SVM Categorizer: A Generic Categorization Tool Using Support Vector Machines

Elias Kapoutsis
Department of Computation, UMIST
Manchester, UK

Babis Theodoulidis
Department of Computation
UMIST
Manchester, UK

Mohammad Saraee
School of Computing, Science and Eng.
University of Salford,
Manchester. UK

Abstract

Supervised text categorisation is a significant tool considering the vast amount of structured, unstructured, or semi-structured texts that are available from internal or external enterprise resources. The goal of supervised text categorisation is to assign text documents to finite pre-specified categories in order to extract and automatically organise information coming from these resources. This paper proposes the implementation of a generic application – SVM Categorizer using the Support Vector Machines algorithm with an innovative statistical adjustment that improves its performance. The algorithm is able to learn from a pre-categorised document corpus and it is tested on another uncategorized one based on a business intelligence case study. This paper discusses the requirements, design and implementation and describes every aspect of the application that will be developed. The final output of the SVM Categorizer is evaluated using commonly accepted metrics so as to measure its performance and contrast it with other classification tools.

Keywords: Support Vector Machine, text categorisation,

1. Introduction

The task of classifying natural language documents into pre-specified categories has become one of the most significant methods for organising on-line information. This task is commonly referred as “text categorisation”. Its applicability ranges to a variety of applications, from categorising Web pages, to indexing news items from various Internet sources. In the past, this task was entrusted to human indexers, who manually assigned the information gathered from various sources into pre-specified categories. The first approach followed to eliminate this deficiency is the knowledge engineering approach. Domain experts manually create classification rules that classify new documents automatically. Still, the construction of hand-made rules is difficult and complex, as well as time consuming. Therefore, this approach proves to be inefficient and impractical. For example, creating and maintaining such rules for a large number of categories will not only create a chaos in terms of information usage, but it can also lead to mistakes, deteriorating the organisational performance.

Furthermore, users demand an immediate access in a friendly navigation environment so as to access the information available. However, the process of manually handling an accurate categorisation of the incoming documents is time-consuming and, therefore, unacceptable. As a consequence, the document classification step is often neglected in favour of speed. In this case, the users or the domain experts do not have the adequate time to review all the available resources; hence they are forced to accept only the limited pieces that they can easily retrieve. The problem therefore is that crucial information is often overlooked, and opportunities for a competitive advantage are lost. A machine learning approach to constructing text classification rules can overcome all these deficiencies. Given a relatively small number of manually pre-classified training documents, the problem of learning text classification rules can be cast as a supervised learning problem [4]. A general inductive process automatically builds rules by learning the characteristics of the categories from a set of pre-defined labelled documents.

The organization of this paper is as follow. In section 2 the architecture of SVM Categorizer is described. Section 3 presents the main interactions made by the participant objects of the SVM algorithm. In section 4, we evaluate the performance of the SVM Categorizer according to some pre-defined performance measures, such as precision, recall, F_1-measure, and accuracy. Section 5 concludes the paper and highlights some future research in this area. Section 2 and most part of section 3 and 4 have been omitted for brevity. Full detail may be obtained on request

2 SVM Categorizer Architecture

In the last few years, there has been a significant increase of interest concerning SVM. SVMs have empirically shown that their use offers good performance on a variety of problems, such as handwritten character recognition, face detection, and text categorisation [5]. This section endeavours to focuses on the development of a generic classification tool, called SVM Categorizer that can be extended and maintained in the future. In contrast to the Use Case view, the Design view looks inside the system. It describes both the static structure (packages, classes, objects, and relationships) and the dynamic collaborations that occur when the objects send messages to each other to provide a given function. The static structure of the SVM Categorizer will be described in class diagrams, whereas the dynamic perspective will be described in sequence diagrams.

The transformation of the user requirements to a generic classification system should include some basic functions as well as some utilities essential for providing software quality. Figure 1 depicts all the sub-packages needed for the development of a generic text categoriser tool.

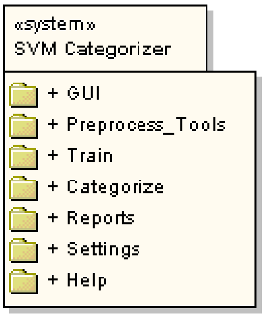

Figure 1. SVM Categorizer system package

The deployment of the system package to its sub-packages reveals the architecture of the SMV Categorizer. In Figure 2, an overview of the architecture is presented. The “Preprocess Tools” package includes all classes responsible for the transformation of the source document into an understandable input for the system. The services of this package will be used by both “Train” and “Categorize” packages as their first stage in order to pre-process the training set and the unseen documents, accordingly.

The “Train” package, which is the core of the machine learning algorithm should be able of taking the pre-processed input and produce the weights of every category, according to a user specific number of features (terms). These weights will be utilized by the “Categorise” package, where after pre-processing an unseen document, it will decide upon whether it is eligible to be categorized under a pre-specified category (binary classification). This decision will be based on the categories features produced in the training phase. It should be mentioned that the classification process will have to support multiple classification, which means that a document could belong in more than one category. Furthermore, the result of the classification can be in a hierarchical form, according to the total weight value of every document.

For example, if a document was found to belong in two categories – “financial” and “biotechnology general”–and the result of the classification was 4.56765 and 5.32323, accordingly, then the document should be classified first in “biotechnology general” category and afterwards to the “financial” one, since 5.32323 > 4.56765. This concept will increase the user’s added value, who will obtain not only the information of classification's outcome, but also the knowledge of the rules that define the relationship between a document and its category.

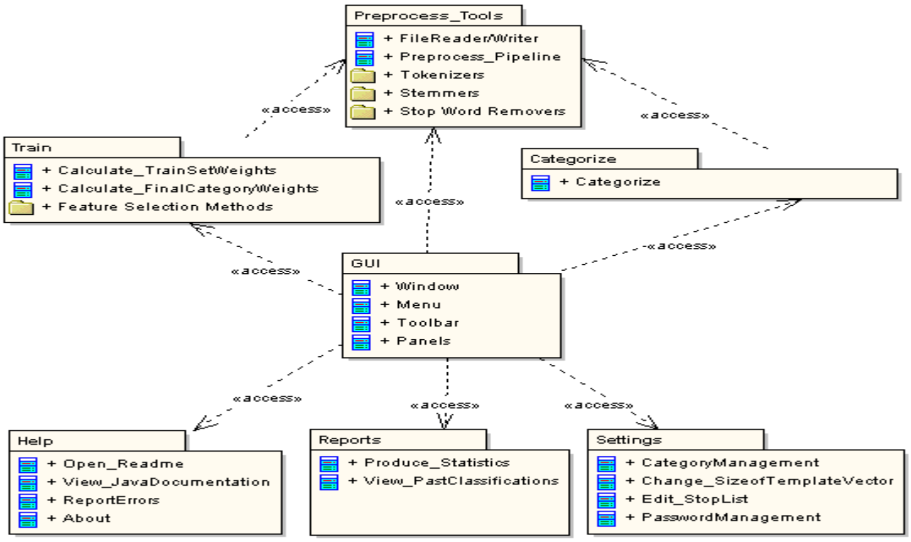

Figure 2. Package diagram – Architectural view of SVM Categorizer

3. Evaluation of the SVM Categoriser

This section evaluates the performance of the SVM Categorizer according to some pre-defined performance measures, such as precision, recall, F_1-measure, and accuracy. This has been achieved by using the data provided by Biovista (www.biovista.com) is a company that specialises in Corporate Intelligence (CI) products and services for the biotechnology and pharmaceutical industries.

The section concentrates on optimal configuration that yields the highest precision to further evaluate and compare it with other popular machine learning algorithms. Besides that, the section evaluates the SVM Categorizer as an integrated piece of software in terms of functionality and ease of use.

3.1 Categorisation Results

In order to evaluate the results the SVM Categorizer 50 testing documents were created, 10 for each category. All these documents are already pre-classified by the domain expert. This happens so as to compare the results produced by SVM Categorizer to the domain expert's estimation. This is done for every category in order to calculate the precision metric for every configuration.

In particular, the precision metric will indicate which configuration yields the highest percentage of correct classifications. Table 1 presents the results of the classification process of the 10 testing documents. In case the document of a pre-specified category is correctly classified, then the word "Yes" is assigned, otherwise the word "No" is assigned. Then, the frequency of "Yes" was calculated, which illustrates the perspicuity of the algorithm (precision metric). Table 2 shows the results and compares them to other classification algorithms. The evaluation of the proposed algorithm, which is based on SVM theory, displays interesting results.

Table 2: Classification results for 10 testing documents pre-classified under the "Biotechnology General"

Con	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6	Doc7	Doc8	Doc9	Doc10
1	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	No
2	No	Yes	No	Yes	Yes	Yes	Yes	No	Yes	No
3	No	Yes	No	Yes	Yes	Yes	Yes	No	Yes	No
4	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes

5	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
6	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
7	Yes	Yes	No	No	No	Yes	Yes	No	Yes	Yes
8	Yes	Yes	No	No	No	Yes	Yes	No	Yes	Yes
9	Yes	Yes	No	Yes	No	Yes	Yes	No	Yes	Yes

In all the metrics selected for this comparison, it seems that the proposed algorithm surpasses many existing ones (WORD, CONSTRUE, etc analyzed in the literature. In fact, its performance can be compared to the most accurate algorithm in the field of text categorisation.

Table 2: Comparison of SVM Categorizer with other algorithms

Algorithms	Recall	Precision	F_1-Measure
SVM	81.20%	91.37%	85.99%
Knn	83.39%	88.07%	85.67%
NNet	78.42%	87.85%	82.87%
NB	76.88%	82.45%	79.56%
SVM Categorizer	*80.00%*	*79.35%*	*79.08%*

More specifically, as highlighted in the previous table, the precision of the SVM Categorizer is 79.35%, which can be characterised as a trustworthy result. The recall criterion is 80%, which outperforms Neural Network algorithm, as well as Naïve Bayesian and Decision Trees. This result is rather promising and should invoke the interest of the research community. Finally, in the combined criterion of F_1-measure, the SVM Categorizer's value is very near (79.08%) to all other algorithms with which it was compared.

5. Conclusions

The contribution of this work is the development of a generic text categorisation tool that is based on a new statistical approach for calculating the feature weights contained in every category. This promise to improve classification performance since it calculates a more reliable threshold weight. The results indicate that the number of training features participating in the classifier's learning is strongly connected with the classification result. In particular, in every configuration that utilised 250 features, the classifier's accuracy was improved. The accuracy in some of the categories is better than others. For example, the "Intellectual" category had a 90% accuracy, whereas the "Research" one 69%. This phenomenon should be further investigated in order to discover the causes for this incongruity. The performance of the SVM Categorizer places it among the five most accurate classifiers, even though it should be tested at the Reuters train corpus in order to compare it under exactly equal terms.

6. References

[1] Cooley R., *"Classification of News Stories Using Support Vector Machines"*, in IJCAI'99 Workshop on Text Mining, Stockholm, Sweden, 1999.

[2] Jain K. A., Dubes C. R., *"Algorithms for clustering data"*, Prentice Hall Inc., Englewood Cliffs, New Jersey 1988.

[3] Joachims T., *"A Statistical Learning Model of Text Classification for Support Vector Machines"*, Proceedings of SIGIR-01, 24th ACM International Conference on Research and Development in Information Retrieval, 2001.

[4] Joachimes T., *"Learning to Classify Text using Support Vector Machines: Methods, Theory, and Algorithms"*, Kluwer Academic Publishers, Boston – Dordrecht – London, 2002.

[5] Makoto I. – Takenobu T., *“Text Categorization based on weighted inverse document frequency”*, Technical Report, Tokyo Institute of Technology, 1994.