Hybrid Parallel Classifiers for Semantic Subspace Learning

Nandita Tripathi (University of Sunderland, U.K.; email: bf39rv@student.sunderland.ac.uk)
Michael Oakes (University of Sunderland, U.K.; email: Michael.Oakes@sunderland.ac.uk)
Stefan Wermter (University of Hamburg, Germany ; email: wermter@informatik.uni-hamburg.de)

Abstract — Subspace learning is very important in today's world of information overload. Distinguishing between categories within a subset of a large data repository such as the web and the ability to do so in real time is critical for a successful search technique. The characteristics of data belonging to different domains are also varying widely. This merits the need for an architecture which caters to the differing characteristics of different data domains. In this paper we present a novel hybrid parallel architecture using different types of classifiers trained on different subspaces. We further compare the performance of our hybrid architecture with a single classifier – full data space learning and show that it outperforms the single classifier system by a large margin when tested with a variety of hybrid combinations. Our results show that subspace classification accuracy is boosted and learning time reduced significantly with this new hybrid architecture.

Keywords: parallel classifiers, hybrid classifiers, subspace learning, significance vectors, maximum significance

1. Introduction

The *curse of dimensionality* [1] degrades the performance of many learning algorithms. Therefore, methods are needed that can discover clusters in various subspaces of high dimensional datasets [2]. Subspace analysis lends itself naturally to the idea of hybrid classifiers. Each subspace can be processed by a classifier best suited to the characteristics of that subspace. Instead of using the complete set of full space dimensions, classifier performances can be boosted by using only a subset of the dimensions. The method of choosing an appropriate reduced set of dimensions is an active research area [3]. Dimensionality reduction via Random Projections [4] has also attracted attention recently. In the Random Subspace Method (RSM) [5], classifiers were trained on randomly chosen subspaces of the original input space and the outputs of the models were then combined. However random selection of features does not guarantee that the selected inputs have necessary discriminant information. Several variations of RSM have been proposed such as Relevant random feature subspaces for co-training (Rel-RASCO) [6], Not-so-Random Subspace Method (NsRSM) [7] and Local Random Subspace Method [8].

The performance of different types of classifiers (Bayesian, Tree based, neural networks, etc) can be improved by combining them with various types of combining rules. Three types of combined classifiers [9] are parallel classifiers, stacked classifiers and weak classifiers combination (e.g. bagging and boosting). Parallel classifiers are frequently, but not necessarily, of the same type while stacked classifiers are mostly of different types. Several researchers have studied classifier combinations with respect to text categorization. In one method [10], text and metadata were considered as independent sources of evidence and the judgements of their corresponding classifiers were combined. Another text categorization method [11] was based on a hierarchical array of neural networks.

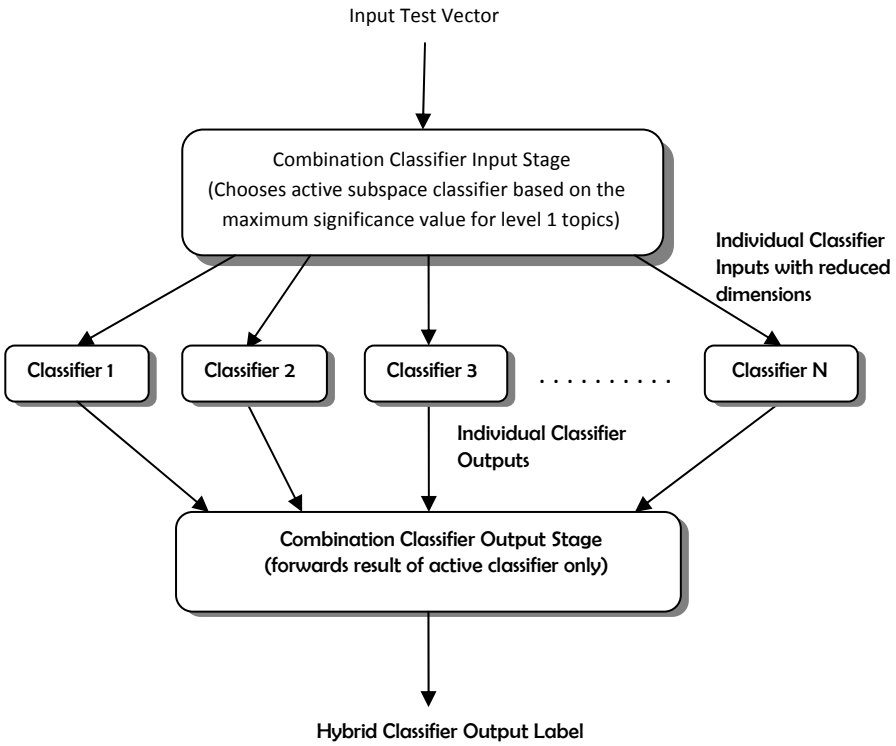
In the real world, documents can be divided into major semantic subspaces with each subspace having its own unique characteristics. The above research does not take into account this semantic division of documents. We present here a novel hybrid parallel architecture (Fig. 1) which takes advantage of the different semantic subspaces existing in the data. We further show that this new hybrid parallel architecture improves subspace classification accuracy as well as significantly reduces training time. In this architecture, we tested various hybrid combinations of classifiers using the conditional

significance vector representation [12] which is a variation of the semantic significance vector [13], [14] to incorporate semantic information in the document vectors. We compare the performance of this hybrid parallel classifier against that of single MLP classifiers using the significance vector as well as the tf-idf vector representation. Our experiments were performed on the Reuters corpus (RCV1) [15] using the first two levels of the topic hierarchy.

2. Methodology Overview and Overall Architecture

We used the Reuters Corpus as it is a well-known test bench for text categorization experiments. It also has a hierarchical organization with four major groups which is well suited to test our hybrid architecture. We used the Reuters Corpus headlines for our experiments as they provide a concise summary of each news article. Each Reuters headline consists of one line of text with about 8 – 10 words. Ten thousand headlines along with their topic codes were extracted from the Reuters Corpus. These headlines were chosen so that there was no overlap at the first level categorization. Each headline belonged to only one level 1 category. At the second level, since most headlines had multiple level 2 subtopic categorizations, the first subtopic was taken as the assigned subtopic. Thus each headline had two labels associated with it – the main topic (Level 1) label and the subtopic (Level 2) label. Headlines were then preprocessed to separate hyphenated words. Dictionaries with term frequencies were generated based on the TMG toolbox [16]. These were then used to generate the Full Significance Vector [12], the Conditional Significance Vector [12] and the tf-idf [17] representation for each document. The datasets were then randomised and divided into a training set of 9000 documents and a test set of 1000 documents.

Fig 1. Hybrid Parallel Classifier Architecture for Subspace Learning



The WEKA machine learning workbench [18] was used to examine this architecture with various learning algorithms. Seven algorithms were compared for our representations to examine the different categories of classification algorithms. Classification Accuracy, which is a comparison of the predicted class to the actual class, and the Training Time were recorded for each experiment run.

3. Steps for Data Processing and Data Generation for Experiments

3.1 Text Data Preprocessing

Ten thousand Reuters headlines were used in these experiments. The Level 1 categorization of the Reuters Corpus divides the data into four main topics. These main topics and their distribution in the data along with the number of subtopics of each main topic in this data set are given in Table 1.

Table 1: Reuters Level 1 Topics

No	Main Topic	Description	Number Present	No of Sub-topics
1	CCAT	Corporate/ Industrial	4600	18
2	ECAT	Economics	900	8
3	GCAT	Government/ Social	1900	20
4	MCAT	Markets	2600	4

Table 2: Some Reuters Level 2 subtopics used for our experiments.

Main Topic	Subtopic	Description	Num-ber
CCAT	C17	Funding/ Capital	377
ECAT	E12	Monetary/ Economic	107
GCAT	G15	European Community	38
MCAT	M14	Commodity Markets	1050

Level 2 categorization further divides these into subtopics. Here we took the direct (first level nesting) subtopics of each main topic occurring in the 10,000 headlines. A total of 50 subtopics were included in these experiments. Some of these subtopics with their numbers present are shown in Table 2. Since all the headlines had multiple subtopic assignment e.g. C11/C15/C18, only the first subtopic e.g. C11 was taken as the assigned subtopic.

3.2 Semantic Significance Vector Generation

We use a vector representation which looks at the significance of the data and weighs different words according to their significance for different topics. References [13] and [14] have introduced the concept of semantic significance vectors. Significance Vectors are determined based on the frequency of a word in different semantic categories. A modification of the significance vector called the semantic vector uses normalized frequencies. Each word *w* is represented with a vector $(c_1,c_2,..,c_n)$ where c_i represents a certain semantic category and *n* is the total number of categories. A value $v(w, c_i)$ is calculated for each element of the semantic vector as the normalized frequency of occurrences of word *w* in semantic category c_i (the normalized category frequency), divided by the normalized frequency of occurrences of the word *w* in the corpus (the normalized corpus frequency):

$$v(w, c_i) = \frac{\text{Normalised Frequency of w in } c_i}{\sum_k \text{Normalised Frequency of w in } c_k}$$

where $k \in \{1..n\}$

For each document, the document semantic vector is obtained by summing the semantic vectors for each word in the document and dividing by the total number of words in the document. This is the version of the semantic significance vector used in our experiments. The TMG Toolbox [16] was used to generate the term frequencies for each word in each headline. The word vector consisted of 54 columns for 4 main topics and 50 subtopics. While calculating the significance vector entries for each word, its occurrence in all subtopics of all main topics was taken into account - hence called *Full Significance Vector*. We also generate the *Conditional Significance Vector* [12] where a word's occurrence in all subtopics *of only a particular main topic* is taken into account while calculating the word significance vector entries.

3.3 Data Vector Sets Generation

As will be described below, three different vector representations (Full Significance Vector, Conditional Significance Vector and tf-idf) were generated for our data. The Conditional Significance Vectors were processed further to generate four main category-wise data vector sets.

3.3.1 Full Significance Vector

Here, the document vectors were generated by summing the full significance word vectors for each word occurring in a document and then dividing by the total number of words in that document. For each Reuters Full Significance document vector the first four columns, representing four main topics – CCAT, ECAT, GCAT & MCAT, were ignored leaving a vector with 50 columns representing 50 subtopics. The order of the data vectors was then randomised and divided into two sets – training set of 9000 vectors and a test set of 1000 vectors.

3.3.2 Category-based Conditional Significance Vectors

Here, the conditional significance word vectors were used to generate the document vectors in the same way as above for the Reuters Corpus. The order of the Reuters Conditional Significance document vectors was randomised and divided into two sets – a training set of 9000 vectors and a test set of 1000 vectors. The training set was then divided into 4 sets according to the main topic labels. For each of these for sets, only the relevant subtopic vector entries (e.g. C11, C12, etc for CCAT; E11, E12, etc for ECAT; etc) for each main topic were retained. Thus the CCAT category training dataset had 18 columns for 18 subtopics of CCAT. Similarly ECAT training dataset had 8 columns, GCAT training dataset had 20 columns and the MCAT training dataset had 4 columns. These 4 training sets were then used to train the 4 parallel classifiers of the Reuters hybrid classifier. The main category of a test data vector was determined by the maximum significance vector entry for the first four columns representing the four main categories. After this, the entries corresponding to the subtopics of this predicted main topic were extracted along with the *actual* subtopic label and given to the classifier trained for this predicted main category.

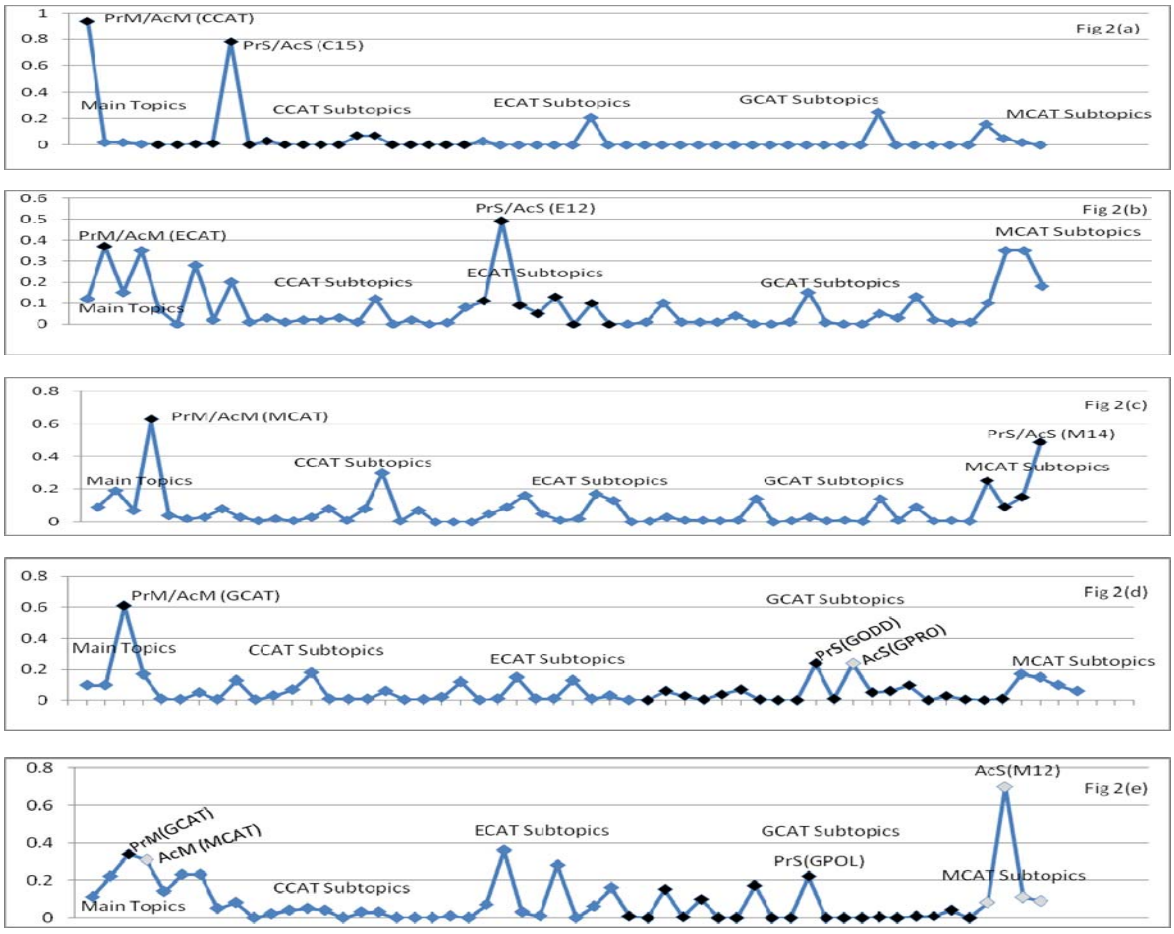
Fig 2 shows the classification decisions for some input vectors. Figures 2(a), 2(b) and 2(c) show correctly classified vectors. In Fig 2(d), the main topic predicted was correct and the vector was presented to the correct classifier but the subtopic classification was wrong. In Fig 2(e), the main topic predicted was wrong and hence the vector was presented to the wrong classifier – resulting in a wrong classification. Fig 2(e) presents an inherent limitation of this system whereby a wrong classifier is chosen by the classifier selection step of the parallel classifier. The accuracy of choosing the correct main topic by selecting the maximum significance level 1 entry has been measured to be 96.80% for the 1000 test vectors i.e. 968 vectors were assigned the correct trained classifiers whereas 3.20% or 32 vectors were assigned to a wrong classifier – resulting in a wrong classification decision for all these 32 vectors. Hence the upper limit for classification accuracy is 96.80% for our hybrid parallel classifier.

3.3.3 TF-IDF Vector generation

The tf-idf weight (Term Frequency–Inverse Document Frequency) is often used in text mining and information retrieval. It is a statistical measure which evaluates how important a word is to a document

in a data set. This importance increases with the number of times a word appears in the document but is reduced by the frequency of the word in the data set. Words which occur in almost all documents have very little discriminative power and hence are given very low weight. The TMG toolbox [16] was used to generate tf-idf vectors for the ten thousand Reuters headlines used in these experiments. The tf-idf vector dataset was then randomized and divided into a training set of 9000 vectors and a test set of 1000 vectors.

Fig 2. Classification Decisions by a Hybrid Parallel Classifier for Some Input Vectors

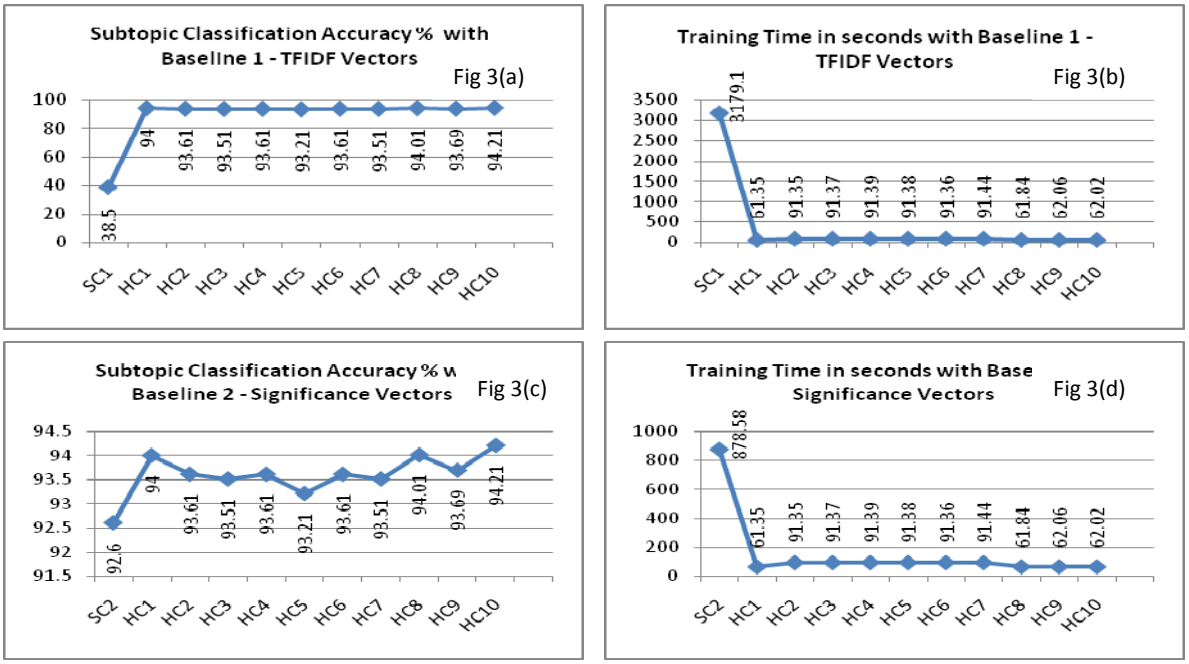


PrM – Predicted Main Topic; AcM – Actual Main Topic; PrS – Predicted Subtopic; AcS – Actual Subtopic
Black data points show the predicted main topic and its associated subtopics.
Light Gray data points show the actual main topic and its associated subtopics

3.4 Classification Algorithms

Seven Classification algorithms were tested with our datasets namely Random Forest, C4.5, Multilayer Perceptron, Naïve Bayes, BayesNet, NNge and PART. Random Forests [19] are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently. C4.5 [20] is an inductive tree algorithm with two pruning methods: subtree replacement and subtree raising. Multilayer Perceptron [21] is a neural network which uses backpropagation for training. Naive Bayes [22] is the simplest form of Bayesian network, in which all attributes are independent given the value of the class variable. BayesNet [23] implements Bayes Network learning using various search algorithms and quality measures. NNge [24] is a nearest neighbor - like algorithm using non-nested generalized exemplars. A PART [25] decision list uses separate-and-conquer. It builds a partial C4.5 decision tree in each iteration and makes the best leaf into a rule.

Fig 3. Hybrid Parallel Classifiers Performance Metrics



Classifier Index

SC1- Single MLP over full data using tf-idf Vectors; SC2- Single MLP over full data using Significance Vectors

HC1- Hybrid Parallel Classifier with category-wise classifiers chosen from training data performance

Hybrid 2-Classifier Systems :

HC2 (MLP/NB)*, HC3 (MLP/BNet)*, HC4 (MLP/NNge)*, HC5 (MLP/PART)*, HC6 (MLP/J48)*, HC7 (MLP/RF)*

Hybrid 4-Classifier Systems :

HC8 (MLP/NB/ NNge/ J48)*, HC9 (MLP/BNet/PART/RF)*, HC10 (MLP/NNge/PART/NB)*

*MLP - Multilayer Perceptron (Neural Network); NB - Naïve Bayes, BNet - BayesNet (Bayesian);
NNge - Nearest Neighbour with Generalised Exemplars, PART - PART Decision List (Rule Based);
J48 - Weka's version of C4.5, RF - Random Forest (Tree Based);

4. Results and their Analysis

Experiments were run using seven algorithms from Weka on the Reuters Corpus. The corpus was divided into 9000 training vectors and 1000 test vectors. For the hybrid classifier, the 9000 training vectors were divided according to the actual main categories and were used to train the chosen category classifier with the relevant subtopic vector entries and actual subtopic labels. The test vectors were divided into main categories based on the maximum significance value among the main topic significance vector entries. The relevant subtopic vector entries of this predicted main topic and the *actual* subtopic labels of these vectors were used to test these classifiers.

In the first step, the algorithms were run using category-wise separated data from the training set to select the algorithm with the highest classification accuracy for each main category. In case of a tie between two algorithms, the one with the lower training time was chosen. Subsequently these selected algorithms were applied to the test data and the performance of the hybrid classifier was measured. The category-wise separated Conditional Significance Vectors were used here. Each of the algorithms was also run on the full (not category-wise separated) data set to provide a comparison for the hybrid classifier. Two vector representations were used for the comparison baseline – the Full Significance Vector and tf-idf. As the performance of many classifiers for each main category was quite close to each other, we also ran some experiments using a predefined set of classifiers. The combination of MLP with

different types of classifiers (Bayesian, rule-based and tree-based classifiers) was evaluated and the best combination was identified. For a two-classifier combination, MLP and the other classifier were used alternately on the main category topics while for a four-classifier system four different classifiers were used on the four main topics.

The charts in Fig 3 show a comparison of the performance of hybrid classifiers with that of MLP. The subtopic classification accuracy percentage and training time in seconds is shown for the Hybrid Parallel classifiers along with that of the baselines. The baseline is a single MLP classifier using full data (not category-wise separated data). This baseline experiment is run with two different vector representations - Significance Vector and tf-idf. The accuracies of all the hybrid parallel classifiers are better than that of the single MLP classifier. The Hybrid 4-classifier system (HC10) shows the best result which is quite similar to that of the hybrid classifier with category-wise classifiers chosen from training set (HC1).

Overall, it was observed that there was an improvement in subtopic classification accuracy along with a significant reduction in training time. The classification accuracies of all the hybrid classifiers were quite close to each other but all of them were much better than the classification accuracy of the single classifier with tf-idf baseline. The difference with the significance vector baseline was smaller but even there the classification accuracies of the hybrid systems were better. The training times showed a very steep reduction compared to both baselines.

The training times of all hybrid classifiers were quite close to each other with HC1, HC8, HC9 and HC10 showing the least training time. The other hybrid classifiers were two-classifier systems with one MLP and one non-MLP classifier alternating on the main topics. Hence for the Reuters data with four main topics, there were two MLPs in all the hybrid 2-classifier systems. This could account for a slightly higher training time of these classifiers versus the hybrid 4-classifier systems (HC8, HC9 and HC10) which have only one MLP in the combination. The hybrid classifier with category-wise classifiers chosen from training set (HC1) had MLP for the CCAT main topic and C4.5 for all other main topics. Since this combination also had only one MLP, its training time was comparable to the hybrid 4-classifier systems. The average of 10 runs was taken for each experiment. In the hybrid classifier, even though we are using more classifiers, the training time is reduced. This is because each classifier now trains on a reduced set of data with a reduced set of vector components. This two-fold reduction translates to a significance decrease in training time.

4. Conclusion

In this work, we attempt to leverage the differences in the characteristics of different subspaces to improve semantic subspace learning. The main objective here is to improve document classification in a vast document space by combining various learning methods. Our experiments show that hybrid parallel combinations of classifiers trained on different subspaces offer a significant performance improvement over single classifier learning on full data space. Individual classifiers also perform better when presented with less data in lower dimensions. Our experiments also show that learning based on the semantic separation of data space is more efficient than full data space learning. Combining different types of classifiers has the advantage of integrating characteristics of different subspaces and hence improves classification performance. This technique can work well in other domains like pattern / image recognition where different classifiers can work on different parts of the image to improve overall recognition. Computational Biology too can benefit from this method to improve recognition within sub-domains.

In our experiments, subspace detection is done by processing a single document vector. This method is independent of the total number of data samples and only compares the level 1 topic entries. The time complexity of the combining classifier is thus O(k) where k is the number of level 1 topics. The novelty of our approach is in the use of a maximum significance based method of input vector projection for a hybrid parallel classifier. Combining MLP in parallel with a basic classifier (Bayesian, tree based or rule based) improves the classification accuracy and significantly reduces the training time. The experiments also show that using maximum significance value is very effective in detecting the relevant subspace of a test vector.

References

[1] Friedman J.H., 1997, “On Bias, Variance, 0/1—Loss, and the Curse-of- Dimensionality,” In Data Mining and Knowledge Discovery, Volume 1, Issue 1, 1997, pp 55 – 77

[2] Parsons L., Haque E. & Liu H., 2004, “Subspace Clustering for High Dimensional Data : A Review,” In ACM SIGKDD Explorations Newsletter**,** Vol 6, Issue 1, 2004, pp 90 – 105

[3] Varshney K.R. & Willsky A.S., 2009, “Learning dimensionality-reduced classifiers for information fusion,” In Proceedings of the 12th International Conference on Information Fusion, pages 1881–1888, Seattle, Washington, July 2009.

[4] Fradkin, D. & Madigan, D., 2003, “Experiments with Random Projections for Machine Learning,” In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003, pp 517-522

[5] Ho, Tin Kam, 1998. "The random subspace method for constructing decision forests", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 20, Issue 8 (Aug 1998), pp 832-844

[6] Yaslan Y. & Cataltepe Z., 2010, "Co-training with relevant random subspaces", Neurocomputing 73 (2010) pp 1652-1661 (Elsevier)

[7] Garcia-Pedrajas N., Ortiz-Boyer D., 2008, "Boosting Random Subspace Method", Neural Networks 21 (2008), pp 1344-1362

[8] Kotsiantis S.B., 2009, "Local Random Subspace Method for constructing multiple decision stumps", International Conference on Information and Financial Engineering, pp 125-129, 2009

[9] Duin, R.P.W. & Tax, D.M.J., 2000, "Experiments with Classifier Combining Rules", J.Kittler and F. Roli (Eds.): MCS 2000, LNCS 1857, pp. 16−29, 2000

[10] Al-Kohafi et al, 2001, "Combining multiple classifiers for text categorization"**,** Proceedings of the tenth international conference on Information and knowledge management , CIKM 2001**,** pp 97-104

[11] Ruiz M.G. & Srinivasan P., 1999, "Hierarchical Neural Networks for Text Categorization", SIGIR 1999

[12] Tripathi N., Wermter S., Hung C. & Oakes M., 2010, "Semantic Subspace Learning with Conditional Significance Vectors", Proceedings of the IEEE International Joint Conference on Neural Networks, pp 3670-3677, Barcelona, July 2010

[13] Wermter S., Panchev C. & Arevian G., “Hybrid Neural Plausibility Networks for News Agents,” In Proceedings of the Sixteenth National Conference on Artificial Intelligence, 1999, pp 93-98

[14] Wermter S., “Hybrid Connectionist Natural Language Processing,” Chapman and Hall. 1995

[15] Rose T., Stevenson M., and Whitehead M., “The Reuters Corpus Volume 1 - from Yesterday’s News to Tomorrow’s Language Resources,” In Proceedings of the Third International Conference on Language Resources and Evaluation (LREC- 02),2002, pp 827–833.

[16] Zeimpekis D. and Gallopoulos E., “TMG : A MATLAB Toolbox for Generating Term Document Matrices from Text Collections, ” Book Chapter in Grouping Multidimensional Data: Recent Advances in Clustering, J. Kogan and C. Nicholas, eds., Springer, 2005

[17] Manning C., Raghavan P. & Schutze H., 2008, “Introduction to Information Retrieval,” Cambridge University Press. 2008

[18] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P. and Witten I., “The WEKA Data Mining Software: An Update,” In ACM SIGKDD Explorations Newsletter, Volume 11, Issue 1, July 2009,pp 10-18.

[19] Breiman L., “Random Forests,” In Machine Learning 45(1), Oct. 2001, pp 5-32

[20] Quinlan J.R., “C4.5 : Programs for Machine Learning,” Morgan Kaufmann Publishers, San Mateo, CA. 1993

[21] Verma B., “Fast training of multilayer perceptrons,” In IEEE Transactions on Neural Networks, Vol 8, Issue 6, Nov 1997 pp 1314-1320.

[22] Zhang H., 2004, "The optimality of Naïve Bayes", American Association for Artificial Intelligence (www.aaai.org)

[23] Pernkopf F., 2007, “Discriminative learning of Bayesian network classifiers, ” In Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications, 2007, pp 422-427

[24] Martin B., 1995, “Instance-Based learning : Nearest Neighbor With Generalization,” Master Thesis, University of Waikato, Hamilton, New Zealand, 1995

[25] Frank E. and Witten I.H., 1998, “Generating Accurate Rule Sets Without Global Optimization,” In Shavlik, J., ed., Machine Learning: Proceedings of the Fifteenth International Conference, Morgan Kaufmann Publishers, 1998