Spam classification using Support Vector Machines

Abstract:

Many email services today provide spam filters that are able to classify emails into spam and non-spam email with high accuracy. The goal of this project is to build a spam filter using SVMs.

Objective: To train a classifier to classify whether a given email, x, is spam (y = 1) or non-spam (y = 0). Before training the model, each email is converted into a feature vector $x \in R^n$. The dataset used for this project is a subset of the SpamAssassin Public Corpus. Only the body of the email (excluding the email headers) is used to build the model in this project.

1. Preprocessing emails

Before starting on a machine learning task, it is usually insightful to take a look at examples from the dataset. The figure given below shows a sample email that contains a URL, an email address (at the end), numbers, and special characters. While many emails would contain similar types of entities (e.g., numbers, other URLs, or other email addresses), the specific entities (e.g., the specific URL or specific special characters) will be different in almost every email. Therefore, one method often employed in processing emails is to "normalize" these values, so that all URLs are treated the same, all numbers are treated the same, etc. For example, we could replace each URL in the email with the unique string "httpaddress" to indicate that a URL was present.

> Anyone knows how much it costs to host a web portal ?
>
Well, it depends on how many visitors you're expecting. This can be anywhere from less than 10 bucks a month to a couple of $100. You should checkout http://www.rackspace.com/ or perhaps Amazon EC2 if youre running something big..

To unsubscribe yourself from this mailing list, send an email to:
groupname-unsubscribe@egroups.com

This has the effect of letting the spam classifier make a classification decision based on whether any URL was present, rather than whether a specific URL was present. This typically improves the performance of a spam classifier, since spammers often randomize the URLs, and thus the odds of seeing any particular URL again in a new piece of spam is very small.

In the attached file processEmail.m, following email preprocessing and normalization steps are implemented :

- Lower-casing: The entire email is converted into lower case (e.g., DATA is converted data).
- Stripping HTML: All HTML tags are removed from the emails. Many emails often come with HTML formatting; we remove all the HTML tags, so that only the content remains.
- Normalizing URLs: All URLs are replaced with the text “httpaddress”.
- Normalizing Email Addresses: All email addresses are replaced with the text “emailaddress”.
- Normalizing Numbers: All numbers are replaced with the text “number”.
- Normalizing Dollars: All dollar signs ($) are replaced with the text “dollar”. In many spam mails, often dollar sign is used to specify a discount coupon or indicationg about a lottery.
- Word Stemming: Words are reduced to their stemmed form. For example, “discount”, “discounts”, “discounted” and “discounting” are all replaced with “discount”.
- Removal of non-words: Non-words and punctuation have been removed. All white spaces (tabs, newlines, spaces) have all been trimmed to a single space character.

The result of these preprocessing steps is shown in the figure below.

anyon know how much it cost to host a web portal well it depend on how mani visitor your expect thi can be anywher from less than number buck a month to a coupl of dollarnumb you should checkout httpaddr or perhap amazon ecnumb if your run someth big to unsubscrib yourself from thi mail list send an email to emailaddr

2. Selecting the words to use:

After preprocessing the emails, we have a list of words for each email. The next step is to choose which words we would like to use in our classifier and which we would want to leave out.

The most frequently occuring words have chosen as our set of words considered. Since words that occur rarely in the training set are only in a few emails, they might cause the model to overfit our training set. The complete list of the words that are choosen for building the model are in the file vocab.txt. Our vocabulary list was selected by choosing all words which occur with a frequency greater than 100 in the spam corpus, resulting in a list of 1899 words.

3. Feature extraction

Feature extraction is implemented to convert each email into a vector of length 1899. Specifically, the feature $x_i \in \{0, 1\}$ for an email corresponds to whether the i-th word in the dictionary (vocab.txt) occurs in the email. That is, $x_i = 1$ if the i-th word is in the email and $x_i = 0$ if the i-th word is not present in the email.

Thus, for a typical email, this feature would look like:

$$x = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^n.$$

The attached file emailFeatures.m contains the code to generate a feature vector for an email, given the word indices.

4. Training SVM model

After completion of feature extraction, the next step is to train a SVM classifier on a training dataset. The data is loaded from the file spamTrain.mat. The training data set contains 4000 examples and the test data set contains 1000 examples. Each original email was processed using the processEmail and emailFeatures functions and converted

into a vector $x^{(i)} \in R^{1899}$. The SVM classifier was trained using a simplified version of the SMO (Sequential Minimal Optimization) algorithm.

The classifier is then tested on the training and testing datasets to get a training accuracy of about 99.8% and a test accuracy of about 98.5%.

6. FInding top predictors of spam

To better understand how the spam classifier works, we can inspect the parameters to see which words the classifier thinks are the most predictive of spam. The next step is to find the parameters with the largest positive values in the classifier and display the corresponding words. Thus, if an email contains words such as “guarantee”, “remove”, “dollar”, and “price”, it is likely to be classified as spam.

The words which are marked as top predictors of spam are shown below.

```
our click remov guarante visit basenumb dollar will price pleas nbsp
most lo ga dollarnumb
```