AN E-MAIL FILTERING AGENT BASED ON SUPPORT VECTOR MACHINES

BY

CONSTANTIN LAZURCĂ and FLORIN LEON

Abstract. E-mail filtering has recently become an important issue due to the increasing popularity of the electronic mail communication. Therefore, there is a constant need to improve the detection of unsolicited messages, or spam. Many researchers have applied machine learning techniques for filtering spam messages, and they were proven to be successful. In this paper we present a spam detection agent based on support vector machines (SVM), one of the best classification methods available today. We test several methods of extracting numerical features from text documents, and assess the optimal values of SVM parameters needed for this classification problem. The best results show a very good classification accuracy of 94%.

Key words: e-mail filtering agent, support vector machines, classification.

2000 Mathematics Subject Classification: 68T50, 68T42.

1. Introduction

Since its early years, Internet was most popularly used as a means of communication. From the beginning, users exchanged messages in a system similar to the postal system. Thus, e-mail service has developed, with all the benefits of speed and flexibility that traditional mail does not have. Today’s e-mail services are available in a wide range of choices, from servers and clients using protocols such as POP (Post Office Protocol) and SMTP (Simple Mail Transfer Protocol) to complex e-mail services that work directly in the browser.

One way to make profit from electronic mail is advertising. Many companies are willing to pay considerable amounts of money to ensure that an advertising message on their products reach a certain number of users. Often the recipient of these messages has not expressed a desire to receive them and this practice regularly prevents the efficient use of his/her e-mail service. Briefly stated, unwanted messages are a source of profit for the senders and a source of frustration for the receivers.
Presently unwanted messages, or spam, total over 80% of all messages sent to the servers of an e-mail service provider. Out of these, 81% spam messages contain information on pharmaceutical advertising, 1% contain advertising information on online casinos and 2.3% are attempts to obtain confidential data such as bank details or personal information from the recipient. The majority of all spam, approximately 80% is automatically sent by computers infected with viruses. Most owners of an infected computer do not know that their device is part of a spamming network.

Unwanted messages in 2007 caused damage of over a hundred billion dollars. Given this considerable sum, e-mail service providers and end users always look for new ways to detect and remove unwanted messages. New methods are continuously being developed to detect them, but spam senders regularly improve their techniques to overcome the detection methods. This creates a continuous battle between spammers and researchers seeking new ways to protect users from the burden of spam.

Spammers have quickly learned how to pass a filter based on content. Thus, parsing researchers reached the conclusion that in order to be efficient, filters based on artificial intelligence have to be frequently updated with new data. Sometimes even the updates were not frequent enough to prevent all unwanted messages, as senders found techniques to hide their content. Therefore, most filters based on artificial intelligence are accompanied by filters based on black-lists or other methods.

2. Spam Detection

Spam research is a fascinating subject taken separately, but it is equally interesting how it relates to other areas. Most spam filters use at least one method of artificial intelligence. Spam research has revealed deficiencies in current artificial intelligence technologies and has helped to remedy them.

2.1. Content-Based Filtering

By the time spam was becoming a major problem, Microsoft research division started work in 1997 on developing methods of artificial intelligence that could be trained to filter spam [9]. In this approach programs are provided examples of wanted e-mail and examples of spam e-mail. A learning algorithm is then used to automatically find features of wanted e-mail in comparison with the characteristics of spam messages. After the training process is complete an incoming message can be classified as having a high probability to be ham (a desired e-mail), a high probability of being spam, or a value in-between. The first attempts were relatively simple when matched up to today’s technologies, using the naïve Bayes method based on how often a word or other features appear in a spam message and in a message that is desired.
Algorithms based on support vector machines can reduce by half the amount of spam passing the filters based on naïve Bayes method. These learning processes may require repeated adjustments of the parameters that influence the results, but a filter based on over 20000 messages can now be trained in less than an hour due to the continuous development of technology.

Spam filtering is not the only beneficiary of the development of training techniques of artificial intelligences, but also motivates new, interesting research. For example, AI algorithms are typically designed to maximize accuracy (how often their predictions are correct). But in practice, on spam filtering, fraud detection, and many other problems, these systems are too conservative. Only if the algorithm is almost certain that a message is spam, the message is to be classified as spam. This problem has recently led researchers to develop special methods of training for these particular situations. A clever technique that can reduce spam by 20% or more, developed by Scott Yih at Microsoft research center, involves the creation of two filters. The first is trained to identify the difficult cases, and the second is only trained to classify these cases. By focusing on such cases the overall results are improved [15].

2.2. Filtering by Using Sender’s Address

Filtering techniques based on message content are sometimes too easily defeated because of the numerous ways to hide content. Thus, many researchers in the field of spam filtering have focused on aspects of spam messages which cannot be hidden, e.g. the sender of a message, identified by his/her IP address, is the most important of them.

2.3. Secure Identity

Numerous attempts have been made to introduce cryptographically secure identity to e-mail, including standards such as PGP and S/MIME, but none has been widely adopted. Identity is particularly important in spam filtering. Almost all spam filters have some form of safe lists, allowing users and administrators to identify senders whose e-mails can be trusted. Traditional cryptographic approaches to identity security resisted most attacks, but they were too difficult to implement for practical reasons.

3. Support Vector Machines

Support vector machines (SVM) are classification systems that use hypotheses constructed in a multidimensional space, driven by an optimization algorithm derived from statistical learning theory [13]. This learning method developed by Boser, Guyon and Vapnik [1] is based on well-defined mathematical principles. Shortly after it was introduced, it quickly overcame in performance the majority of other systems such as the classical multilayer
perceptron neural networks in a wide range of applications. Support vector machines have become popular because of the success they had in recognizing handwritten digits with an error of only 1.1% on the test set. Due to good experimental performance, they are considered by many researchers as the best current method of classification [3].

In the support vector machine classification problem we are given \(l\) examples \((x_i, y_i), \ldots, (x_l, y_l)\), with \(x_i \in \mathbb{R}^n\) and \(y_i \in \{-1, 1\}\) for all \(i\). The goal is to find a hyperplane and threshold \((w, b)\) that separate the positive and negative examples with maximum margin, also penalizing points inside the margin based on a user-selected regularization parameter \(C > 0\). The SVM classification problem can be restated as finding an optimal solution to the following quadratic programming problem:

\[
\begin{align*}
\min_{w,b, \xi} & \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{l} \xi_i \\
\text{subject to} & \quad y_i (w \cdot x_i + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l \\
& \quad \xi_i \geq 0
\end{align*}
\]

This formulation is motivated by the fact that minimizing the norm of \(w\) is equivalent to maximizing the margin; the goal of maximizing the margin is in turn motivated by attempts to bound the generalization error via structural risk minimization [8].

Due to the very large number of training vectors and problem dimensions, it was found that a much more efficient way of solving this optimization problem was to address its dual form:

\[
\begin{align*}
\max_{\alpha} & \quad W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \\
\text{s.t.} & \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m \\
& \quad \sum_{i=1}^{m} \alpha_i y_i = 0
\end{align*}
\]

Support vector machines use a function \(\Phi\) to first map the examples into a higher dimensional space and then construct a separating hyperplane there. The idea is to transform the data into a new space where the data becomes linearly separable. Then, using the hyperplane as a decision function, we can classify unseen data based on which side of the hyperplane they lie.

Transforming data with \(\Phi\) can be expensive in high dimensional spaces. Instead, an SVM employs a kernel function \(K\) which gives the dot product of the two examples in the higher dimensional space without actually
transforming them into that space. This notion dubbed as the “kernel trick”, allows us to perform the $\Phi$ transformation for purposes of classification to large dimensional spaces.

One issue with SVMs is finding an appropriate kernel for the given data. Most research relies on a priori knowledge to select the correct kernel and then tweaks the kernel parameters via machine learning or trial-and-error [11].

We can thus apply a kernel function to the dual form of the optimization problem from Eq. 2, giving:

$$\begin{align*}
\max_\alpha W(\alpha) &= \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y^{(i)} y^{(j)} \alpha_i \alpha_j K(x^{(i)}, x^{(j)}) \\
\text{s.t.} &\quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m \\
&\quad \sum_{i=1}^{m} \alpha_i y^{(i)} = 0
\end{align*}$$

The choice of the kernel function $K(x^{(i)}, x^{(j)})$ and the resultant feature space determines the functional form of the support vectors; thus, different kernels produce different levels of performance. Some commonly used kernels are [4], [5] and [7]:

(4) Linear: $K(x, y) = (x \cdot y)$

(5) Polynomial: $K(x, y) = (x \cdot y)^d$

(6) Radial Basis Function (RBF): $K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$

(7) Sigmoid: $K(x, y) = \tanh(x \cdot y + \theta)$

Sequential Minimal Optimization (SMO) is an algorithm for solving large quadratic programming (QP) optimization problems, widely used for the training of support vector machines. First developed by John C. Platt [7], SMO breaks up large QP problems into a series of smallest possible problems, which are then solved analytically [14].

4. Case Studies

Although support vector machines are a very good classification technique, they have to be configured for each type of problem, if the user wants the best results. For every type of classification problems it was necessary to develop a specialized support vector machine with a specialized radial basis function kernel of the form: $K(x, y) = \exp\left(-\gamma \cdot \|x - y\|^2\right)$ and unique parameters.
4.1. Methodology

Training a support vector machine is a long process requiring a large amount of data and many experiments to achieve optimal results. For this application we chose to train the machine on data freely available from TREC [6], a project supported by the U.S. government, primarily aimed at building a set of text data as a benchmark for classification systems. Gordon V. Cormack founded and coordinated a special collection to evaluate spam filters on real e-mail messages. More importantly, it defines standard measures and collections for future tests. It is based on two collections of e-mail:

- **Synthetic**, consisting of a collection of published e-mail, combined with a set of recent, carefully modified spam messages. The collection can be freely distributed, and researchers can test their filters on it;
- **Private**, where researchers submit their code to testers which run it on private collections and return summary results only, thus maintaining confidentiality.

The results in terms of differentiating spam e-mails from desired e-mail on the two collections are similar, suggesting that conclusive tests can be performed on the synthetic data set without affecting the validity of the results. Therefore, we decided to use this public data set for training and testing the support vector machine.

The current version and the one used for the tests presented in this paper is TREC 2007 and contains over 75000 messages.

The documentation provided by Chang and Lin [2] describes the cross-validation method for finding optimal parameters for training a support vector machine. The method involves dividing the data into a number of subsets, usually five, using one part as test data, and the others as training data. The method iterates over different values of the $C$ parameter, which is used to balance the desire to obtain fewer errors on the training set and the desire to create a more general-purpose machine and the $\gamma$ parameter, which gives a measure of the influence a support vector has on the data space. Standard heuristics for iterating over the values of these parameters are:

- For the $\gamma$ parameter the values iterated should be from $2^{-15}$ to $2^3$ with $2^2$ as the iteration step;
- For the $C$ parameter the values iterated should be from $2^{-5}$ to $2^{15}$ with $2^2$ as the iteration step.

Thus, for each combination of values $C$ and $\gamma$ a training and an evaluation phase are performed, computing the accuracy as the number of vectors correctly classified relative to the total number of vectors, expressed as a percentage value. After each training stage the resulting accuracy is compared with the previous values and the values for $C$ and $\gamma$ with the highest precision are saved. Once the parameters $C$ and $\gamma$ that give the best results are determined, a final training cycle is run for the whole training data set, and a test set is then used to evaluate the generalization capabilities of the model.
To perform experiments with the TREC 2007 data set, we needed a processing method to convert text data to numerical values relevant to the support vectors machine. The established method to extract numerical features from text is to use a dictionary of words and count the number of times these words appear in the document. The dictionary containing the words that will be counted is loaded, and the documents are divided into words. When a word from the dictionary is found, the corresponding components of a vector used to represent the document is incremented. The result is passed to the support vector machine for classification. In this case, the dictionary of terms is the complete English dictionary with all possible forms of basic terms totaling over 58000 words. Thus, the resulting data belongs to a 58000-dimensional space.

The first approach for classification using the TREC 2007 set used cross-validation with classical heuristics on a relatively small subset of data. We chose, for the first experiment, 1000 legitimate messages and 1000 spam messages for the training set, and the same numbers for the test set. After running the algorithm over all values for parameters using the heuristics suggested above, we achieved 100% accuracy for training data suggesting a perfect classification, but running the trained classifier on the test data resulted in an accuracy of only 64%, which suggested the over-fitting of the train data and a loss of generality.

A first finding was that the $\gamma$ parameter had a negative impact on classification accuracy, a higher value of this parameter resulted in a lower accuracy of the classification regardless of the value of $C$. Therefore, we decided that for the following experiments $\gamma$ would have a very low value and iterate only the values of parameter $C$. It was also noticed that most classification errors were caused by false-positives, i.e. the classification of legitimate messages as spam.

False-negative results, the wrong classification of spam as legitimate e-mail, were also found. This error occurred with very long text message that apparently determined the support vector machine to ignore the fragments of text that would normally identify a message as spam.

The conclusion drawn from these results was that there were too few training data to construct a sufficiently general model. Therefore, we decided to increase the number of examples to 3000 legitimate messages and 3000 spam messages, for both the training set and the test set.

New methods to extract the numerical features of the text were used, as suggested by Sculley and Wachman [10]. They rely on a dictionary of terms like the word counting method, but instead of using a dictionary of complete words they use dictionaries of word fragments of 3 or 4 characters. These dictionaries have been built from the complete English dictionary by extracting fragments of 3 or 4 characters and adding them to the new dictionary.

Thus, we obtained 26485 words for the dictionary of 4-character words and 5776 terms for the dictionary of 3-character words. The vectors representing the documents are built by parsing and counting the occurrences of words from the dictionary in the document. This is done similarly as for the
method in which we use the complete English dictionary. The document is divided into its component words, but instead of looking up the word in the dictionary as in the classical method, the word is divided into 3 or 4 character fragments which will be looked up in the dictionary. At first glance this method requires more time because each word is divided into 3 or 4 fragments of all possible characters which are then looked up in the dictionary. However, this is not true. Although every word with more than 3 or 4 characters is divided into all the possible fragments, the words with fewer characters than those present in the dictionary are ignored, and thus the time required for the parsing stage remains about the same as the time needed for the classical method.

Intuitively, we could think that the computing time for a training method that uses 4-character dictionary words to be lower than the time required by the classical method, and the time required for training using a dictionary of 3-character words should be less than the time required in the 4-character case. However, the opposite is true.

A very advantageous aspect in working with support vector machines on data vectors in a space with a large number of dimensions is that the values of many dimensions of a vector are 0, which greatly reduces the computing time. For example, when using the complete dictionary of English words, it has 58000 terms, therefore the number of dimensions is 58000. But an e-mail contains an average of 100 to 1000 different words, which means that the vector representing the document has the 0 value on more than 75% of its dimensions. By decreasing the number of terms, and implicitly the number of dimensions as it is the case when using dictionaries of 3 or 4 characters, one can greatly increase the number of dimensions with values different from 0 for the vector representing the document.

More specifically the method which was found to have the shortest computing time was the one that used the full English dictionary, and the most inefficient in terms of computing time was the method that uses the dictionary of 3-character terms.

4.2. Results

For simplicity we use the following convention, the method that uses the complete English dictionary will be called “Full Words”, the method that uses a dictionary consisting of terms of 4 characters will be called “4-Grams”, and the method that uses a dictionary consisting of terms of 3 characters will be called “3-Grams”.

For the new data set built on the basis of TREC 2007 from 3000 spam messages and 3000 legitimate messages we used a series of successive trainings with the training set and tests with the test set using several values for the $C$ parameter.

The first method used was “Full Words”, and its results are presented in Fig. 1. There were successive trainings using the full set of training data, incrementing the $C$ parameter from 150 to 950 with a step of 50.
The following experiment was conducted for the same data set using the “4-Grams” method incrementing over values from 150 to 950 with a step of 50 for the $C$ parameter. The results are shown in Fig. 2.

A final experiment was performed using the “3-Grams” method on the same set of data, but incrementing the value of $C$ from 20 to 180 with a step of 20. The results are shown in Fig. 3.

From all these graphs it can be seen that in order to achieve a high classification accuracy the value of the $C$ parameter must be large. It means that the margin of the decision limit of the support vector machine has to be “flexible.” It can also be seen that there is a rise in the classification accuracy with increasing the value of $C$, after which the precision is constant and then a decline follows. It is less steep than the initial growth, but noticeable.

In addition, there was a difference in execution times corresponding to the machine that used the 3-feature extraction methods. Training took 10 min on average using the “Full Words” method, 12 min using the “4-Grams” method and 15 min with “3-Grams” method.
Since we wanted to get better results, a higher classification accuracy and a more general model for the support vector machine, we decided to increase the training data set to 5000 legitimate messages and 5000 spam messages. The size of the testing set was increased to the same values.

The first experiment was performed with “Full Words” method, covering the range from 50 to 950 with a step of 50 for the value of $C$. The results are shown in Fig. 4.

Fig. 4 shows a cap on a fairly large range of $C$ values without a significant decrease. Since we anticipated similar results for the “4-Grams” method, we decided to decrease the step to 20 and based on the results of the experiments carried out on the set of 6000 messages we decided to establish the interval between 20 and 800. The results are shown in Fig. 5.
We performed an experiment on the same data set using “3-Grams” with the same step for iterating over the $C$ parameter, but considering the sharp decrease observed in the experiment presented in Fig. 3, we decided to limit the maximum value to 400. The results are shown in Fig. 6.

This last set of experiments also shows the capping and slight decrease of the accuracy for large values of the $C$ parameter. As in the experiments conducted on the data set with 6000 messages the method used for determining the numerical characteristics of a document based on the complete dictionary of English words gives better results than the other methods. In addition, the average computing time of a training cycle using “Full Words” increased to 20 min for the dataset with 10000 messages, while for method “4-Grams” it increased to 30 min and 40 min for the “3-Grams” method. The tests were made on a computer equipped with an Athlon AM2 4400 processor, dual core, each with 2200 MHz frequency, and 2 GB of RAM.

Given the noticeably better results obtained using the “Full Words” method and shorter computing time for a training cycle when compared with the
other methods, we chose to use only this method for the final experiment. The training data set for the last experiment was increased to 10000 legitimate messages and 10000 spam messages, the same changes being made on the test set. The iterations for the \( C \) parameter were made in the interval 50 to 2000 with a step of 50. The results are shown in Fig. 7.

![Figure 7: Results for the “Full Words” method on a data set of 20000 messages.](image)

This experiment also shows same capping of the accuracy for large values of the \( C \) parameter but a slight decline after reaching the maximum of 94% at 1550. The computing time for a training cycle with the data set of 20000 messages was 45 min on average.

Therefore, the final model of the support vector machine for the spam detection agent is the one trained on the dataset of 20000 messages with the \( C \) parameter equal to 1550. It has a very good accuracy and taking into account the large amount of data that was used, it can be considered general enough to be used in real applications.

Table 1 shows an overview of the three methods used to build the numerical characteristics of text documents, for an easier comparison.

<table>
<thead>
<tr>
<th>Text processing method</th>
<th>Accuracy for 6000 training instances</th>
<th>Accuracy for 10000 training instances</th>
<th>Accuracy for 20000 training instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Words</td>
<td>0.9275</td>
<td>0.9335</td>
<td>0.9401</td>
</tr>
<tr>
<td>4-Grams</td>
<td>0.8987</td>
<td>0.9155</td>
<td></td>
</tr>
<tr>
<td>3-Grams</td>
<td>0.9061</td>
<td>0.9034</td>
<td></td>
</tr>
</tbody>
</table>

The columns indicate the total number of e-mails used for training and the lines mark the methods used to build the numerical characteristics of the messages. It can clearly be seen that the “Full Words” method based on the
complete dictionary of the English language is better than other methods based on dictionaries that use words formed of fragments of 3 or 4 characters.

Another important aspect to be noted is that for the last experiment the amount of training data was doubled, but the accuracy increased by only 0.5%. Therefore, we can say with a high degree of certainty that further increasing the number of messages of the training set will not significantly improve the classification accuracy.

Table 2 presents the average execution times of a training cycle for the experiments performed.

<table>
<thead>
<tr>
<th>Text processing method</th>
<th>Training time for 6000 instances</th>
<th>Training time for 10000 instances</th>
<th>Training time for 20000 instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Words</td>
<td>10</td>
<td>20</td>
<td>45</td>
</tr>
<tr>
<td>4-Grams</td>
<td>12</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>3-Grams</td>
<td>15</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>

5. Conclusions

Support vector machines are one of the best classification methods currently available (arguably the best). They are successfully applied in industry and research in areas such as text classification, handwritten character identification and gene classification based on protein sequences. SVMs have a solid mathematical background and they can produce very good results if properly used. This article shows that SVM can be used to obtain good results in the constantly challenging field of filtering unwanted messages, or spam. The results were very good with all three presented methods, but the best was obtained using the complete dictionary of the English language in terms of both classification accuracy, 94%, and in terms of computing time.

Acknowledgements. This work was supported by CNCSIS-UEFISCSU, project number PNII-IDEI 316/2008, Behavioural Patterns Library for Intelligent Agents Used in Engineering and Management.

Received: July 8, 2010 "Gheorghe Asachi" Technical University of Iași, Department of Computer Engineering e-mail: fleon@cs.tuiasi.ro

REFERENCES


AGENT DE FILTRARE A MESAJELOR DE E-MAIL BAZAT PE MAŞINI CU VECTORI SUPORT

(Rezumat)

Filtrarea mesajelor de e-mail a devenit în ultimul timp o problemă importantă datorită popularității în continuă creștere a comunicării prin intermediul poștei electronice. De aceea, există o nevoie constantă de a îmbunătăți detecția mesajelor nesolicitate, a spam-ului. Multii cercetători au aplicat tehnici de învățare automată pentru filtrarea mesajelor spam iar acestea s-au dovedit încuminate de succes. În acest articol se prezintă un agent de detecție a spam-ului bazat pe mașini cu vectori suport (engl. “support vector machines”, SVM), una din cele mai bune metode de clasificare disponibile în prezent. Se testează câteva metode de extragere a trăsăturilor numerice din documentele text și se evaluatează valorile optime ale parametrilor SVM necesare pentru această problemă de clasificare. Cele mai bune rezultate indică o precizie foarte bună a clasificării, de 94%.