Genetic Algorithm Based Text Categorization Using OLEX Method

S.Manjula
Dept of Software Engineering,
Periyar Maniammai University,Vallam,Thanjavur,Tamilnadu

Abstract-**The system describes new similarity-based genetic algorithm (*GA*) and thresholding Strategies (*R&SCut* variants). *GA* was designed to give appropriate weights to terms according to their semantic content and importance by using their co-occurrence information and the discriminating power values for similarity computation. After investigating the existing common thresholding strategies, design multi-class text categorization in which documents may belong to variable numbers of categories.The proposed System conducted extensive comparative experiments on two standard text collections (the Reuters-21578 and the 20-Newsgroups). The experimental results using a standard evaluation method, *F*1, for micro and macro-averaged performance. The results show that *GA* and *R&SCut* variants work better than other widely used techniques.**

Keywords— Genetic algorithm, Olex method, Classification, Text categorization

I. INTRODUCTION

Text Categorization (or Classification) is the task of assigning natural language texts to one or more thematic categories on the basis of their contents. A number of machine learning methods have been proposed in the last few years, including k-NN, Probabilistic Bayesian, Neural Networks and SVMs. In a different line, rule learning algorithms, such as Ripper and C4.5, have become a successful strategy for classifier induction. Rule-based classifiers provide the desirable property of being readable and, thus, easy for people to understand (and, possibly, modify).

Genetic Algorithms (GA's) are stochastic search methods inspired by biological evolution. Their capability to provide good solutions for classical optimization tasks has been demonstrated by various applications, including TSP and Knapsack. Rule induction is also one of the application fields of GA's. The basic idea is that each individual encodes either a classification rule or a classifier, and that its fitness is expressed in terms of predictive accuracy.

The problematic indexing feature space dimensionality reduction has been tackled by a two-level supervised scheme, implemented by a noisy terms filtering and a subsequent redundant terms compression. Gaussian probabilistic categorizers' concomitance of sparsely in ATC. have been revisited and adapted to the concomitance of sparsely in ATC.

The proposed greedy heuristics allows to efficiently inducing accurate and reliable classifiers. The restrictions on the d-terms generated by the proposed heuristics may actually limit rule effectiveness. A term can appear in at most one conjunctive term, so that the literals occurring in the rules of a classifier cannot share common terms.

II SCOPE OF THE PROJECT

The simplicity of the hypothesis language, in Olex-Genetic Algorithm an individual represents a candidate classifier (instead of a single rule).

The fitness of an individual is expressed in terms of the F-measure attained by the corresponding classifier when applied to the training set.

This several-rules-per-individual approach (as opposed to the single-rule-per-individual approach) provides the advantage that the fitness of an individual reliably indicates, quality and a measure of the predictive accuracy of the encoded classifier rather than of a single rule. Once the population of individuals has been suitably initialized, evolution takes place by iterating elitism, selection, crossover and mutation, until a predefined number of generations is created. Unlike other rule-based systems (such as Ripper) or decision tree systems (like C4.5) the proposed method is a one-step learning algorithm which does not need any post -induction optimization to refine the induced rule set. This is clearly a notable advantage, as rule set-refinement algorithms are rather complex and time - consuming tasks.

The experimentation over the standard test sets Reuters-21578 and Ohsumed confirms the goodness of the proposed approach: on both data collections, Olex-GA showed to be highly competitive with some of the top-performing learning algorithms for text categorization, notably, Naive Bayes, C4.5, Ripper and SVM (both polynomial and rbf). Furthermore, it consistently defeated the greedy approach to problem MAX-F we reported in [19]. In addition, Olex-GA turned out to be an efficient rule induction method faster than both C4.5 and Ripper.

III. EXISTING SYSTEM

One point that is noteworthy is the relationship between Olex -Greedy and Olex- GA, in terms of both predictive accuracy and time efficiency. Olex-GA consistently beats Olex- Greedy on both data sets. Effectiveness of GA's in rule induction is a consequence of their inherent ability to cope with attribute interaction as, thanks to their global search approach, more attributes at a time are modified and evaluated as a whole. This is in contrast with the local, one-condition-at-a-time greedy rule generation approach. On the other search strategy which straight leads to a suboptimal hand, concerning time efficiency, Olex-Greedy showed to be much faster than Olex-GA. This should not be surprising, as the greedy approach, unlike GA's, provides a solution.

IV. Problem of Existing System

Text categorization (TC) is to assign a pre-defined category to natural language texts based on its content. TC used to be done by human experts to develop handcraft classification rules. This is very time consuming and as the amount of documents increases, it becomes infeasible.Automatic classification is proposed consequently. This is mainly done by using machine learning approaches which are to solve supervised learning tasks. To solve the problem of TC, there are some unique challenges.

First, no matter what representation method is used, due to the vast amount of natural language words, a TC problem must have a great amount of features. **Second**, due to the amount of features and flexibility of representation of documents, preprocessing is very important to TC. The quality of preprocessing can have a big impact on the performance of building classifiers and the final generalization ability of the model. **Third**, because TC has a very wide application range, incorporating domain specified knowledge into different stages of solving the problem can play an important role.

One is to use multi-class classifiers such as decision trees or do binary classification on a document on every category using a binary-class classifier. Single layer classification vs. hierarchical structure categorization: In some applications such as web-page categorization, documents need to be assigned a sub -category under an upper level category. This can be accomplished by building a general model for main categories first and then different models for sub-categories.

V. System Design

Texts cannot be interpreted by a classifier or a classifier building algorithm directly. The transforming process is usually called document indexing. As mentioned above, there are many ways to index a single document. However, different choices of indexing methods (representation) can have great impact on the building time and the generalizing ability of the model. Usually a document is represented by a vector of terms. In this case if 'term' means 'word', the representation is to regard each different word as a separate feature and whether a document process a word decided the value of the corresponding feature of that document. This value may be binary to indicate the appearance of the word in the document or an integer telling how many times the word appeared. This is often called 'bag of words' representation approach. Of course 'term' can mean something else more sophisticated than single words, for example 'phrase'. However, in a number of experiments revealed that these methods do not yield significantly better results. Lewis argued that the reason for this is while 'phrase' indexing has superior semantic quality, it loses statistical information about the texts which word indexing contains.

Conditions	Suffix	Replacement	Examples
(m>0)	eed	ee	feed.>feed
			agreed.>agree
(*v*)	ed	NULL	plastered.>plaster
			bled.>bled
(*v*)	ing	NULL	motoring.>motor
			sing.>sing

Table 1. Stemming Rules

While conventional classifiers usually disregard the context of the words, they take that into account. How the presence or absence of a word contributes to the classification of that document depends on the context of the word.

The concept of context can be ambiguous and involves different criterion. But despite of this, these two algorithms present extremely good performance in many TC problems They are also good candidates in certain tasks.

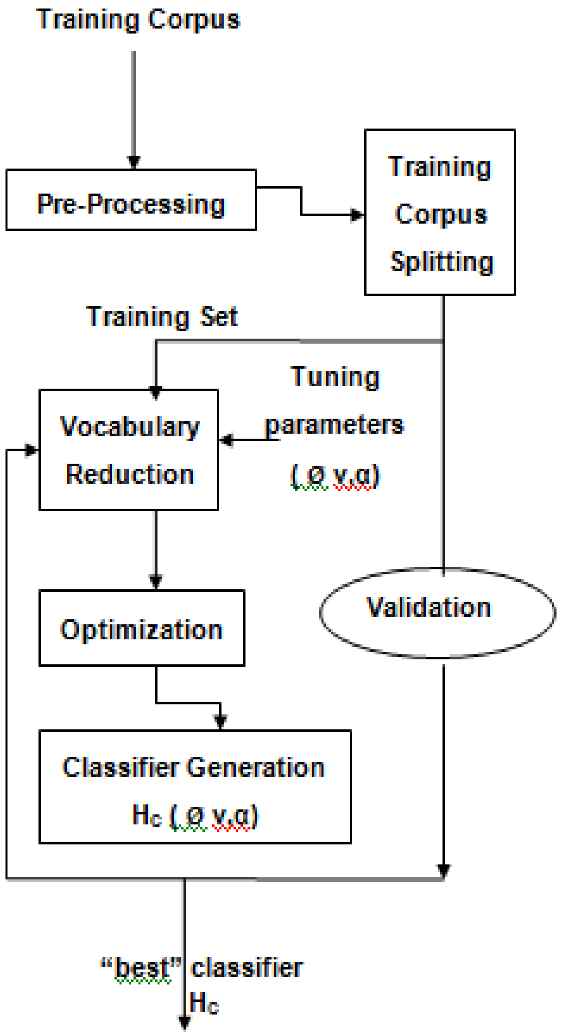

Fig. 1. System Architecture

VI. CONCLUSION

The experimental study described in the previous sections shows that Olex can induce classifiers that are both accurate and compact. It also indicates that the induction process is efficient. Interestingly, all these properties have consistently been observed on all training data, on which Olex showed a uniform behavior. Given the very different application domains the corpora refer to, this is a clear proof of robustness. The observed behavior is general, not corpus-dependent. Quite obviously, accuracy is consequence of a powerful hypothesis language, while compactness and efficiency result from an effective optimization heuristics which infers few, high-quality discriminating terms. Efficiency also stems from the simplicity of the induction model (one-step process). Next, we address some of the advantages and issues of the proposed approach, and relate it to other learning algorithms.

VII. FUTURE ENHANCEMENT

The experimental results obtained on the standard data collections Reuters- 21578 and Ohsumed show that Olex-NN quickly converges to very accurate classifiers. In particular, in the case of Ohsumed , it defeats all the other evaluated algorithms. Further, on both data sets, Olex-NN consistently beats Olex-Greedy. As for time efficiency, Olex-NN is slower than Olex -Greedy but faster than the other rule learning methods (i.e., Ripper and C4.5). The experiments reported in this paper somewhat preliminary, and feel that performance can further be improved through a fine-tuning of the GA parameters.

ACKNOWLEDGEMENT

I would like to thank the management of Periyar Maniammai University and my family members.

REFERENCES

[1] Agresti A. (2002) 'Categorical Data Analysis', Wiley-Interscience.

[2] Anthony M. and Biggs N. (1992) 'Computational Learning Theory' Cambridge Univ. Press.

[3] Antonie M. and Zaiane O. (2004) 'An Associative Classifier Based on Positive and Negative Rules,' Proc. Ninth ACM SIGMOD Workshop Research Issues in Data Mining and Knowledge Discovery (DMKD),.

[4] Apte C. Damerau F.J. and Weiss S.M. (1994) 'Automated Learning of Decision Rules for Text Categorization' ACM Trans. Information Systems, Vol. 12, no. 3, pp. 233-251.

[5] Baralis E. and Garza P. (2006) 'Associative Text Categorization Exploiting Negated Words' Proc. 21st Ann. ACM Symp. Applied Computing (SAC '06), pp. 530-535.

[6] Caropreso M.F., Matwin S. and Sebastiani F. (2001) 'A Learner - Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization' Text Databases and Document Management: Theory and Practice, A.G. Chin, ed., pp. 78-102.

[7] Cohen W.W. (1995) 'Text Categorization and Relational Learning' Proc. 12th Int'l Conf. Machine Learning (ICML).

[8] Cohen ,W.W. and Page, C.D. (1995) 'Polynomial Learnability and Inductive Logic Programming: Methods and Results' New Generation omputing, vol. 13, no. 34, pp. 369-409.

[9] Cohen W.W. and Singer Y. (1999) 'Context-Sensitive Learning Methods for Text Categorization' ACM Trans. Information Systems, vol. 17, no. 2, pp. 141-173.

[10] Debole F. and Sebastiani F. (2004) 'An Analysis of the Relative Difficulty of Reuters-21578 Subsets' Proc. Fourth Int'l Conf. Language Resources and Evaluation (LREC '04).

[11] Dzeroski S. , Muggleton S. and Russell S.J. (1992) 'PAC-Learnability of Determinate Logic Programs' Proc. Fifth Ann. ACM Workshop Computational Learning Theory (COLT)

[12] Forman G. (2003) 'An Extensive Empirical Study of Feature Selection Metrics for Text Classification' J. Machine Learning Research, vol. 3, pp. 1289-1305.

[13] Gottlob G., Leone N. and Scarcello F. (1997) 'On the Complexity of Some Inductive Logic Programming Problems' Proc. Seventh Int'l Workshop Inductive Logic Programming (ILP '97), pp. 17-32.

[14] Hersh W., Buckley C., Leone T. and Hickman, D.(1994) 'OHSUMED: An Interactive Retrieval Evaluation and New Large Text Collection for Research' Proc. 17th ACM Int'l Conf. Research and Development in Information Retrieval (SIGIR '94), W.B. Croft and C.J. van Rijsbergen, eds., pp. 192-201.

[15] Japkowicz ,N. and Stephen, S. (2002) 'The Class Imbalance Problem: A Systematic Study' Intelligent Data Analysis J., vol. 6, no. 5, pp. 429-449.

[16] Joachims, T. (1998) 'Text Categorization with Support Vector Machines: Learning with Many Relevant Features' Proc. 10th European Conf. Machine Learning (ECML '98), C. Ne´dellec and C. Rouveirol, eds., pp. 137-142.

[17] Johnson, D.E., Oles, F.J. , Zhang, T.and Goetz, T. (2002) 'A Decision-Tree- Based Symbolic Rule Induction System for Text Categorization' IBM Systems J., vol. 41, no. 3, pp. 428-437.

[18] Kietz, J.U. (1993) Some Lower Bounds for the Computational Complexity of Inductive Logic Programming' Proc. Sixth European Conf. Machine Learning (ECML '93), vol. 667, pp. 115-123.

[19] Kietz J.U. and zeroski, S. D (1994) 'Inductive Logic Programming and Learnability' SIGART Bull., vol. 5, no. 1, pp. 22-32.

[20] Kloesgen W. (1996) 'Explora: A Multipattern and Multistrategy Discovery Assistant' Advances in Knowledge Discovery and Data Mining, pp. 249-271.

[21] Lewis, D.D. (1997) "Reuters-21578 Text Categorization Test Collection," Distribution 1.0, http://metaxa.net/,.

[22] Lewis, D.D. and Hayes, P.J. (1994) 'Guest Editors' Introduction to the Special Issue on Text Categorization' ACM Trans. Information Systems, vol. 12, no. 3, p. 231.