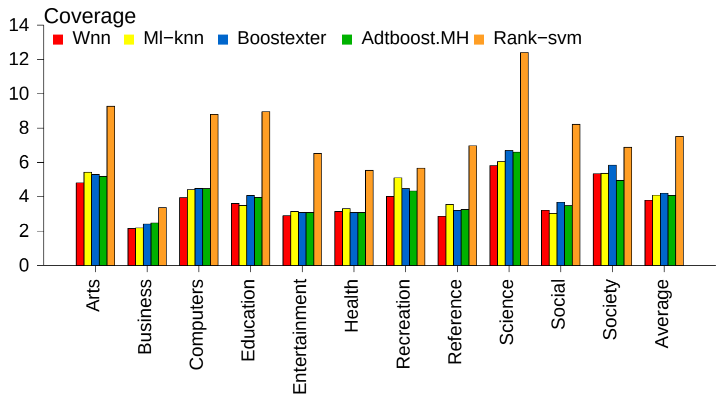


Fig. 4. Experimental results of each multi-label learning algorithm on the Web page data sets in terms of *coverage*. The smaller the value, the better the performance.

[2] Note that ADTBOOST.MH does not provide a ranking of classes; thus, this *ranking loss* metric could not be used.

For BOOSTEXTER and ADTBOOST.MH, the number of boosting rounds was set to be 500 and 50, respectively. For RANK-SVM, polynomial kernels with degree 8 were used. For ML-KNN, k was set to 10 and Euclidean metric was used to measure distances between documents[3]. For each data set, the multi-label algorithms were trained with the 2000 documents in the the training(-and-validation) set and tested with the 3000 documents in the test set.

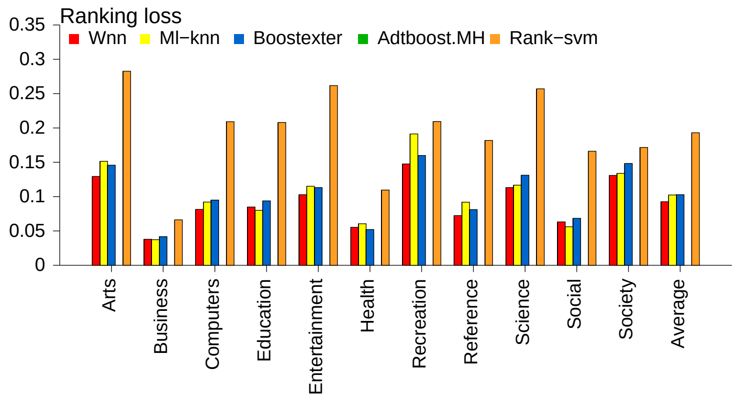


Fig. 5. Experimental results of each multi-label learning algorithm on the Web page data sets in terms of *ranking loss*. The smaller the value, the better the performance.

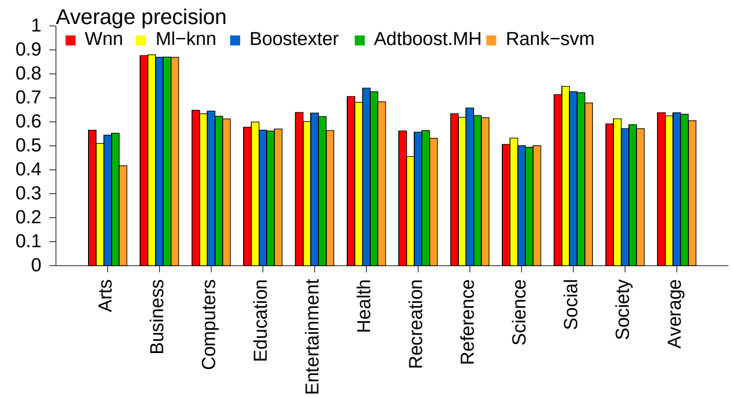


Fig. 6. Experimental results of each multi-label learning algorithm on the Web page data sets in terms of *average precision*. The larger the value, the better the performance.

Figures 2 to 6 show the experimental results of each multi-label classification technique on all the Web page data sets in terms for *hamming loss*, *one-error*, *coverage*, *ranking loss*, and *average precision*, respectively. These plottings also show the averages for each evaluation metric over all data sets. On average, VG-

[3] The results for BOOSTEXTER, RANK-SVM, ADTBOOST.MH, and ML-KNN were obtained from the work presented in [11].

RAM WNN performs better than the other algorithms in terms of *hamming loss*, *coverage*, *ranking loss* and *average precision*, and is far superior than RANK-SVM in terms of *coverage* and *ranking loss*—VG-RAM WNN shows gains of 49.36% in terms of *coverage* (Figure 4) and 52.02% in terms of *ranking loss* (Figure 5), considering RANK-SVM as the baseline. VG-RAM WNN also performs better than ML-KNN in terms of *ranking loss* (Figure 5) and *coverage* (Figure 4), showing gains of 9.57% and 7.21%, respectively. Finally, VG-RAM WNN performs better than ADTBOOST.MH in terms of *hamming loss* (Figure 2) and than BOOSTEXTER in terms of *average precision* (Figure 6), with gains of 0.47% and 0.05%, respectively. VG-RAM WNN shows inferior, although comparable performance in terms of *one-error*; but, it is worthy to note that all classifiers examined here perform poorly in terms of *one-error*—on average, in 45.81% of the documents tested the top-ranked class was not in the set of appropriate classes predicted for the documents (Figure 3).

5 Conclusions and Future Work

In this work, we presented an experimental evaluation of the performance of virtual generalizing random access memory weightless neural networks (VG-RAM WNN [13]) on multi-label text classification. We compared the VG-RAM WNN classification performance with that of the boosting-style algorithm BOOSTEXTER [3], the multi-label kernel method RANK-SVM [5], multi-label decision tree ADTBOOST.MH [6], and multi-label lazy learning techniques ML-KNN [11]. Our experimental results showed that, on average, VG-RAM WNN outperforms the comparing algorithms in terms of *hamming loss*, *coverage*, *ranking loss* and *average precision*, and is far superior to RANK-SVM in terms of *coverage* and *ranking loss*, showing gains of up to 52%. As future work, we intend to evaluate the classification performance of VG-RAM WNN using different multi-label application scenarios, such as image annotation and gene functional prediction. We also intend to use correlated VG-RAM WNN [21] and examine other mechanisms for taking more advantage of the correlation between classes in order to improve classification performance.

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34**(1) (2002) 1–47
2. McCallum, A.: Multi-label text classification with a mixture model trained by EM. In: Working Notes of the AAAI'99 Workshop on Text Learning. (1999)
3. Schapire, R.E., Singer, Y.: BoosTexter: a boosting-based system for text categorization. Machine Learning **39**(2/3) (2000) 135–168
4. Clare, A., King, R.D.: Knowledge discovery in multi-label phenotype data. In: Lecture Notes in Computer Science. Volume 2168. (2001) 42–53
5. Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: Advances in Neural Information Processing Systems. Volume 14., MIT Press (2002) 681–687

6. Comité, F.D., Gilleron, R., Tommasi, M.: Learning multi-label alternating decision tree from texts and data. In: Lecture Notes in Computer Science. Volume 2734., Springer (2003) 35–49
7. Ueda, N., Saito, K.: Parametric mixture models for multi-label text. In: Advances in Neural Information Processing Systems. Volume 15., MIT Press (2003) 721–728
8. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recognition **37**(9) (2004) 1757–1771
9. Kazawa, H., Izumitani, T., Taira, H., Maeda, E.: Maximal margin labeling for multi-topic text categorization. In: Advances in Neural Information Processing Systems. Volume 17., MIT Press (2005) 649–656
10. Zhang, M.L., Zhou, Z.H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transactions on Knowledge and Data Engineering **18**(10) (2006) 1338–1351
11. Zhang, M.L., Zhou, Z.H.: ML-KNN: A lazy learning approach to multi-label learning. Pattern Recognition **40**(7) (2007) 2038–2048
12. Gao, S., Wu, W., Lee, C.H., Chua, T.S.: A MFoM learning approach to robust multiclass multi-label text categorization. In: Proceedings of the 21st International Conference on Machine Learning. (2004) 329–336
13. Ludermir, T.B., Carvalho, A.C.P.L.F., Braga, A.P., Souto, M.D.: Weightless neural models: a review of current and past works. Neural Computing Surveys **2** (1999) 41–61
14. Austin, J., ed.: RAM-Based Neural Networks. World Scientific (1998)
15. Salton, G.: Developments in automatic text retrieval. Science **253** (1991) 974–980
16. Aleksander, I.: Self-adaptive universal logic circuits. IEEE Electronic Letters **2** (1996) 231–232
17. Aleksander, I.: From WISARD to MAGNUS: a Family of Weightless Virtual Neural Machines. In: RAM-Based Neural Networks. World Scientific (1998) 18–30
18. Mitchell, R.J., Bishop, J.M., Box, S.K., Hawker, J.F.: Comparison of Some Methods for Processing Grey Level Data in Weightless Networks. In: RAM-Based Neural Networks. World Scientific (1998) 61–70
19. Dumais, S.T., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representation for text categorization. In: Proceedings of the 7th ACM International Conference on Information and Knowledge Management. (1998) 148–155
20. Komati, K.S., De Souza, A.F.: Using weightless neural networks for vergence control in an artificial vision system. Applied Bionics and Biomechanics **1**(1) (2003) 21–31
21. Carneiro, R., Dias, S.S., Jr., D.F., Oliveira, S., d. Garcez, A.S., De Souza, A.F.: Improving VG-RAM neural networks performance using knowledge correlation. Lecture Notes on Computer Science **4232** (2006) 427–436