Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 61 (2013) 16769-16772

ISSN: 2229-712X

A Case Study on Document Classification Using SVM

Rakesh Chandra Balabantaray[1] and Parismita Goswami[2]

[1]IIIT Bhubaneswar, Bhubaneswar, Odisha, India.

[2]Department of Information Technology, Gauhati University, Guwahati, India.

ARTICLE INFO

Article history:
Received: 4 June 2013;
Received in revised form:
24 July 2013;
Accepted: 1 August 2013;

Keywords
SVM,
Rampant,
Domain.

ABSTRACT

The need of automatic classification of text has become very crucial with the rampant growth in the amount of information available in today's world of fast digitization. Text classification (also called text categorization) is the process of classifying documents into pre-defined categories. A number of techniques have been devised over the ages for text classification. One such technique for text categorization is Support Vector Machine (SVM). SVMs are binary classifiers which classifies text into exactly two categories. Many researchers have found that SVM works reasonably well. In this paper, this technique is used for classifying a set of documents. Five models have been created from five different domains and data from each domain are classified using these models and the accuracy is recorded.

Introduction

In the recent era, the web has facilitated the finding of formation by the mere typing of the words on the keyboard or by a simple mouse click. This has led to the explosive growth of online information and with it has arisen the need to categorize texts into pre-defined categories for better organisation of data and retrieval. Thus automated text categorization or classification has become a booming interest amongst researchers in the field of information retrieval. Classification enables automatic routing of a particular document into some pre-specified sub-collection. According to [1], if d_i is a document of the entire set of documents D and $\{c_1, c_2, \dots, c_n\}$ is the set of all the categories, then text classification assigns one category c_j to a document d_i. Text classification has become a necessity in this world of fast digitization. The first step in text categorization is to transform documents, which typically are strings of characters, into a representation suitable for the learning algorithm and the classification task. Text categorization requires a feature-value representation of text. Each distinct word fi corresponds to a feature, with the weight of the word as the value. The weight can be calculated by finding the TF-IDF [6] of each word. This representation scheme leads to very high-dimensional feature spaces containing thousands dimensions and more. A number of classifiers have been devised for the purpose of document classification. A relatively new learning method called the Support Vector Machine(SVM) can be used for the purpose of classification[2]. SVM's are binary classifiers which mainly centers on finding a hyperplane that maps data in an input space into a high A number of classifiers have been devised for the purpose of document classification. A relatively new learning method called the Support Vector Machine(SVM) can be used for the purpose of classification[2]. SVM's are binary classifiers which mainly centers on finding a hyperplane that maps data in an input space into a high-dimensional feature space in such a way as to render a problem linearly separable. In this paper, classification of documents is carried out using SVM. This is implemented on the software called SVM-Light. A collection of four hundred documents from five domains namely entertainment, literature, politics, sports and zoology are taken which are used to create the train and sets for SVM-Light. Five different models, one from each domain, are created with the training data that is taken, and these models are used to classify three test sets from each domain. The accuracy value for each test set is recorded.

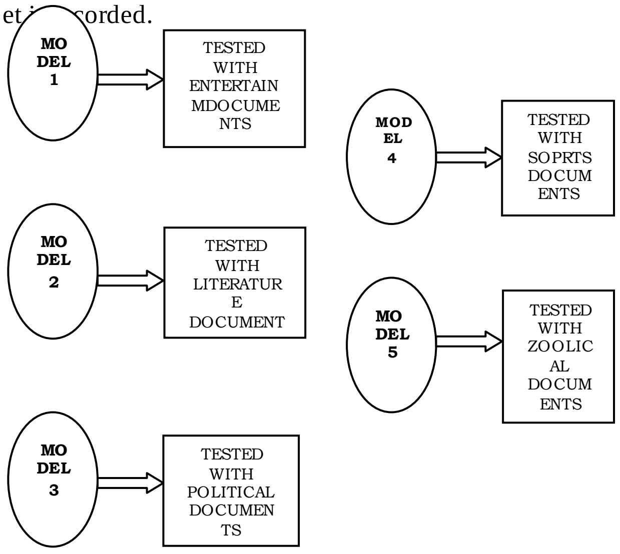

MO DEL 1
TESTED WITH ENTERTAIN MDOCUME NTS
MO DEL 2
TESTED WITH LITERATUR E DOCUMENT
MO DEL 3
TESTED WITH POLITICAL DOCUMEN TS
MOD EL 4
TESTED WITH SOPRTS DOCUM ENTS
MO DEL 5
TESTED WITH ZOOLIC AL DOCUM ENTS

Figure: Five Models for training SVM

Support Vector Machines

Support Vector Machine (SVM) is a learning machine that uses supervised learning to perform data classification. In SVM, each line within the data set is given a label and SVM learns the data and puts the new data in the group that is closest to the learned data. Support Vector Machines are based on the concept of decision planes that define decision boundaries. The SVM analysis attempts to find a 1-dimensional hyperplane (i.e. a line) that separates the cases based on their target categories. The SVM's algorithm first starts learning from data that has already been classified, which is represented in numerical labels (e.g. 1, 2, 3, etc.) with each number representing a category. Once SVM

Tele:
E-mail addresses: rakeshbray@gmail.com

determines the points that are closest to each other, it calculates the hyperplane, which is a plane that separates the labels.SVM uses a *kernel function* to map the data into a different space where a hyperplane can be used to do the separation. To do this, we define a mapping z = Φ(x) that transforms the d dimensional input vector x into a (usually higher) d0 dimensional vector z. For example, a simple kernel like Φ(X1) = (X1,$X1^2$) can be chosen. This kernel is actually a polynomial type. This kernel will map the points to a 2-dimensional feature space by multiplying the points to the power of 2. The next step is finding a hyperplane. This can be done by using the following equations[7]:

$\langle w \cdot x \rangle + b = +1$ (positive labels)

$\langle w \cdot x \rangle + b = -1$ (negative labels)

$\langle w \cdot x \rangle + b = 0$ (hyperplane)

The above equations are solved by using linear algebra, to determine the values of w and b which satisfies the above equations. Many times, there are more than one solution or there may be no solution, but SVM can find the optimal solution that returns a hyperplane with the largest margin.In SVM, this model is used to classify new data. With the solutions, new data can be classified into category. For example, if the result is less than or equal -1, the new data belongs to the -1 class and if the result is greater than or equal to +1, the new data belongs to the +1 class.

Reviews

Document classification is one of the important techniques to handle and manage the large amount of information and data that have come into being after the inception of digitization. An illustration was made of the process of text classification using various techniques of machine learning[1]. The support-vector network is a new learning machine for two-group classification problems. The idea behind the support-vector network was previously implemented for the restricted case where the training data can be separated without errors. However, a new technique was discussed to demonstrate high generalization ability of support-vector networks utilizing polynomial input transformations [2]. A comparison between four different classifiers for classification and three classifier combination approaches were investigated, the performance of each method recorded and the accuracy achieved evaluated [3]. An introduction to Support Vector Machines was made and the various aspects and working of SVM discussed[4]. Some challenging research problems that can be found in the area of text classification was proposed and then the feature set reduction methodology as one of the key topics discussed[5]. The results of applying Term Frequency Inverse Document Frequency (TF-IDF) to determine what words in a corpus of documents might be more favorable to use in a query was discussed along with highlights at the advantages and disadvantages of using the TF-IDF scheme. Though TF-IDF is a relatively old weighing scheme, it is simple and effective[6].

Methodology

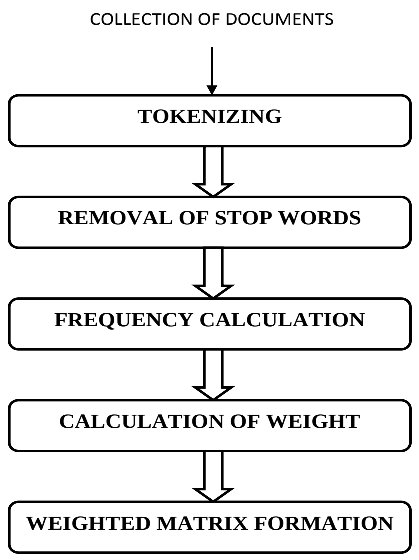
Before the commencement of the process of document classification the dataset must be converted into a format which is suitable for SVM-Light. This involves proper preparation of the data set which can be achieved by the process of pre-processing.

Data Preparation For SVM

A collection of four hundred documents are taken as the data set. These documents are from five different domains namely Entertainment, Literature, Political, Sports and Zoology, eighty from each domain. For each pre-defined category, we need a set of training documents known to belong to that category. From the training documents, we train a classifier. These documents are now used to make the test and train sets. The test set comprises of the documents which are to be classified. These documents undergoes pre-processing are then converted into SVM format.

Preprocessing

Preprocessing consists of steps that take as input a plain text document and outputs a set of tokens.



Tokenization

Tokenization splits sentences into individual tokens, typically words. The aim of the tokenization is the exploration of the words in a sentence. Furthermore, tokenizer can cater for consistency in the documents. Use of tokenization is identifying the meaningful keywords.

Stopwords Removal

Removal of words which is used in abundance, ample use of which may lead to high dimensionality problem is carried out. Words like “a”, “an”, “the”,“ is”, “ are” ..etc., which is used again and again has been removed.

Frequency Calculation

The frequency of occurrence of each word in a document is calculated and is put in a Hashtable named frequencyMap . The key-value pair in the Hashtable is given by the current word as the key and frequency as the value.

Weight calculation

The weight of each word has been calculated by using the TF-IDF metric. TF-IDF (term frequency–inverse document frequency) is used to determine the importance of a word in a document.

The term frequency tf of term *t* in document *d* is defined as the number of times that *t* occurs in *d* and it can be calculated as:

$$\textbf{Tf=}\ \frac{\textit{frequency of word t in document d}}{\textit{total number of words in document d}}$$

The inverse document frequency is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

IDF is calculated as:

$$\log \frac{\textit{Total number of documents taken}}{\textit{the number of documents in which the word t is present}}$$

Document representation

Thus the high dimensional data can be now represented in two dimensional format in the form of a matrix. The rows in the matrix represents document and the columns represent tf-idf weights of the words pertaining to a document.

$$\begin{array}{c|cccc|} & T_1 & T_2 & \dots & T_t \\ D_1 & w_{11} & w_{21} & \dots & w_{t1} \\ D_2 & w_{12} & w_{22} & \dots & w_{t2} \\ : & : & : & & : \\ : & : & : & & : \\ D_n & w_{1n} & w_{2n} & \dots & w_{tn} \end{array}$$

Figure 3.9: Document-Weight matrix

Converting Data Set Into SVM Format

The SVM algorithm works on numeric data. Thus the documents must be converted into a format which the SVM understands. The SVM input format is:

<line> .=. <target> <feature>:<value> <feature>:<value> ... <feature>:<value> #
<info>
<target> .=. +1 | -1 | 0 | <float>

Each line represents a document or a training or a testing example (as be the case).

Target:

It is sometimes referred to as 'class', the class (or set) of our classification. Usually we put integers here. The target value denotes the class of the example. +1 as the target value marks a positive example, -1 a negative example respectively.

Feature:

These are ordered indexes, usually continuous integers. They generally represent the position of each word in the document.

Value:

The value represents the weight of each word which is calculated by finding the tf-idf values of each word as mentioned above.

Feature/value pairs MUST be ordered by increasing feature number. Features with value zero can be skipped. This is the input format of SVM-Light. The train and the test files are made in this format. These files should have the extension ".dat".

Working With SVM-Light

Working with SVM requires the creation of train and test set in the format afore mentioned. The train set is used by the classifier to train the classifier. The SVM-Light has two modules: svm_learn and svm_classify. The svm_learn creates a model file based on the supplied training data. This model file is used by the module svm_classify to predict the categories of the test data. "svm_learn" is called with the following parameters:

svm_learn [options] train_file model_file

In all modes, the result of svm_learn is the model which is learned from the training data in train_file. The model is written to model_file. The model file generated is now used to make predictions on the test-file using svm_classify. The predictions given by the svm_classify module is then compared with the original label of each test document. "svm_classify" is called using the parameters:

svm_classify test_file model_file predictions

Creating The Training Data

In this paper, the creation of five models each from entertainment, literature, political, sports and zoological domains is discussed. This requires the making of five train set from each domain. The train sets consists of 120 documents. For example, for entertainment model, 60 documents from entertainment and 60 documents from the rest of the domains are used to create the training data. The entertainment documents are given a target value of +1 and the rest are given a target value of -1. Similarly for literature, 60 documents from literature domain and 60 documents from the rest of the domains are used to create the model and so on. The train set for entertainment domain is given below:

```
+1 1:0.020297224142817732
4:0.006860685370363139
5:0.003824861980419004 …..
+1 1:0.02567834868300662
2:0.02567834868300662 3:0.0128391743415033
……
-1 1:0.013234573371862351
2:0.005412593104751395
4:0.00697314215836723…
```

This train set is used for learning by the SVM. Based on this train set, the SVM creates a model file using the command for svm_learn by typing the following command on the command-promt:

svm_learn ent/train_ent.dat ent/model

Creating The Test Data

The "model" file created is used by SVM-Light to classify the test sets. The documents in the test are the documents to be classified by the software. The test file is created exactly like the train set. Here we have tested the model with three different test sets each with different test documents. The first test "test_ent1.dat" set contains of 100 documents, 20 from entertainment, and 80 from other domain. The second test "test_ent2.dat" set contains 20 documents from the entertainment domain and 20 from the rest. The third set "test_ent3.dat" contains 20 documents from the entertainment domain and 4 from the rest. The accuracy given for each test set is then recorded and the test set that gives the best accuracy is recorded. Similarly, it is done for the other four domins.

Classification Using SVM-Light

The test sets created above are then classified by SVM with the help of the model file produced by it. The SVM gives as output a "prediction" file based on which it calculates the accuracy of prediction, i.e. how accurately it has classified the test documents. Classification is done by the module svm_classify which is done by typing the command as shown below, in the command prompt:

svm_classify ent/test_ent1.dat ent/model ent/ent_prediction1

Results And Discussions

The result of classification obtained by testing with SVM-Light is given in the table below:

This training data supplied is used by SVM to learn and then create the models which are used for predicting the class of the test data. These models are tested with three sets of testing data containing different set of documents. From the above table it can be observed that the third test set with 24 documents gives the highest accuracy from the entertainment, literature, political and sports. The zoological documents, however, gives the highest accuracy for the second test set with a total of 40 documents.

Model1: for entertainment:

NO OF DOCUMENTS IN TRAIN SET	NO. OF DOCUMENTS IN TEST SET	ACCURACY
60 FROM ENTERTAINMENT 60 FROM REST	20 Entertainment 80 Rest	33%
	20 Entertainment 20 Rest	47.50%
	20 Entertainment 4 Rest	91.4%

Model 2: For Literature

NO OF DOCUMENTS IN TRAIN SET	NO. OF DOCUMENTS IN TEST SET	ACCURACY
60 FROM LITERATURE 60 FROM REST	20 Entertainment 80 Rest	52%
	20 Entertainment 20 Rest	70%
	20 Entertainment 4 Rest	86.96%

Model 3: For Political

NO OF DOCUMENTS IN TRAIN SET	NO. OF DOCUMENTS IN TEST SET	ACCURACY
60 FROM POLITICAL 60 FROM REST	20 Entertainment 80 Rest	45%
	20 Entertainment 20 Rest	60%
	20 Entertainment 4 Rest	91%

Model 4: For Sports

NO OF DOCUMENTS IN TRAIN SET	NO. OF DOCUMENTS IN TEST SET	ACCURACY
60 FROM SPORTS 60 FROM REST	20 Entertainment 80 Rest	44%
	20 Entertainment 20 Rest	50%
	20 Entertainment 4 Rest	87.50%

Model 5: For Zoology

NO OF DOCUMENTS IN TRAIN SET	NO. OF DOCUMENTS IN TEST SET	ACCURACY
60 FROM ZOOLOGY 60 FROM REST	20 Entertainment 80 Rest	58%
	20 Entertainment 20 Rest	65%
	20 Entertainment 4 Rest	41%

Conclusion

The result of classification based on the working of SVM-Light have been found and recorded. Five different models based on the training data from five domains namely entertainment, literature, political, sports and zoological domain have been created by taking 120 documents as the training set. These models have been used to classify documents arranged in three different test sets from each domain. The accuracy given by each test set is then recorded. This paper is an attempt at understanding the working of SVMs for document classification. The working of the software called SVM-Light has also been highlighted in this paper.

References

1. Ikonomakisantis,M. , Kotsiantis, S., Tampakas, V. WSEAS TRANSACTIONS on COMPUTERS,4, Pages 966-974, August 2005
2. Cortes,C., Vapnik,V. Support-Vector Networks. Machine Learning, 20, Pages 273-297,1995.
3. LI, Y. H., JAIN, A. K. Classification of Text Documents. The Computer Journal 41(8), Pages 537-546, 1998.
4. Boswell, D .Introduction to Support Vector Machines, 2002.
5. Fuka, K., Hanka,R., Feature Set Reduction for Document Classification Problems. IJCAI-01 Workshop: Text Learning: Beyond Supervision. 2001.
6. Ramos, J. Using TF-IDF to Determine Word Relevance in Document Queries. Proceedings of the First Instructional Conference on Machine Learning. 2003.
7. Vapnik, V. N. “The Nature of Statistical Learning Theory”,1999.