

**School of Computer Science and Software Engineering**  
**Monash University**

Bachelor of Computer Science with Honours (1608)  
Research Proposal Semester 1, 2002

**Text Classification with Support Vector  
Machines**

by

**Aleksandar Milisic** 12834556  
milisic@mail.csse.monash.edu.au

**Supervisor: Dr. David Albrecht**  
dwa@mail.csse.monash.edu.au

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Aims and Significance . . . . .	1
<b>2</b>	<b>Research Context</b>	<b>3</b>
2.1	General Text Classification Algorithms . . . . .	3
2.2	Text Classifiers Learning with Unlabeled Data . . . . .	4
2.3	Text Classification with Support Vector Machines . . . . .	6
<b>3</b>	<b>Research Plan and Methods</b>	<b>6</b>
3.1	Research Methods . . . . .	6
3.2	Proposed Thesis Chapter Headings . . . . .	8
3.3	Timetable . . . . .	9
3.4	Special Facilities Required . . . . .	10
<b>4</b>	<b>Relevance of the Research</b>	<b>10</b>
<b>5</b>	<b>References</b>	<b>11</b>

# 1 Introduction

## 1.1 Aims and Significance

Recently, the amount of information available on the Internet and in other electronic forms has experienced a rapid growth. Unfortunately, humans are not very good at managing and utilizing huge collections of documents. As the amount of information stored in electronic form increases further, the need for developing tools to help people retrieve, filter and manage documents becomes more obvious.

One of the most important information management tasks that helps people handle large numbers of documents is text classification, the process of assigning text-documents to one or more predefined categories.

Text classifiers are used in everyday life. We often save our files and e-mail messages and organize them into different folders depending on their content. They are also used to classify news stories, organize document databases or even find interesting articles on the WWW. However, organizing huge collections of documents manually can be a daunting and sometimes impossible task. For these reasons, it is necessary to develop techniques that will automate the process of text classification. This has caused text classification to become an area of great interest to researchers around the world, who have developed various machine-learning algorithms for automating this task. These algorithms are commonly based on the vector space model (Kwok 1999) where sparse vectors represent the text documents and each component corresponds to a feature extracted from the document. The features describe a document in a particular way, with one of the more common approaches having them indicate the presence of words in the document (this creates a *word presence matrix*). However, text has special properties that make the task of classifying much harder than seems at first sight. Some of those special properties are presented in the paper by Joachims (1998):

- *High dimensional input spaces:* the document vectors are very high dimensional and in general have thousands of components representing features for some document collections.
- *Few irrelevant features:* feature selection (i.e. determining irrelevant features that can be discarded) might not be as promising as it sounds in terms of dimension reduction. In Joachims (1999), tests on the Reuters data collection indicate that the number of irrelevant features is insignificant. This, however, is an area that requires further investigation as to whether other types of features can produce better results.

- *Document vectors are sparse:* in most cases a single document contains a small number of words compared to the total number of different words in a document collection. This causes the document vectors to have only a few non-zero entries.

As pointed out in the paper by Kwok (1999), many existing machine-learning algorithms, such as Neural Networks, naïve Bayes and hierarchical Bayesian classifiers, rely heavily on dimension reduction which can be a computationally expensive task. Also, many of the existing techniques don't work well when new collections of documents become available after the initial training. In that case, complete re-training of the classifier is often needed. For these reasons, other techniques and approaches are being explored by researchers involved in text classification.

Another problem area in text classification is the need for machine learning algorithms to be capable of learning accurately with a small set of labeled training data and a large set of unlabeled data. Labeling documents manually is a very expensive and time-consuming task, which emphasizes the need for techniques that will work with a high level of precision even if trained with only a small set of labeled examples.

Research done on Support Vector Machines (SVM's)(Kwok 1999; Joachims 1998; Joachims 2001; Dumais 1998) indicates that this particular machine learning technique can handle the special characteristics of text very successfully as well as learn accurately given a small labeled and a large unlabeled training set, targeting them as a particularly interesting area of research. However, there still exist various issues regarding the use of SVM's in text classification such as:

- *How do SVM's handle large data sets?* The algorithms used with SVM's are quadratic with the number of variables equal to the number of data points (Kwok 1999). What could be done in order to simplify this problem? Feature selection, i.e. discarding features that don't provide valuable information, could be a starting point in exploring ways of overcoming this problem.
- *What are the most efficient techniques for handling classification problems where there are more than 2 categories?* Text classification requires this analysis since in the vast majority of cases there exist more than 2 document types (in news stories for example: weather, sports, politics, entertainment, IT etc.) and different techniques may provide different results.
- *Which document representation works best?* SVM's are designed to work with features. Therefore, data is converted to vectors before any

classification is done. There exist various techniques for converting text documents to vectors, however the question is can these techniques be further improved? In the paper by Kwok (1999), a possible alternative to document representation is mentioned which would give more emphasis to words that occur in headlines in news stories, an interesting approach that should be further investigated.

- *Can the performance of SVM's in text classification be further improved by combining them with other existing techniques?* Recent work by Raskutti et al. (2002) indicates that the performance of SVM's can be enhanced by combining them with Rasmussen's single pass clustering algorithm. This could provide a useful guide in combining SVM's with other clustering algorithms in order to possibly obtain even better results.

This project will address the above-mentioned issues related to applying Support Vector Machines to the problem of text classification. Emphasis will be placed on Transductive Support Vector Machines (TSVM's), a special flavour of SVM's, since they have achieved very promising results in training with a small set of labeled and large set of unlabeled examples, as shown in the paper by Joachims (1999).

## 2 Research Context

### 2.1 General Text Classification Algorithms

Research in text classification has been very extensive in recent years, providing many exciting and promising results. As described by Kwok (2001), there exist many different machine-learning methods used in text classification. Some of the more popular ones include:

**Linear Models**, based on linear discriminant analysis, iterative learning algorithms etc., these algorithms often encounter computational problems (for example, when dealing with very large matrices) and dimension reduction is often required.

**Neural Networks**, nonlinear classifiers that can be useful since collections of documents can't always be separated by using linear classifiers. However, dimension reduction is necessary in order to decrease the network size and, as mentioned by Kwok (1999), experimental results indicate that neural networks still don't considerably outperform linear classifiers.

**Probabilistic Models**, these include various algorithms such as the naïve Bayes and the hierarchical Bayesian classifiers. A disadvantage when

using these models is that they often make the assumption that words in a document are totally independent of each other for a given class, which in practice hardly ever holds.

**The k-Nearest Neighbour (k-NN) Classifier**, especially the **Weight Adjusted k-NN (WAKNN)** (Han et al. 2001), has shown some promising results in text classification. In WAKNN, each word from the training set is assigned an importance measure and a weight vector is maintained storing the importance for all the words in the training set. The weight vector is used to compute the similarity between documents, where more important words (as shown in the weight vector), contribute more to the final result of the "similarity" computation. However, currently this method also requires expensive computations when calculating the similarity between a new document and every training document.

For all of these models it is common that they often encounter computational problems when presented the task of classifying large volumes of text documents, as well as the fact that they don't work well in dynamic environments where a trained machine might have to handle frequent additions to the document collection.

## 2.2 Text Classifiers Learning with Unlabeled Data

As mentioned earlier, the main goal for machine learning algorithms used in text classification is the possibility of training the machines with a small set of labeled and a large set of unlabeled data. Various algorithms have been designed for solving this particular problem. Following is a brief overview of some of these algorithms.

A combination of the Expectation-Maximization (EM) algorithms and a naïve Bayes classifier, as presented in Nigam et al. (2000), shows that the accuracy of classifiers can be improved by augmenting a small number of labeled training documents with a large collection of unlabeled documents. To achieve improvement with the EM algorithm, some assumptions about how the data are generated must be satisfied (for example, the naïve Bayes classifier assumes words in text are independent of each other for a given class). However, in practice, text rarely complies with these assumptions, causing the algorithms performance to actually deteriorate. In order to avoid that, extensions to the existing algorithm are suggested (Nigam et al. 2000) with one of the possibilities being a weighting factor that influences the unlabeled data's contribution to parameter estimation in EM.

In their paper, Blum and Mitchell (1998), describe the co-training algorithm. They assume that there exist two redundant sets of labeled data and

that both would be sufficient for successful classification if there was enough labeled data. This allows the training of two separate classifiers that after being initially trained use their predictions to label the much larger unlabeled data set. The predictions on unlabeled data, produced by both classifiers, are then used to increase the training set of the other classifier. This means that the redundancy of features allows learning two distinct classifiers that in turn can now train each other over unlabeled data. An experiment on classifying web pages, where a classifier was learned for finding home pages of various Computer Science departments in 4 American universities, showed that the use of unlabeled data actually provides superior performance compared to just using the small labeled set.

A variation of the co-training algorithm mentioned above is described in the work by Goldman and Zhou (2000), where there are no assumptions made about having two redundant views that are sufficient for successful classification. Instead, there are two learning algorithms that are both trained on the labeled data, and then each learner takes a subset of the unlabeled data and labels it for the other. This is repeated until there is no more data to be selected for labeling. In the end, the two resulting hypothesis are combined to give the final hypothesis. Experiments done by Goldman and Zhou have shown that this technique outperforms certain learning algorithms such as ID3 and HOODG.

Some algorithms don't label the unlabeled data, but rather use other techniques for utilizing the information provided with the unlabeled documents. The following two are typical examples:

In the paper by Zelikovitz and Hirsh (2000), a technique is developed for improving the classification of short text strings (for example, labeling the title of a paper) using a combination of labeled and unlabeled but related documents. It involves the use of WHIRL, a tool that uses database-like functions for classifying data. The unlabeled data here is viewed as "background knowledge", so the labeled training example can then be useful for classifying a test document if there exists some background knowledge that is similar to both the labeled and test example. This is called a "second-order" approach to text classification.

A method that has produced some very promising results is adding cluster parameters to the classification (Raskutti et al. 2002). This method clusters both the labeled and unlabeled data and then analyzes those clusters to obtain new features which are added to the input feature space. It doesn't make any assumptions about prior class distributions, which makes it suitable for cases where the class distributions of the test and training data differ. This in turn allows it to be able to manage dynamic environments where new additions to the document collection are frequent. Experiments

using this clustering technique have been done on SVM's and they show an improvement in the performance of SVM's even if the labeled set is only 0.25% of the total training set. It must be noted, however, that experiments were done using the word presence matrix and Rasmussen's clustering algorithm. Further improvements might be possible by choosing different document representations or clustering algorithms, an idea that should be investigated.

## **2.3 Text Classification with Support Vector Machines**

Research done so far (Joachims 1998; Joachims 2001; Kwok 1999; Dumais 1998), indicates that Support Vector Machines perform very well when it comes to text classification. They handle high dimensional input spaces better than most other common algorithms (Joachims 2001), since they are independent of the dimensionality of the feature space and this in turn allows them to adapt efficiently to dynamic environments that require frequent additions to the document collection. The superior performance of SVM's in text classification makes them an interesting method to explore, and several techniques implementing this method have been devised.

Joachims (1999), presents an extension on SVM's, Transductive SVM's (TSVM). They have proved to be an improvement over ordinary SVM's, especially for learning with large sets of unlabeled documents. The main difference between the two is that while SVM's induce a general decision function for a learning task (i.e. for the whole test set), TSVM's consider a certain test set and try to minimize misclassifications of just those particular examples. Experiments using these techniques (done on standard benchmarks such as Reuters, WebKb and 20 Newsgroups) show that they provide superior performance compared to other algorithms.

As mentioned earlier, the aim of this project is to explore and combine these existing techniques in search of further improvements in terms of the performance of Support Vector Machines in text classification.

# **3 Research Plan and Methods**

## **3.1 Research Methods**

This project will continue on the current research done on SVM's and TSVM's and their application to text classification, with the aim of resolving some of the issues mentioned in the introduction as well as improving performance.

Since the clustering algorithm combined with SVM's has shown some promising results (Raskutti et al. 2002), this project will explore a combination of the clustering method and TSVM's. This technique could possibly provide even better results, considering the advantage TSVM's have over SVM's in terms of performance (Joachims 1999). Also, different clustering algorithms may be examined in search of one that could outperform Rasmussen's clustering algorithm, used by Raskutti et al.(2002). No particular clustering alternative has been chosen as yet, but SNOB, among others, may be a possibility.

Experiments done by Raskutti et al. (2002) were done using the word presence matrix, a simple representation that worked well, but this project will try and explore different ways of representing documents that might enhance performance. One possibility could be the Inverse Document Frequency (IDF) (Kwok 1999), but other methods may be explored as well.

Feature selection, the process of eliminating redundant features that do not give any valuable information, will be explored, even though there are some indicators (Joachims 1999) that text in fact hasn't got many irrelevant features. However, this project will further investigate methods for selecting irrelevant features according to their distribution. In Joachims (1999), this particular property of text was discovered after experiments done with a naïve Bayes classifier, so the possibility of obtaining more promising results with other classifiers will be explored.

As time allows, the problem of text classification where there are more than 2 classes (which most commonly is the case) and ways of dealing with this problem will be explored.

Testing the different approaches and solutions to problems mentioned above will be done using the SVM-Light package since this program has the option of using the transductive classifier. Programming will be done in C and the results of these experiments will be constantly verified by the use of the precision-recall method on the Reuters corpus since this seems to be a widely accepted standard in the area of text classification research.

Potential problems could be caused by the fact that words are not probabilistically independent of each other which could cause difficulties when investigating ways of discarding irrelevant features from the data. Also, some methods used for document representations, such as giving different weights to words depending on where they appear in the document, might not be easy to generalize since not all document collections have the same format.

## 3.2 Proposed Thesis Chapter Headings

1. Introduction
  - (a) Purpose of Research
  - (b) Objectives of Research
2. Text Classification -Problems and Approaches
  - (a) Problems in Text Classification
  - (b) General Approaches
  - (c) Support Vector Machines - Advantages and Issues
3. Support Vector Machines
  - (a) What are They?
  - (b) How do They Work?
  - (c) Transductive Support Vector Machines?
4. Clustering Techniques
  - (a) Overview
  - (b) k-Nearest Neighbour(k-NN)
  - (c) Naïve Bayes
  - (d) The Expectation-Maximization (EM) Algorithm
5. Can It Get Better?
  - (a) Combining TSVM's with Clustering Algorithms
  - (b) Feature Selection
  - (c) Document Representation
  - (d) Multi-class classification
6. The Final Verdict
  - (a) Results
  - (b) Discussion
7. Conclusion and Future Work
8. Bibliography
9. Appendix A Test Data
10. Appendix B Performance Charts
11. Appendix C Programs and Algorithms

### 3.3 Timetable

Date	Activity
2 May	Research Proposal Finalized
3 May	Begin Literature Review
3 May	Begin Implementation of Clustering Algorithm and TSVM
20 May	Prepare Literature Review Draft
22 May	Begin Implementation Testing
2 June	Prepare for Interim Presentation
6 June	Interim Presentation
13 June	Finalize Literature Review Draft
14 June	Finalize Implementation of Clustering Algorithm and TSVM
22 June	Commence Work on Feature Selection
3 July	Commence Feature Selection Testing
22 July	Finalize Work on Feature Selection
1 August	Finalize Literature Review
2 August	Commence Work on Document Representations
12 August	Commence Document Representation Testing
29 August	Finalize Work on Document Representations
30 August	Thesis - Chapters 1 and 2
8 September	Thesis - Chapters 3 and 4
17 September	Thesis - Chapters 5 and 6
27 September	Submit Thesis Draft to Supervisor
28 September	Commence Work on Multiple Classes (if time allows)
8 October	Commence Multiple Class Classification Testing
13 October	Finalize Work on Multiple Classes(if time allows)
15 October	Thesis - Update Chapters 5 and 6, Finalize Chapter 7
21 October	Submit Thesis Draft to Supervisor
22 October	Prepare for Final Presentation
28 October	Final Presentation
30 October	Finalize Thesis

### **3.4 Special Facilities Required**

The facilities offered Honours students at Monash University (Clayton) are sufficient to do the research.

## **4 Relevance of the Research**

Improving the performance of text classifiers is of great importance in various segments of the industry (business, news) because the number of documents that are being handled on a daily basis is increasing rapidly by the day and becoming more and more daunting for humans to cope with.

Research done so far on Support Vector Machines shows their great potential in application to text classification and superior performance in comparison with other common classifiers, especially when the classifier is learned only with a small set of labeled and a large set of unlabeled examples. However, there still remain certain issues to be resolved that could further enhance the performance of SVM's in terms of both space complexity and precision.

This project will tackle those issues in search for more efficient and effective solutions and hopefully provide useful guidelines for future work done on this topic.

## 5 References

### References