The spam vs. non-spam e-mails classifier
A real-life case study about using PolyAnalyst™ for building high precision classification models

A white paper written for Megaputer Intelligence Inc. by
Valentin Militaru
Contents

General Context and Industry Problem ......................... 3
Task Definition ............................................................ 3
Some Methodological Considerations ......................... 4
Data Mining: Building the Classification Model .......... 4
  Data Inspection and Preprocessing ....................... 4
  Algorithm Selection ............................................. 6
  Model Building (Data Mining) ............................ 6
  Assessment of the Results ................................. 7
Conclusions ............................................................. 7
Further Applications ................................................. 8
Acknowledgements .................................................. 9
The spam vs. non-spam e-mails classifier

General Context and Industry Problem

The present case study is based upon a real-life situation, which generated the task for the Data Mining Cup 2003 Contest (www.data-mining-cup.de) that took place in Chemnitz, Germany.1

The problem of unrequested received e-mails, most of them with doubtful content, is well-known. It is estimated that each day, about 25 Mio. of unwelcome emails (so-called spam mails) are sent. This corresponds to 10 percent of all emails worldwide. Performed investigations with the employees of many enterprises turned out that, on average, 40 percent of their daily received emails are spam. With some of the employees, this share reaches up to 90 % of the received e-mails.

In spite of corresponding laws, an effective legal prosecution is nearly impossible. The receiver faces thus the problem of separating the important emails from the spam mails which are often not recognizable at the first sight. Since this is a time-consuming process, in the last years, the wish to recognize the spam emails and to select them automatically has motivated different producers from the software industry to build dedicated applications, equipped with more or less “artificial intelligence”, that should automate the classification task. The efficiency of these solutions varies widely and the key component of any good email filter is its ability to classify emails on the basis of different measured and synthesized attributes and thus to correctly assign spam and non-spam emails with a high probability.

Task Definition

During a program aiming at the optimization of the communication process, the management of a medium-size German company observed that an exceedingly large part of all incoming emails are advertisement mails. The large amount of working time spent by the 120 employees for the daily sorting and final erasure of the spam emails has unveiled a high potential of rationalization.

For this reason, all emails of the company have been collected over some time and stored separately by spam and non-spam emails. Every email was described by a set of attributes.

The task was to build a classification model for precisely identifying spam mails. In this respect, two datasets were considered:

a) a training dataset, containing information about 8,000 e-mails;

b) a dataset of 11,177 rows, upon which the classification model should have been applied.

The objective was to minimize the number of the spam emails which have passed the filter, subject to the constraint that within the filtered emails only, a maximum of 1% non-spam emails were allowed.

As result, a list had to be created, containing the IDs of those emails which have passed the filter (i.e. all emails which have been recognized as potential non-spam).

1
Some Methodological Considerations

Many data mining techniques implemented by PolyAnalyst could have been taken into consideration for solving this task: k-Nearest Neighbor, Artificial Neural Networks, Decision Trees, etc...

With its complex base of data mining engines, Poly Analyst can address a highly diverse suite of problems. Intuitively, classification tasks can be easily approached with the Memory Based Reasoning engine, that implements a k-Nearest Neighbor algorithm, or with the Decision Tree/ Decision Forest engines, etc... Nevertheless, it is known that by data preprocessing, any classification task can be transformed in a prediction task and vice versa. Therefore, one might assume that Stepwise Linear Regression engine or the Artificial Neural Networks engine, that implements the PolyNet Predictor algorithm can be employed too. The logic of selecting the appropriate data mining algorithm is presented later in this whitepaper.

Still, an important observation is that the analyst deals with a supervised classification problem, therefore cluster-based or association analysis engines are practically useless for model building.

On the other hand, solving the task imposed the use of both descriptive and predictive data mining techniques.

The project was developed in 4 steps:

i) data inspection and preprocessing
ii) algorithm selection
iii) model building (data mining)
iv) assessment of the results

Each of these steps will be explained in detail further in this material.

Data Mining: Building the Classification Model

Data Inspection and Preprocessing

Within the Data Mining Cup 2003 Contest, two datasets were provided:

• the training dataset, containing 8,000 records and
• the classification dataset, containing 11,177 records (emails) which had to be classified in spam or non-spam emails.

According to the information provided by the organizers of the Data Mining Cup 2003 contest, both training and classification data were taken from the same sample and, thus, had the same class distribution. Strictly speaking, all data was taken from a sample of a total of 19,177 emails.

The structure of both datasets was similar, with the exception of one categorical field (target) which was present only in the training dataset. Values for the target field were assigned based on past experience (target = "yes" - + record is spam mail, target = "no" — * record is non-spam mail).

Each record of the training dataset was described by 834 attributes, as follows:

• the first attribute (id) was numeric and provided the identification number of the current record in the dataset
• the last attribute (target) was used to classify the dataset in spam and non-spam records
• the other 832 attributes, all of Boolean type, provided various information about the records; these descriptive attributes were coded according to those of the popular open-source project Spam Assassin².

Distribution of values for the "id" attribute for both spam and non-spam mails

The first important information about the data was knowing which attributes were really relevant for the classification of the records in two categories, namely target = "yes" and target = "no". For this purpose, the Summary Statistics engine confirmed that out of 833 Boolean attributes, only 544 were set to "1" at least once in the entire training dataset, therefore, only these were providing useful information in the classification process. The other 289 attributes were constant for all the 8,000 records of the training dataset, therefore their respective columns were accounted as redundant data and were not taken into consideration any further.

Another descriptive technique implemented in Poly Analyst revealed an additional valuable information about the training dataset. The histogram chart was used for building a view for the distribution of values for the "id" attribute. For a clearer illustration, the training dataset was split into two subsets, one containing only spam emails (target = "yes") and one containing non-spam emails (target = "no"). The graphic representation revealed a surprising distribution of values: all the records having an id higher than 384140 were spam e-mails (target = "yes"). Therefore, the id attribute had to be taken into consideration when building the classification model.

These preliminary analyses demonstrated that no data transformation was required; data pre-processing effort was therefore reduced to selecting the attributes with minimum of relevance to the classification process (544 descriptive attributes plus the id field).
Algorithm Selection

As mentioned before, Poly Analyst offers a powerful base of data mining engines. Yet, the following characteristics of the challenge had to be considered when choosing the appropriate algorithm:

- the algorithm should have handled very well datasets with large number of attributes
- it should have been able to work with both Boolean and numeric attributes
- it should have been able so find a nonlinear solution
- the classification model should have been highly precise
- the algorithm should have taken into consideration all the attributes indicated by the analyst, since all the 544 + 1 attributes were identified as relevant in the previous step.

According to the above-mentioned requirements, the only algorithm suitable for the data mining process was Decision Tree. As stated in Poly Analyst's documentation, the DT algorithm is well poised for analyzing very large databases and its calculation time scales very well (grows only linearly) with increasing number of data columns. Although k-Nearest Neighbor might also have looked like a potential solution, its computation time is proportional to the square of the number of records. Thus this algorithm is not designed for exploring large datasets.

Relative to the high number of attributes, it is important to notice that Poly Analyst, unlike other data mining tools, can handle datasets with more than 255 columns. This ability proved to be vital for this particular application.

Model Building (Data Mining)

The use of the DT algorithm on the training dataset lead to a classification model (tree) with the following characteristics:

- Number of non-terminal nodes: 47
- Number of leaves: 48
- Depth of constructed tree: 19

As expected, the DT rule, based on such a large number of attributes, generated a very complex tree structure, difficult to represent completely as a graph.

In order to verify the accuracy of the model, the DT rule was tested on the training dataset (with all the 834 attributes selected), with the following results:

<table>
<thead>
<tr>
<th>Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total classification error:</td>
<td>0.90%</td>
</tr>
<tr>
<td>% of undefined prediction cases:</td>
<td>0.00%</td>
</tr>
<tr>
<td>Classification probability:</td>
<td>99.10%</td>
</tr>
<tr>
<td>Classification efficiency:</td>
<td>97.69%</td>
</tr>
<tr>
<td>Classification error for class &quot;Non-spam&quot;:</td>
<td>0.86%</td>
</tr>
<tr>
<td>Classification error for class &quot;Spam&quot;:</td>
<td>0.96%</td>
</tr>
</tbody>
</table>

According to this test, the situation of real vs. predicted cases was satisfactory, only 42 out of 4,888 non-spam mails were misclassified (0.86%) and only 30 out of 3,112 spam mails were accounted as non-spam (0.96%). The overall test error was therefore 72/8,000 cases, that is 0.9%.
Considering the total error as reasonable (0.9%), the DT rule was applied to the classification dataset, which lead to the selection of 6753 records (out of 11,177) that were regarded as non-spam emails.

### Assessment of the Results

The evaluation of the model followed a procedure which is typical for every contest. The jury knew which of the 11,177 emails to be classified were actually non-spam emails or spam emails. This information was later available to the participants and thus it became possible the testing of the model against real data.

The following table reflects the model's efficiency:

<table>
<thead>
<tr>
<th>Real/ Predicted cases</th>
<th>Non-spam</th>
<th>Spam</th>
<th>Undefined</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-spam</td>
<td>4,846</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>Spam</td>
<td>30</td>
<td>3,083</td>
<td>0</td>
</tr>
</tbody>
</table>

Going back to the definition of the task, the project's objective was to minimize the number of spam emails which have passed the filter. The actual number of non-spam emails in the classification dataset was 8,803. Only 54 (0.79%) spam mails incorrectly passed the filter and were misclassified as being non-spam mail. Still, there were 104 non-spam mails that the model considered to be spam. In terms of percentage, the model did not recognize 1.52% of the non-spam emails.

To conclude these calculations, testing the model on the classification dataset proved that its real precision was of 98.59%, with only 158 misclassified records out of 11,177.

### Conclusions

Most important, this case study adds up evidence about PolyAnalyst's successful application in real-life projects. Its efficiency can be summarized by two aspects: speed and accuracy.

As presented in Megaputer's technical documentation, "Decision Trees is PolyAnalyst's fastest algorithm when dealing with a large amount of records and/or fields". Running on a Pentium III at 650 MHz with only 64 MB of RAM, the classification model took less than 1 hour to build with a demo version of PolyAnalyst 4.5 under Windows 98. Using the same 8,000 records dataset, but this time on a Pentium III at 650 MHz with 128 MB of RAM, the time was reduced to 20 minutes, for a licensed version of PolyAnalyst 4.6, under Windows XP. Applying the model on the 11,177 rows dataset was only a matter of seconds.
The spam vs. non-spam e-mails classifier

As regards the accuracy of the model, Poly Analyst's decision tree algorithm proved its capacity to build high precision classification models. From a dataset of 11,177 records, only 158 were misclassified, which represents a global error of than 1.4%. Such level of precision is critical in some areas of application (such as medicine) and a huge advantage in others, like finance or CRM.

Yet, a third aspect deserves to be mentioned in our conclusion: the software's ability to handle large volumes of data. This characteristic was critical in our example, since the training dataset contained more than 6.6 Mil Boolean records and the total size of the project reached 16 Mil Boolean records (835 columns x 19,177 rows).

Further Applications

Such data analysis model can be further replicated in many other areas of interest. Classification tasks are widely encountered in data mining projects, some of the most common fields being:

*Customer Relationship Management*: which prospects are most likely to respond to a new promotion or which customers are about to switch to a competitor?

*Fraud detection*: which transactions are most likely to be fraudulent?

*Retail/point-of-sales analyses*: which location is most appropriate for a future store?

*Portfolio analysis*: which securities/ assets would fit a predefined portfolio at a certain point in time?

*Medicine*: what diagnostic or medication is suitable for a complex of symptoms?
Acknowledgements

The author wishes to thank Mr. Richard Kasprzycki and Mr. Hrishikesh Pore from Megaputer Intelligence Inc for their confidence and invaluable contribution to the writing of this material. Also, the author congratulates the organizers of the DMC 2003 contest and thanks Mrs. Sandra Homke and Mr. Jens Scholz from Prudsys A.G. for their acceptance to further using the data provided in the contest.

Notes:

1 The Data Mining Cup contest (www.data-mining-cup.de) is a competition that is organized every year at Chemnitz, Germany, by the Chemnitz University of Technology (www.tu-chemnitz.de), Prudsys AG (www.prudsys.de) and the European Knowledge Discovery Network of Excellence (www.kdnet.org). The contest is addressed to students of universities, advanced technical colleges as well as universities of cooperative education from Germany and abroad. In 2003 there were 514 registered participants, from 199 universities and colleges around the world (39 countries), out of which only 156 found a solution to the problem.

2 SpamAssassin(tm) is a mail filter to identify spam. Using its rule base, it uses a wide range of heuristic tests on mail headers and body text to identify unsolicited commercial email. For more information see http://spamassassin.org