A Computational Geometric Approach on Rule Based SPAM Filtering

MARIOS POULOS[*] NIKOLAOS KORFIATIS[†]
SOZON PAPAVLASSOPOULOS[*]

Department of Archive and Library Sciences
Ionian University [*]
Corfu, Greece

Department of Management Science and Technology
Athens University of Economics and Business[†]
Athens, Greece

Abstract

In this paper we address a method for categorizing SPAM messages using categorization rules. Rules that categorize a message as SPAM cannot address the issue of semantic paraphrasing that is a common method to bypassing text filters. Our approach which is based on a computational geometry algorithm works by making a representation of SPAM terms and message subsets using convex polygons and examining the correlation of these polygons in the plane. We argue that this extension can be very efficient in the context of SPAM classification.

Key-Words : Computational Geometry, SPAM , Text Categorization

1 Introduction

With the rigorous development of internet infrastructures, e-mail has emerged as the primary mean for computer mediated communication. Several types of internet users are using e-mail in order to accomplish several tasks varying from simple communication inquires through complex social processes such us shopping or coordination of activities (e.g in the context of a workgroup). From the other side the zero cost of e-mail and the affect rate that it has to it's users have mutated it to an excellent (in terms of cost) tool for untargeted advertisements expressed via unwanted e-mail messages that are received in a daily bases from every user of an e-mail software client. This privacy issue is a raising phenomenon affecting all users of e-mail and posses a serious threat to the use of e-mail as a mean for computer mediated communication. In order to address the above manner several text categorization techniques have been developed in order to help users having automatically classified their e-mail messages according their preferences.

A popular type of mail classification is "rule based filtering"[1]. This family of methods which requires a type of interaction between the user and software can also be used to classify the unsolicitated messages according to the user perceptions regarding the message content.

We propose an extension of these methods to address the problem of grammatic paraphrasing of message subsets that are responsible for inaccurate results of rule based SPAM Filters by using a geometrical representation of the subset encoding and correlating the convex polygons constructed.

2 Rules as a Method for e-mail Classification

In the field of text categorization a rule can be seen as the formal expression of the classification function which is in the boolean form $\Phi : S \times C \rightarrow \{True, False\}$. A vector $\vec{S}$ represents the input subset, in our case the subset of the mail message, such us $\vec{S} = \langle S_1, S_2, \ldots, S_n \rangle$. The approximation of the above function categorizes the contents of the input vector in the categories C_i which reside in the category vector $\vec{C}$ such us $\vec{C} = \langle C_1, C_2, \ldots, C_n \rangle$ The classification function also poses a rule vector $\vec{P}$ also known as a constrain vector such us $\vec{P} = \langle P_1, P_2, \ldots, P_n \rangle$. Each element P_i of the rule vector addresses a constrain expressed via a type of a logical operator. In our case the subset κ_i of the mail message S_i can be represented in a rule P_i having the form: $P_i : \kappa_i \mapsto \lambda$ where λ represents a set of logical operators expressed in the form: equal/not-equal, contain/not contain etc In modern e-mail software clients such us Outlook and Mozilla [2] a rule engine is implemented and a kind of interaction with the user is required in order to construct the rule vector. Having implemented a rule engine means that for every rule element P_i that is defined by the user, the software makes a lookup to the subset κ_i of the input message S_i that is targeted when the approximation of the function Φ is true or false. In case the proximity is true an action vector $\vec{\alpha}$ is been used in order to define the actions that are required by the software to be accomplished after the rule makes the approximation of Φ as true. In a typical e-mail client the interaction that is required by the user can be divided in the following steps:

1. The user defines a specific subset κ_i of the mail message as a part of the input vector $\vec{S_i}$. This can be the title of the mail message, the body, the sender or the recipient of the message etc.

2. For the specific subset κ_i a rule has to be constructed in order to run as an input to the classification function. Usually the user defines a type of logical operator expressed in terms mentioned above

3. Having constructed the rule the user also has to define an action or a series of actions which address the software behavior when the proximity of the classification function used by the rule engine is true.

The simplicity that a modern software client addresses when a user interacts with it in order to create a rule is critical for the success of this kind of filtering method. We have reviewed a generic use of rules as a method for categorizing e-mail. In the next paragraph we examine the use of rules in the context of SPAM Filtering.

3 Using rules as a SPAM filtering method

SPAM [3] addresses an unsolicitated form of e-mail messages which are in an un-targeted disparse when are being sent to a user. Typically a SPAM message contains several terms which correspond to advertisements related to pornography, health issues or financial information. As the rate of SPAM messages is growing in a micro-level measure ,e.g. expressed via the amount of SPAM that a typical e-mail user receives in his mailbox, there is a growing need for an automated method of SPAM protection. Since the classification of a mail message is rather a cost-sensitive task the need for an accurate method has also to be addressed. Re-examining message rules as a method for filtering or categorizing mail messages, when putting them in the context of SPAM classification we need to address the following issues:

1. The rule has to be accurate in terms of classifying the message according to the input subset and rule vector. Accurate means that it will categorize messages giving a positive proximity of the categorization function to the appropriate categories.

2. The input vector is needed to have a semantic accuracy in terms of validating the message subset against the appropriate rule.

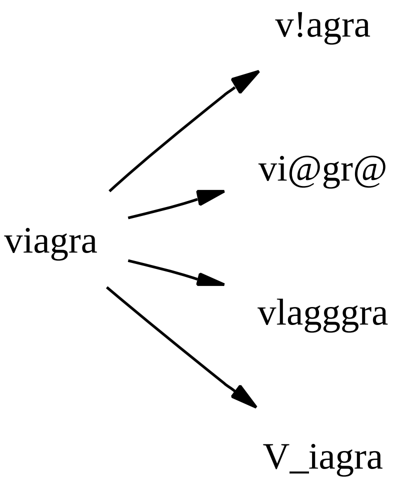

Figure 1: Grammatical Instances of an Input Term discovered in a SPAM corpus

The second issue which is very important comes due to examination of current rule failures appeared in our local computers. The main reason of these failures is the semantic paraphrasing that is used when sending the SPAM message. Semantic paraphrase of a message term means that it is codified not with an accurate grammatic part of a natural language but with symbols that give to the user the same meaning when reading the SPAM message. For example as can be inferred from the Figure 1 a typical term (viagra) in a SPAM message subset can have four or more instances in a SPAM corpus. The above has been inferred from a lexical analysis in a small subset of the spamarchives.org message corpus. In order to address this issue we have extended the typical rule engine by using a method based in a computational geometry algorithm which represents representing an input SPAM term as a convex polygon using the ASCII encoding of the term as the numerical values. Next we construct the appropriate polygon for each subset of the message vector that has to be validated against the initial rule.

Finally we address the following lemma which we will prove in the next paragraph: *one or more sides of the convex polygon representing the message subset are equal and parallel to the convex polygon representing the SPAM term, then the message is categorized as SPAM.* We describe our method in the next paragraph by giving some examples.

```
>>S='This is a message to test the double "command".'
>>double(S)
ans=
 Columns 1 through 12
 84 104 105 115 32 105 115 32 97 32 109 101
 Columns 13 through 24
 115 97 103 101 32 116 111 32 116   101   115   116
  Columns 25 through 36
 32 116 104 101 32 100 111 117 98 108 101 32
  Columns 37 through 46
 34 99 111 109 109 97 110 100 34 46
```

Figure 2: Matlab Example of the double command

4 Extending SPAM Rules using Computational Geometry

4.1 Pre-processing Stage

In this stage we suppose that a selected text is an input vector $\vec{x} = (x_1, x_2, x_3, \ldots, x_n)$, where $x_1, x_2, x_3, \ldots, x_n$ represent the characters of the selected text. Then using a convertion procedure where a symbolic expression (in our case an array of characters of a text) is converted to ASCII characters in string arithmetic values, we obtained a numerical value vector $\vec{S} = (s_1, s_2, s_3, \ldots, s_n)$ where this values ranged between 1-128.

In our case this conversion was achieved by using the *double.m* function of the MATLAB Package. This function converts strings to double precision and equates with converting an *ASCII* character to its numerical representation. For better comprehension we give an example via MATLAB as can be seen from the Figure 2.

4.2 Processing Stage

Our method is based on the following proposition: We considered that the set of elements of vector $\vec{S}$ for each selected text contains a convex subset, which has a specific position in relation to the original set [4],[5]. This position may be determined by using a combination of computational geometry algorithms, which is known as Onion Peeling Algorithms [6] with

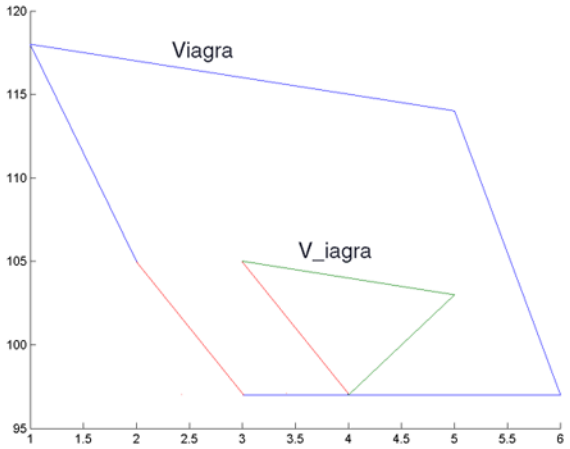

Figure 3: A matching case where the input subsets have an equal and parallel side in their convex polygons

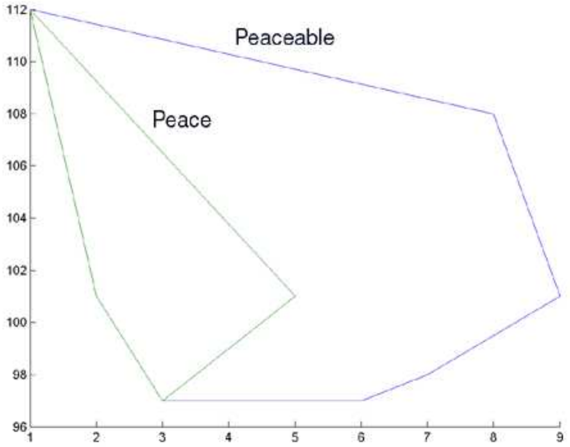

Figure 4: Terms with the same grammatical root have a common side in their convex polygons

overall complexity $O(d \cdot nlogn)$ times, where d is the depth of the smallest convex layer and n is the number of characters in the numerical representation (according to the previous section).

Thus the smallest convex layer $\vec{S}_x$ of the original set of vector $\vec{S}$ carries specific information. In particular, vector $\vec{S}_x$ may be characterized as a common geometrical area of all the elements of vector $\vec{S}$. In our case, this consideration is valuable because this subset may be characterized as representing the significant semantics of the selected text.

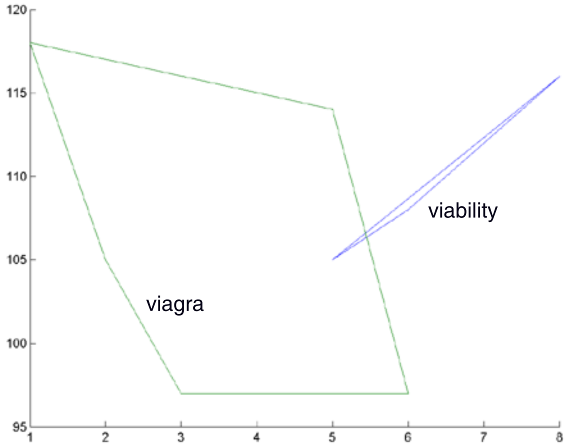

Figure 5: Subset terms with different semantic instance have different positioning in the plane

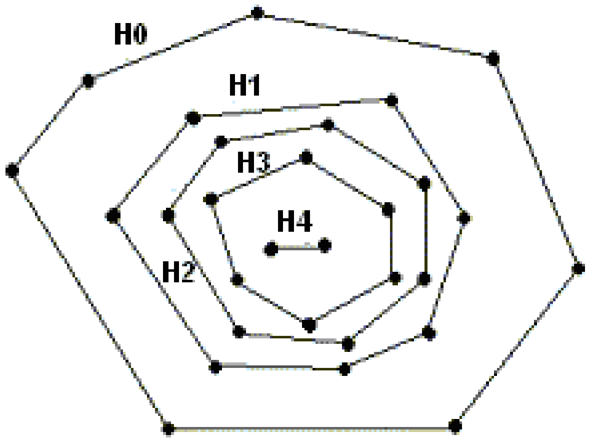

Figure 6: Onion Layers of a set of points

4.3 Implementation

We consider the set of characters of a selected text to be vector $\vec{S}$. The algorithm starts with a finite set of points $S = S_0$ in the plane. The following iterative process is considered. Let S_1 be the set $S_0 - \partial H(S_0)$: S_0 minus all the points on the boundary of the hull of S_0. Similarly, define $S_{i+1} = S_i - \partial H(S_i)$. The process continues untill the set is ≥ 3 (see Figure 6). The hulls $H_i = \partial H(S_i)$ are called the layers of the set, and the process of peeling away the layers is called onion peeling for obvious reasons.

5 Conclusions and Future Work

The work presented is an ongoing research in progress work. Currently we are developing a rule engine that will support our classification operator by extending available software such us Mozilla's rule engine. The next step is to validate the accuracy of the proposed operator by conducting experiments of realistic uses of the rule engine.

References

[1] Cohen W. Learning Rules that Classify E-Mail. In *AAAI Spring Symposium on Machine Learning in Information Access*. AAAI Press, Menlo Park, California, 1996.

[2] Mozilla SPAM Filtering. http://www.mozilla.org.

[3] Cranor L.F. and LaMacchia B.A. Spam! *Communications of ACM*, vol. 41(8), 1998, pp. 74–83.

[4] Poulos M. and Rangoussi M. Parametric person identification from the EEG using computational geometry. In *Sixth International Conference on Electronics, Circuits and Systems ICECS'99*. IEEE, Pafos, Cyprus, September 1999.

[5] Poulos M., Magkos E., Chrissikopoulos V., and Alexandris N. Secure Fingerprint Verification based on image Segmentation using Computational Geometry Algorithm. In *Proceedings of IASTED International Conference Signal Processing Pattern Recognition & Application*, 2003.

[6] Bose P. and Toussaint G. No Quadrangulation is Extremely Odd. In *6th International Symposium on Algorithms and Computation (formerly SIGAL International Symposium on Algorithms)*, 1995.