International Journal of Sciences: Basic and Applied Research (IJSBAR)

ISSN 2307-4531
(Print & Online)

http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

A 2-means Clustering Technique for Unsupervised Spam Filtering

Kostas Fragos*

Associate Researcher, Technological Institute of Athens, Dep. Of Computer Science, Ag. Spyridonos, 12210, Athens, Greece

Email: kfragos@sch.gr

Abstract

Unsolicited commercial e-mail, or "Spam", implies a waste of network bandwidth and waste of human effort in internet and mobile phones communication. It is also a hard problem to distinguish legitimate from spam emails. The majority of the proposed algorithms use supervised learning techniques. Unfortunately, these approaches have the drawback of training over a large amount of manually and costly tagged email corpora. In this paper, we present an unsupervised method to address the problem of filtering spam emails without the need of training over such corpora. Using a 2-means clustering technique we perform a 2-way classification. To overcome the serious complications imposed by the large dimensionality of the data, the algorithm first transforms the data into a low dimensional component space applying a Principal Component Analysis over the data and then performs clustering on them. The method was proved to be promising when evaluated over the publicly available corpus, called "SpamAssasin", which is provided by the Open Project for evaluation purposes. The achieved performance is comparable to the performance of systems based on supervised learning techniques.

Keywords: Spam filtering; 2-means clustering; principal components analysis; feature selection.

1. Introduction

Unsolicited commercial e-mail, or "Spam", has emerged as a serious problem related to the waste of network bandwidth and waste of human effort to pick out the useful message from a "pile of garbage".

---

* Corresponding author.

Two classes of methods have been shown to be useful for classifying e-mail messages and solving the Spam problem. The rule based method, which uses heuristic rules to classify e-mail messages and the statistical based approach, which models the difference of messages statistically. Kim and his colleagues. [1] present a review of the currently available methods on spam detection techniques, filtering, and mitigation of mobile SMS spams. In their work they try to propose light and quick algorithm through which SMS filtering can be performed within mobile devices independently. More concretely, they propose a FR (Frequency Ratio) measure for evaluating lightness and quickness of filtering methods so that SMS filtering can be performed independently within mobile devices. The existing research literature is critically reviewed and analyzed. The most popular techniques for SMS spam detection, filtering, and mitigation are compared, including the used data sets, their findings, and limitations. Statistical filters automatically learn and maintain rules and easily adapt to the new circumstances when new data arrives. The most popular and effective statistical spam filter is the naïve-Bayes one. Reference [2] examines the effectiveness of statistically-based approaches Naïve Bayesian anti-spam filters, as it is content-based and self-learning (adaptive) in nature. Learning algorithms that uses the Naive Bayesian classifier have shown promising results in separating spam from legitimate mail. An encoded and fragmented database approach that resembles radix sort technique has been proposed and applied for first time to improve Paul Graham's Naive Bayes machine learning algorithm for spam filtering [3]. Reference [4] created an artificial neural network based on email classifier; He applied a neural network (NN) approach to the classification of spam employing attributes comprised from descriptive characteristics of the evasive patterns that spammers employ rather than the context or frequency of keywords in the messages. However the reported precision was significantly lower than that of other machine learning approaches. Furthermore, support vector machines seem to be more appropriate selection for this type of problem due to the small time they demand for training [5]. Reference [6] applied logistic regression algorithms, and performed a comparison to Naïve Bayes classifier. The results were calculated on their collection of e-mails and were approximately the same. Hence, there was not any reason to substitute Bayesian filtering with genetic algorithms. In this paper, a different approach is adopted using an unsupervised learning technique. Using the well-known k - means clustering algorithm, we perform a 2-way unsupervised classification to categorize an incoming e-mail. First, the incoming data is transformed into a low dimensional space using Principal Components Analysis (PCA). Then, we perform clustering. The paper is organized as follows: In section 2 we present the k - means clustering algorithm. In section 3 the PCA for the reduction of the dimensionality of space is presented in some detail. In section 4 the proposed method is described. Results from the evaluation procedure are presented in section 5, as well as a comparative evaluation with other approaches. Finally, section 5 concludes the paper.

2. K-means Clustering Technique

The well–known Clustering problem could be described as the following situation: there is no class to be predicted and the items, which are drawn from a specific data set (or domain, or data point), are divided into groups (clusters). More precisely, clustering techniques (algorithms) are applied when items have a strong resemblance to one another and hence can be divided into groups (clusters). K-means is a simple and effective unsupervised learning algorithm that solves the clustering problem [7,8,9]. It follows a simple and easy way to

classify a given data set through k - clusters. The main idea is to define k central points, which are called centroids, or means, one for each cluster. These centroids are initially placed each one as far as possible from all the other ones. Then, each point of the data set is associated to the nearest centroid. If no point is pending then the first step is completed and an early classification is done. Then, the recalculation of the new central points is done to specify the k new centroids. A new binding follows between the same data set points and the nearest new centroid and this is repeated with the k centroids to change their location step by step until no more changes are done. The algorithm aims at minimizing an *objective function*, which a squared error function is given by the following formula:

$$C = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_{ij} - c_j \|^2 \qquad (1)$$

Where $\| x_{ij} - c_j \|$ is the distance between the data point x_{ij} and the center of the class c_i.

The following procedure can be used for the k-means clustering.

Start

Place K points (i.e. the initial centroids) into the space represented by the data points.

Consider the set of data points that are going to be clustered.

Repeat

Assign each data point to the group that has the closest centroid.

When all objects have been assigned, recalculate the K new centroids.

Until *the centroids no longer move.*

The algorithm is very sensitive to the initial randomly selected cluster centers. However, it can be proved that the procedure will always terminate. The k-means algorithm does not necessarily find an optimal solution ("configuration"). A simple example will be given to illustrate the procedure.

Suppose that we have vectors ($\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n$) of n sample features, all from the same class. These features fall into k compact clusters, $k < n$. Let p_i be the mean of the vectors in cluster i. If the clusters are well separated, we can use a minimum-distance classifier to separate them. We can define that $\mathbf{x}$ is in cluster i if $\| \boldsymbol{x} - p_i \|$ is the minimum of all the k distances. Hence, the following algorithm can be used:

Start

Select initial values for the means $p_1, p_2, ..., p_k$

Repeat

Use the estimated means to classify the samples into clusters

For i = 1 to *k*

Replace p_i with the mean of all of the samples for cluster i

Until *there are no changes in any mean*

The above procedure is a simple version of the k-means procedure. It can be viewed as a greedy algorithm for partitioning the n samples into k clusters so as to minimize the sum of the squared distances to the cluster centers. The results depend on the metric used to measure $\| \boldsymbol{x} - p_i \|$.

In this work, a 2 - means clustering algorithm has been adapted to solve the spam filtering classification

3. Reduction of Dimensionality Space

The need for reducing the dimensionality of data is common in Natural Language Processing tasks. PCA has been used as a method that reduces data dimensionality by performing covariance analysis [10], [8], [9]. It is suitable for NLP tasks where a significant number of features are included in the data. PCA is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The aim of PCA is to reduce the dimensionality (number of variables) of the dataset but retain most of the original variability in the data. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

Suppose that $e = (f_1, f_2, ..., f_p)$ is a p-dimensional random vector. The *k* principal components of e are *k* (univariate) random variables $h_1, h_2, ..., h_k$ which are defined by the following formulas:

$$
\begin{aligned}
h_1 &= \lambda_1{}^T e = \lambda_{11} f_1 + \lambda_{12} + .. + \lambda_{1p} f_p \\
h_2 &= \lambda_2{}^T e = \lambda_{21} f_1 + \lambda_{22} + \ldots + \lambda_{2p} f_p \\
&\ldots \\
h_k &= \lambda_k{}^T e = \lambda_{k1} f_1 + \lambda_{k2} + \ldots + \lambda_{kp} f_p
\end{aligned}
$$

In the first principal component h_1, the coefficients $\lambda_{11}, \lambda_{12},..,\lambda_{1p}$ are chosen to ensure that the variance

$Var(\lambda_{11} f_1 + \lambda_{12} f_2 + \ldots + \lambda_{1p} f_p) = Var(\lambda_1^T e)$ is maximum and $\| \lambda_1 \| = 1$.

In the second principal component h_2, the coefficients $\lambda_{21}, \lambda_{22},..,\lambda_{2p}$ are chosen to ensure that the variance

$Var(\lambda_{21} f_1 + \lambda_{22} f_2 + \ldots + \lambda_{2p} f_p) = Var(\lambda_2^T e)$ is maximum, $\| \lambda_2 \| = 1$ and the covariance

$$Cov(\lambda_1^T e, \lambda_2^T e) = 0.$$

In the same way, we choose the coefficients $\lambda_{k1}, \lambda_{k2},..,\lambda_{kp}$ for the k principal components h_k to satisfy the following conditions: the variance $Var(\lambda_{k1} f_1 + \lambda_{k2} f_2 + ... + \lambda_{kp} f_p) = Var(\lambda_k^T e)$ is maximum, $\| \lambda_k \| = 1$ and $Cov(\lambda_j^T e, \lambda_k^T e) = 0, \forall j < k$.

In other words, the principal components are those linear combinations of the original variables which maximize the variance of the linear combination and which have zero covariance (and hence zero correlation) with the previous principal components. The PCA is also known as the standard eigenvalue problem with the symbols λ_i in our case to represent the eigenvalues of the covariance matrix of the data.

4. The Proposed Spam Filtering Method

E-mails are represented as vectors of binary features: $e=(f_1, f_2 ,.. f_N)$, where N is the number of features. For a given email, the feature f_j is equal to 1 if the email contains the feature and 0 otherwise. The heart of the system is the maintenance both of a Feature Inventory (*FI*), which defines the space of the problem system, and a Principal Components List (*PCL*), which contains the first 15 principal components of the domain which account for the most variability of the data. When an incoming email $e_i=(f_1, f_2 ,.. f_M)$ enters into the system it is scanned for features and the extracted features are inserted into the *FI*. Then, the e-mail is represented in the feature domain space as $e_i=(f_1, f_2 ,.. f_N)$, where N is the total number of the features in the *FI*. A feature components' analysis is performed over the matrix e, which represents the domain of the so far incoming e-mails and the *PCL* is updated. The domain of the incoming emails e is then converted into the new space specified by the *PCL* components. This is where the 2-means clustering algorithm is applied to classify the space into two categories.

The selected features are extracted from all the available fields of an incoming e-mail. First, we scan the *body* (field) of the e-mail and everything is selected and added to the *FI*. Second, we scan the html code and select features from fields like the following ones: *received_from*, *delivery date*, *message-id*, X-*keywords*, *Content Type*, *subject*, *body*, *size.* Other types of information (features) are also included: Html tags for *fonts* and *colors*, URL's for multimedia resources, (features extracted from) *java scripts* code etc. All those features are extremely useful in the discrimination procedure, so we include them in the feature catalogue. The following algorithm outlines the basic steps involved in the operation:

Start

Read the incoming e-mail.

Extract all the available features.

Represent the e-mail as a vector $e_i=(f_1, f_2 ,.. f_N)$ into the feature domain space.

If *all the extracted features are contained in the inventory*

Project the incoming email into the first principal component space specified by the PCL.

Classify the incoming e-mail using the 2-means clustering algorithm.

Else

Insert the new features into the Feature Inventory (FI).

Represent the so far incoming e-mails e into the updated feature space specified by the FI.

Perform a Principal Components analysis to update the PCL.

Project the incoming e-mail into the updated Principal Component space specified by the PCL.

Classify the incoming e-mail using the 2-means clustering algorithm.

End

The more the Feature Inventory (*FI*) grows the more the need arises for computational resources. To avoid such a situation we must periodically update the *FI* eliminating those features that appear rarely in a small number of e-mails. In this work we used a lower limit of 15 e-mails for the elimination of e-mails.

5. Evaluation

Our experiments have been carried on a publicly available corpus, provided by the Open Project SpamAssasin for evaluation purposes and benchmarking of unsolicited bulk e-mails filters [11]. This is a selection of mail messages, created especially for benchmarking of spam-filtering systems. The most recent collection *20030228_spam_2* has been selected for our experiments. The legitimate part of the collection consists of two sub-collections of e-mails: the *20030228_hard_ham*_2 and *20030228_easy_ham* containing 250 and 2500 non-spam messages respectively. The *hard_ham_2* corpus contains non-spam messages. It is difficult, these e-mails to be discriminated from spam messages. The presence of several features in these messages, use of HTML, unusual HTML markup, colored text, "spammish-sounding" phrases etc., implies their high similarity to typical Spam [11]. The *easy_ham* corpus contains non-spam messages that are easily discriminated from Spam messages, since they do not contain any spammish signatures (like html etc). The *20030228_spam_2* collection also contains the *spam* corpus which comprises 1397 spam messages.We mixed the non spam corpora and the spam corpus to make a single testing corpus of about 4.147 emails for the evaluation purposes and tested our algorithm trying to classify these emails without training. For each email from the testing corpus we scanned html code and extracted everything, which can be used as a candidate feature for discrimination (see also section 4). All these features are extremely useful in the discrimination procedure and are included in the *FI*. To illustrate how difficult the classification task is we plotted in Figure 1 a part of the testing corpus emails, specifically the first 400 spam emails and the first 200 non spam e-mails projected onto its first 2 principal

components. In this figure the *star* represents non spam-emails and the *dot* represents spam emails. Notice how the spam emails of the testing corpus are closely "concentrated" on the same area and are overlapped with non-spam emails. The small overlapping area in the figure is very difficult to be discriminated.

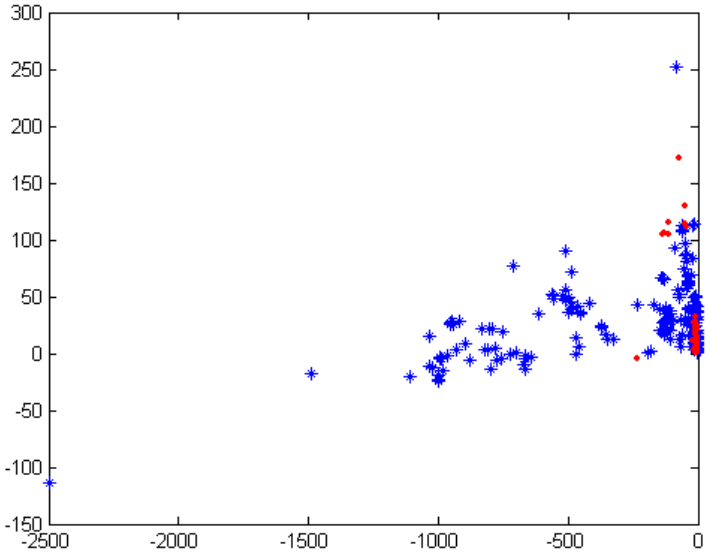

Figure 1: The 400 spam emails and the first 200 non-spam emails from the testing corpus projected onto its first 2 principal components

We eliminated from the *FI* the features with low frequency that is, those features that appeared at most into 15 incoming emails. Hence, the demand for computational resources, during the calculation of the first *PCL*, is reduced. The values of the first 10 eigenvalues which were used in the calculation of the Principal Components in the *PCL* list at the end of the clustering procedure are shown in the Table 1.

Table 1: The values of the first 10 eigenvalues λ_i of the covariance matrix of the testing data at the end of the testing clustering procedure.

Eigenvalue	value
1	64982
2	21179
3	5557.6
4	3767.5
5	2887
6	1099.4
7	968.27
8	926.04
9	756.99
10	621.78

In the Figure 2, we can see how the percentage of the total variability in the data is related to be explained by the first principal components.

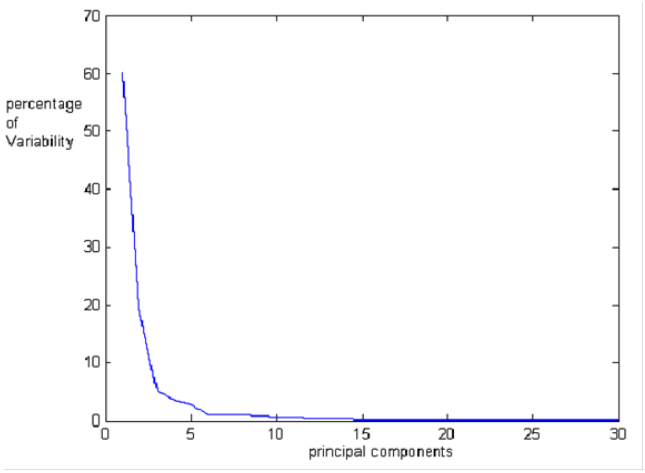

Figure 2: Percentage of the total variability explained by the first principal components in the testing corpus. As it was expected the 15 first principal components practically account for the total variability in the data.

Precision and *recall* are used as the evaluation measures in this work:

- *Precision* of a classification system for a category *C* evaluated over a sample of items, is the proportion of the items correctly classified as *C* in the total number of items classified as *C* (correctly classified and misclassified).
- *Recall* is the proportion of the items correctly classified as *C* in the total number of *C* items in the sample.

Table 2 shows the results achieved by our system in the evaluation experiments.

Table 2: Recall and Precision ratings evaluated for legitimate and spam mails.

	Recall	Precision
Spam	98.57%	90.77%
Legitimate	94.91%	99.24%

Results in Table 2 illustrate our experiments and imply a strong indication about the robustness of the proposed method. From the results, 2-means clustering technique has proven to be very effective technique for spam filtering document classification. It is still surprisingly accurate for its simplicity. It treats its features as only booleans, meaning that either a feature exists in an email or it does not exist in email. Given only that information, it averaged around a 90.77% precision. Moreover, it is still applicable to individual user situations, as it is far simpler than many other classifiers and still does an adequate job of filtering out spam messages. Other researchers also present high precision [12,3,13,14]. However, their calculations are based on test data with low similarity between legitimate and spam mail. Such an approach makes the classification process an easier task and there is little (or no) effect if tuning parameters are applied [14]. In recent research work based on versions of the same corpus, which was used in our work, lower degrees of precision and recall have been

reported [15], by applying SVM (Support Vector Machine). Furthermore, they used an updated version of the corpus (e.g. HTML comments and formatting tags have been removed) instead of using the hard_ham corpus as we have done in our evaluation.

6. Conclusion

It is a hard problem to distinguish unsolicited commercial e-mail, or "Spam", or spam emails, from legitimate ones and to avoid wasting of network bandwidth, and human time and effort in electronic communication. The methods used can be borrowed from Machine Learning. Popular methods include Naïve Bayes, Neural Networks, Nearest Neighbor, and Support Vector Machines. In all these methods, a collection of items is used to train a statistical model and then this model is applied to new "incoming" items. There are various proposed algorithms using supervised learning techniques. The training of the algorithms is based on large manually and costly tagged email corpora. Their calculations and the evaluation of the algorithms are usually based on test data with low similarity between legitimate and spam mail. There are also recent research works based on the publicly available corpus, called "SpamAssasin", provided by the Open Project for evaluation purposes. In this paper, we present an unsupervised method to address the problem of filtering spam emails without the need of training over huge corpora. A 2-means clustering technique is used and then, we perform a 2-way classification. To overcome the serious complications imposed by the large dimensionality of the data, our algorithm first transforms the data into a low dimensional component space applying a Principal Component Analysis over the data and then performs clustering on it. The proposed algorithm was evaluated over the "SpamAssasin" collection of e-mails. The method seems to be promising. The performance, which was achieved, is comparable to the performance of systems based on supervised learning techniques. It is an advantage that our method avoids the drawback of training over large manually and costly tagged email corpora. In the future, other clustering techniques will be used to classify spam emails without learning: Hierarchical Cluster Analysis (HCA) and Fuzzy C-Means clustering techniques [16,7].

7. Constraints and Recommendations

One constraint of the proposed algorithm is that it does not yield the same result with each run, since the resulting clusters depend on the initial random assignments. One way to addresses this problem is by seeking to choose better starting clusters. Another inherent limitation from k-means clustering techniques is that k-means assumes the variance of the distribution of each attribute is spherical and all attributes have the same variance. Our approach does not take into consideration other factor like images and any other attachments may be part of an email. There is a high probability that a spam email may contain no textual content but only an image or an attachment. Training the filter with a corpus containing non-textual content would improve its effectiveness during the classification phase. In the case of a hyperlink for example, we can have a web crawler that would visit the mentioned site and collect textual information to finally apply the same approach to classify the email.

References

[1] S.-E. Kim, J.-T. Jo, and S.-H. Choi, ''SMS spam filtering using keyword frequency ratio,'' The

International Journal of Security. vol. 9, no. 1, pp. 329–336, Apr 2015.

[2] D. Mallampati. “An Evaluation of Naïve Bayesian Classifier for Anti-Spam Filtering Techniques,” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 6, Issue 10, Oct 2017.

[3] K. Sharma and N Jatana. “Bayesian Spam Classification: Time Efficient Radix Encoded Fragmented Database Approach” IEEE, pp. 939-942., Feb 2014.

[4] R. Drewes. (2202, June 1). An artificial neural network spam classifier, available: www.interstice.com/drewes/cs676/spam-nn [3-2-2018].

[5] H. W. Feng and J. Sun. “A support vector machine based naive Bayes algorithm for spam filtering,” presented at Performance Computing and Communications Conference (IPCCC), IEEE 35th International, Las Vegas, USA, 2016.

[6] P. Tsangaratos and I. Ilia. "Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments," journal Catena Vol. 145 pp. 164-179, Jun 2016.

[7] Y.Lei, D.Yu, Z. Bin, and Y.Yang. "Interactive K-Means Clustering Method Based on User Behavior for Different Analysis Target in Medicine," journal Computational Mathematics Methods Medicine, Published online doi: 10.1155/2017/4915828, Oct 2017.

[8] A. Bansal, M. Sharma and S. Goel. "Improved K-mean Clustering Algorithm for Prediction Analysis using Classification Technique in Data Mining," International Journal of Computer Applications (0975 – 8887) Volume 157 – No 6, January 2017.

[9] U. R. Raval and C. Jani. "Implementing & Improvisation of K-means Clustering Algorithm," journal IJCSMC, Vol. 5, Issue. 5, pp.191 – 203, May 2016.

[10] C. R. Rao. “The Use and Interpretation of Principal Component Analysis in Applied Research” Journal of Sankhya, A **26**, pp. 329 -358, Feb. 1964.

[11] http://www.csmining.org/index.php/spam-assassin-datasets.html [3-5-2018].

[12] I. Androutsopoulos, J. Koutsias, K. Chandrinos, and C. Spyropoulos. “An experimental comparison of naïve Bayesian and keyword-based anti-spam filtering with personal e-mail messages,” presented at International Conference of SIGIR, May 2000.

[13] J. Hidalgo. “Evaluating Cost Sensitive Bulk Email Categorization,” in Proc. SAC, 2002, pp 615-620.

[14] X. Carreras and L. Marquez. “Boosting trees for anti-spam email filtering. In Proceedings of RANLP-01,” presented at International Conference on Recent Advances in Natural Language Processing,

Tzigov Chark, BG, 2001.

[15] E. Michelakis, I. Androutsopoulos, G. Paliouras, G. Sakkis and P. Stamatopoulos. “Filtron: A learning based Anti-Spam Filter,” presented at Conference on email and Anti-Spam, CA, USA, 2004

[16] M. X. Gong and M. B. Richman: “On the application of cluster analysis to growing season recipitation in North America east of the Rockies,” Journal Climata, pp. 8:897-931, 1995.