Clustering Ensemble for Spam Filtering

Santiago Porras[+], Bruno Baruque[+], Belén Vaquerizo[+], Emilio Corchado[*]

[+] Civil Engineering Department. University of Burgos.
[*] Departamento de Informática y Automática. Universidad de Salamanca.

Abstract. One of the main problems that modern e-mail systems face is the management of the high degree of spam or junk mail they recieve. Those systems are expected to be able to distinguish between legitimate mail and spam; in order to present the final user as much interesting information as possible. This study presents a novel hybrid intelligent system using both unsupervised and supervised learning that can be easily adapted to be used in an individual or collaborative system. The system divides the spam filtering problem into two stages: firstly it divides the input data space into different similar parts. Then it generates several simple classifiers that are used to classify correctly messages that are contained in one of the parts previously determined. That way the efficiency of each classifier increases, as they can specialize in separate the spam from certain types of related messages. The hybrid system presented has been tested with a real e-mail data base and a comparison of its results with those obtained from other common classification methods is also included. This novel hybrid technique proves to be effective in the problem under study.

1 Introduction

One of the main problem that e-mail systems face nowadays is the management of spam. By this term we refer to the action of sending of indiscriminate unwanted e-mails, usually done in order to attract potential clients to use e-commerce systems to purchase articles or services or even to fraudulent services in order to obtain personal information that will be used afterwards for criminal activities.

To help users to cope with this flow of unwanted e-mails, almost every modern e-mail service includes a spam filtering service. This kind of filtering is ideally intended to automatically distinguish between spam and normal e-mails to present the final user only the wanted e-mail messages, freeing them of the task of classifying and eliminating those unwanted messages by themselves. The task of discerning which kind of messages are of interest to the final user is a complex task to perform in advance; so these systems must rely on a predefined configuration and try to adjust it to the preferences that each user provides to the system through his/her normal daily use. This kind of situation is clearly a task where automated learning can be of much use, as an automatic algorithm can be trained to perform this classification in a transparent way for the final user, by adapting its behaviour as it is used.

Many approaches exist to tackle this very common problem. According to what parts of the e-mail the filtering system analyzes, the solutions can be divided into header or meta-information based or content based. According to the architecture of the system a rough division could be between individual and collaborative systems. Usually, a modern spam filtering system would try to combine all of these techniques to benefit from the straights of each.

This study presents a novel hybrid intelligent system using both unsupervised and supervised learning that can be easily adapted to be used in an individual or collaborative system. It makes use of the Self-Organizing Map for a initial partitioning of data and the Naive Bayes for the final e-mail classification.

The rest of this work is organized as follows: Section 2 presents a brief overview of the concept of ensemble learning, Section 3 includes a very simplified description of the SOM algorithm and more detailed explanations about its use a clustering algorithm. Section 4 introduces the hybrid model used for spam classification while Section 5 details the experiments performed with the algorithm and their results, compared with similar models. Finally conclusions and future work are described in Section 6.

2 Ensemble Learning

At its inception, the ensemble meta-algorithm [1] was created to improve the capabilities of existing models for data classification. The main concept behind ensemble learning model is the simple intuitive idea of a committee of experts working together to solve a problem.

The strength of the ensemble meta-algorithm is its potential to achieve a compromise between the desired result of both a small variance and a small bias; as a trade-off between fitting the data too closely (high variance) and not taking data into account at all (high bias). An important element is the effective combination of the classifiers, which relies in part on the presence of a certain variance in the components of the ensemble that is generally referred as 'diversity'. There is considerable evidence to suggest that the use of ensembles can lead to an improvement in the performance of single models in classification or regression tasks [2,3,4]. The underlying reason for increased reliability through the use of ensembles is that different classification algorithms will show different patterns of generalization. More formal explanations of the way ensembles can improve performance may be found in [5,6].

There are generally two ways to combine the decisions of classifiers in ensembles: classifier selection [7] and classifier fusion [6,8]. Classifier selection assumes that each classifier is an "expert" for some part of the feature space. In contrast, classifier fusion assumes that all classifiers are trained over the entire feature space. In the case of this work, the first approach is the one used.

3 Topology Preserving Maps

3.1 The SOM Algorithm

Topology preserving mapping comprises a family of techniques with a common target: to produce a low-dimensional representation of the training samples that preserves the topological properties of the input space. From among the various techniques, the best known is the Self-Organizing Map (SOM) algorithm [9]. SOM aims to provide a low-dimensional representation of multi-dimensional data sets while preserving the topological properties of the input space. The SOM algorithm is based on competitive unsupervised learning; an adaptive process in which the neurons in a neural network gradually become sensitive to different input categories, which are sets of samples in a specific domain of the input space . The update of neighbourhood neurons in SOM is expressed as:

$$w_k(t+1) = w_k(t) + \alpha(t)\eta(v,k,t)(x(t) - w_k(t)) \tag{1}$$

where, x denotes the network input, w_k the characteristics vector of each neuron; α, is the learning rate of the algorithm; and $\eta(v,k,t)$ is the neighbourhood function, in which v represents the position of the winning neuron (Best Matching Unit or BMU) in the lattice, and k the positions of the neurons in its neighbourhood.

3.2 Data Clustering Capabilities

The clustering capabilities of the SOM algorithm have been extensively used in many works [10,11]. Mainly intended as a visual aid for data clusters exploration, several calculations over the final map have been proposed. After the calculation is performed, it can be represented over the final resulting map as a colour scheme. In the case of this study the three measures presented in [12,13] are used.

U-Matrix: This matrix represents for each unit of the map, the sum of the euclidean distances between its corresponding weights vector and all the data samples that are recognized by it. It permits to get a visual idea of how concentrated or disperse is the data set in the manifold represented by the map. Obviously, it is a valuable tool to detect and determine clusters and cluster borders in the data set.

P-Matrix: In this case, the values represented in the matrix are calculated as the density measured in the data space at a specific point, where that point is the weight vector associated with each unit of the SOM. As with previous matrix, the measure of the concentration of data serves as a very good measure for finding clusters in data.

U-Matrix:* This final matrix combines the distance based U-Matrix and the density based P-Matrix. It consists in using the U-Matrix as a basis for the final matrix, and the P-Matrix. Again, this matrix is used to obtain data clusters, especially in thin populated regions of the data space; where distances are more important that density to determine the similarity of data.

As the results obtained by all these three matrix calculations consist basically in numerical values for each of the units composing the maps, they can be used as an automatic way of determining the number of clusters in a data set but more importantly in this case, can inform about things like the boundary of those clusters or the density of data included in them. That information is used in the presented model to decide the structure that the ensemble of base classifiers will have.

4 Proposed Model

The intelligent hybrid model proposed in this research tries to take further the idea initially expressed in [14]. This previous work uses the well known k-nearest neighbours algorithm [15] as a way to initially explore the data set input space to latter construct an ensemble of classifiers that use that previously gather information to split data into the different components in a more informed way that other ensemble methods that use random or probability based distributions [16,17]. That way, an ensemble is constructed, ensuring that the base classifiers composing it will be expert in different regions of the data space.

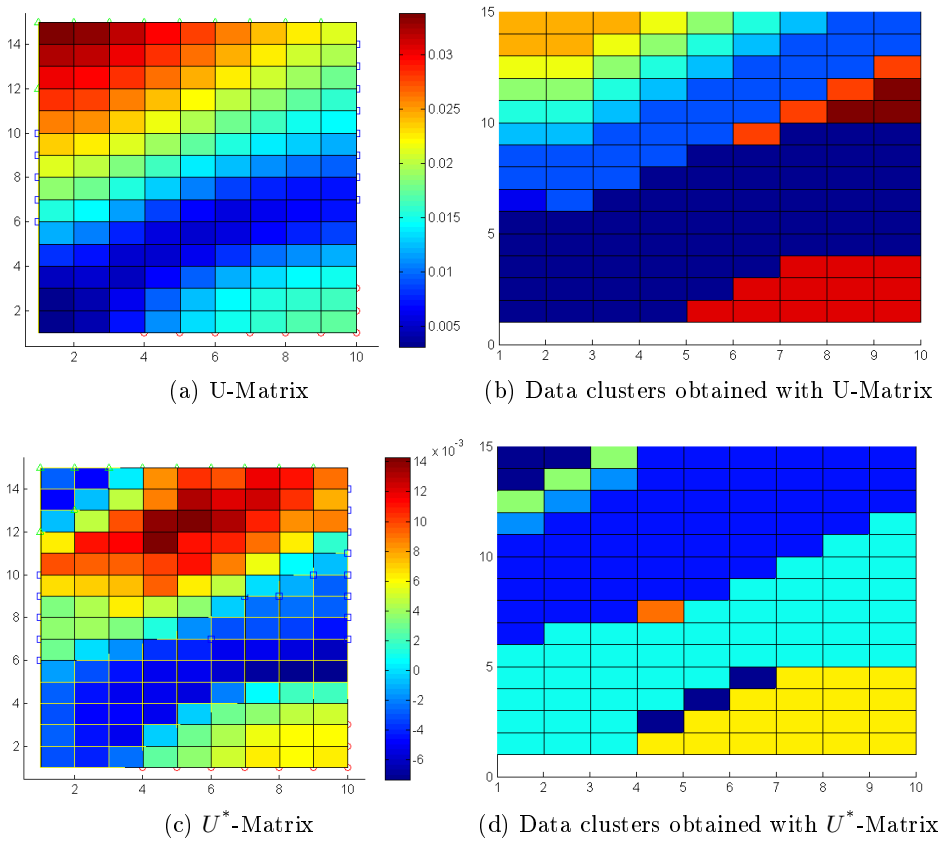
In this case the SOM algorithm will be used to perform automatically this previous analysis and division of the data space. The use of this algorithm has several advantages over using other, simpler algorithms, such as the k-nearest neighbours.

In first place, the SOM is a complete grid extending over the data input space. That implies that is easy to observe and take into account not only similarity between samples, but to determine an ordered impression of that similarity, just using the distance between units on the map. Using the measures described in Section 3.2 several options for data clustering can be used, in a very straightforward way, to decide which configuration or architecture could be better for the construction of the ensemble and the data set partitions to train the model.

In the case of this study the SOM enables to split data between clusters, but also it is easy to find cluster frontiers were more detail and precision is required; therefore, data in those areas can be used to train several classifiers to enhance accuracy in those regions. For example, data situated in the borders of two clusters could be assigned to two different classifiers to try to obtain several different classifiers for those regions. Also, clusters that are either too extensive or that contain a high density of data can have more classifiers to back the classifications of others.

Other advantage that comes from the use of the SOM is that there is no need to use the number of clusters as a parameter to the algorithm, instead a

threshold can be used to control how sensitive the algorithm is to changes in values for neighbouring units.



(a) U-Matrix (b) Data clusters obtained with U-Matrix

(c) U^*-Matrix (d) Data clusters obtained with U^*-Matrix

Fig. 1: Two of the mentioned matrix calculated on a 15x10 units SOM trained over the Iris data set

Figure 1 shows the values obtained for two of the matrix mentioned on Section 3.2 (Figs. 1a and 1c) and the corresponding clusters found with thresholds 0.0012 and 0.002 respectively (Figs. 1b and 1d) in the well-known case of the Iris data set [18]. Figs. 1a and 1c represent the values of the matrix in a color scale. It is easy to observe a gap with very low values that divides the data in two main groups. Figs. 1b and 1d depict the different clusters or divisions the algorithm has found in the data space. Each colour represents a different cluster.

The processing devised can be therefore summarized as in Algorithm 1.

Algorithm 1 SOM Clustering and Selecting Ensemble

Input: A data set to be classified $D \epsilon \mathbb{R}^n$*, a clustering threshold* θ_c*, a inclusion threshold* θ_m

Output: A model able to classify novel entries $C_1 ... C_n$

procedure CONSTRUCT ENSEMBLE($C_1 ... C_n$)
    Train a Self-Organizing Map over the input data set.
    Label the units of the map with the entries for which it has been considered as the BMU.
    Calculate the requested matrix values for each map unit (U-Matrix, P-Matrix, U^*-Matrix).
    Perform a clustering of units depending in the difference in the value calculated in step 3 for each unit and its neighbours, using θ_c.
    Include each data sample in each of the clusters found in step 4, according to which cluster its corresponding BMU belongs to. Eliminate the clusters with a number of samples lower than θ_m.
    Train a base classifier with the data entries that form part of each of the clusters.
end procedure
procedure CLASSIFY SAMPLE(s)
    Present sample to the Self-Organizing Map and find the BMU
    Present sample to the classifier(s) corresponding to the BMU
    Output the majority vote of classifications obtained in the previous step
end procedure

5 Experiments and Results

The spam data used in this study is the SpamAssassin Public Corpus, published by the Apache Software Foundation [19]. Its main characteristics are: it includes instances of 6047 e-mail messages, with about a 31% spam ratio, sub-divided in three different classes "easy ham", "hard ham" and "spam". Much more detailed information can be found on the Internet public repository.

5.1 Information Representation

In order to be able to work with the information contained in the e-mail is necessary to translate from plain text to a more manageable representation for an automated learning algorithm. This usually means to extract statistical characteristics form the analysis of the text, so each of the analyzed messages can be represented by an array of numerical values. This is a very widely known approach, basic in the discipline known as Information Retrieval (IR) [20,21]; used in tasks such as text clustering or classification, web searching, etc. In the case of this study, the codification used is the well-known "bag of words".

5.2 Experiments Description and Results

The experiments devised have the purpose of comparing different types of data partition methods to use for the ensemble construction. For all the five differ-

ent partition schemes compared, the results are obtained from a 5-fold cross-validation process.

The five schemes are: the standard Bagging algorithm [16], which generates overlapping data sub-sets; and the k-means clustering algorithm and three variants of the presented model (U-Matrix, P-Matrix and U^*-Matrix), which generate disjoint sub-sets. The k-means clustering algorithm was previously used in [14] and is included here for the great parallelism with the presented model.

The tests have the same structure independently of the models chosen: first the data set is split between training and test sets. The training set is split into several sub-sets and a base classifier is trained over the data of each sub-set. All models used the classic Naive-Bayes classifier as their base classifier. Then, the test set is presented again to each model. The split model used to include training data in different sub-sets is used again to separate the test set and each sub-set is used as the inputs for their corresponding classifier.

Data set	Size (units)	Learning Rate	Epochs	Neighbourhood
Iris	8x8	0.1	1200	4
SPAM (25%)	15x15	0.1	10000	7
SPAM (complete)	30x30	0.1	15000	15

Table 1: Parameters used for the SOM training in each case

Table 1 presents the parameters used for the SOM training in each of the data sets of the comparative.

			SOM		
Data set	Bagging	k-means	U-Matrix	P-Matrix	U^*-Matrix
Iris	5	5	13	4	1
SPAM (25%)	5	5	26	7	7
SPAM (complete)	5	5	58	27	15

Table 2: Number of divisions used for each model and data set

It is interesting to note that the bagging and k-means methods require as inputs the number of folds/clusters in which the data set will be divided. This assumes that the user will have a relative knowledge about the data set. As explained before, the proposed algorithm does not need a so hard restriction over the algorithm, but only a difference threshold parameter; that lets the process to calculate the most adequate number of clusters. Table 2 shows the difference of the models in this regard.

Table 3 shows the results obtained with the data set under analysis in this work. The Iris data set is one of the widely used for automatic learning test

			SOM		
Data set	Bagging	k-means	U-Matrix	P-Matrix	U^*-Matrix
Iris	4.66%	5.33%	**3.42%**	4.66%	4.66%
SPAM (25%)	19.1%	11.1%	10.17%	10.51%	**9.62%**
SPAM (complete)	50%	44.03%	39.54%	39.42%	**37.3%**

Table 3: Percentage of classification error obtained using different techniques

and is included for comparative purposes. Finally, the two data sets object of the study are a fraction of the original Spam data set, used to try to avoid the computational complexity of training the model with the complete data set; and the complete one.

As results show, the use of hybrid system consisting on the SOM to cluster the data as a previous step to construct the ensemble can be generally regarded as an improvement over the other compared techniques. As hinted before, this situation comes from the fact that the SOM can provide a more detailed manifold to cluster the data, as opposed to the k-means, which is a much simpler technique.

When dealing with a very simple data set, as it is the case of the Iris data set, the extra complexity of calculating the SOM might not be so rewarding (the improvement is of less than a 2%). On the contrary, when working with bigger and more complex data sets (as the complete Spam, studied in this work) where there are large parts of the data set with overlapping classes, the more detailed divisions of the SOM can be of more clear use. In this study, the SOM obtains an error more than 5% lower than that of the k-means (see Table 3).

Regarding the different variants of the measures that can be calculated over the SOM, results seem to be better for the U-Matrix for the simpler data sets and for the U^*-Matrix for the more complex ones. This is result that will need further study.

6 Conclusions and Future Work

This study presents a model for the detection and filtering of spam based data on a hybrid intelligent system including both unsupervised and supervised learning. The model is variant of the idea of the clustering and selection concept, previously proposed by using a SOM as the clustering algorithm.

This novel hybrid model can be considered interesting for this task for several reasons. As explained in this work, one interesting topic on spam-filtering is the collaborative approaches, where several filtering systems work together to detect spam. The cluster and selection scheme studied here can be very easily adapted to work in these cases to distribute the classification among more specialized systems. This provides a heterogeneous framework where many models could be included. An additional reason is that the model can be used to provide the top-level mailing system administrator with complementary information, as the

SOM included in the model has multi-dimensional data visualization as one of its main characteristics. Also, having an auto-adaptive model such as the SOM, which functioning can vary with the continuous training; could serve to soften the effect of concept drifts or modifications in the behaviour of spammers.

Regarding future work, it is interesting to mention that the meta-model presented here can be extended and improved in many ways, as it can be used as a framework for more complex schemes. As an example, in the test showed only a base classifier was trained for each of the clusters calculated. But due to the more detailed capabilities of the SOM for clustering, it could be adapted to train classifiers for the data samples near the borders of clusters or to train several different classifiers to strengthen classification in clusters where classes overlap in a high degree.

Acknowledgments. This research has been partially supported through project of the Spanish Ministry of Science and Innovation TIN2010-21272-C02-01 (funded by the European Regional Development Fund). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., within the framework of the MAGNO2008 - 1028.- CENIT Project also funded by the MICINN.

References

1. D. Ruta and B. Gabrys, "An overview of classifier fusion methods," *Computing and Information Systems* **Vol. 7, No. 1**(ISSN 1352-9404), pp. pp. 1–10, 2000.
2. R. E. Schapire, "The strength of weak learnability," *Machine Learning* **vol. 5, no. 2**, pp. 197–227, 1990.
3. B. Baruque and E. Corchado, "A weighted voting summarization of SOM ensembles," *Data Mining and Knowledge Discovery* **21**, pp. 398–426, 2010. 10.1007/s10618-009-0160-3.
4. E. Corchado and B. Baruque, "Wevos-visom: An ensemble summarization algorithm for enhanced data visualization," *Neurocomputing* **in press**, 2011.
5. A. Sharkey and N. Sharkey, "Combining diverse neural nets," *Knowledge Engineering Review* **12, 3**, pp. 1–17, 1997.
6. L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, 2004.
7. R. Jacobs, M. I. Jordan, N. S. J., and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation* **3**, pp. 79–87, 1991.
8. R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine* **6**(3), pp. 21–45, 2006.
9. T. Kohonen, *Self-Organizing Maps*, vol. 30, Springer, Berlin, Germany, 1995.
10. J. Lampinen and E. Oja, "Clustering properties of hierarchical self-organizing maps," *Journal of Mathematical Imaging and Vision* **2**, pp. 261–272, 1992.
11. R. Dara, S. C. Kremer, and D. A. Stacey, "Clustering unlabelled data with SOMs improves classification of labelled real-world data," in *Proc. IEEE World Congress on Computational Intelligence*, pp. 2237–2242, May 2002.
12. A. Ultsch, "Self-organizing neural networks for visualization and classification," in *Proc. Conf. Soc. for Information and Classification*, 1992.

13. A. Ultsch, "U*-matrix: A tool to visualize clusters in high dimensional data," tech. rep., Department of Computer Science, University of Marburg, 2003.
14. L. I. Kuncheva, "Clustering-and-selection model for classifier combination," in *KES*, pp. 185–188, 2000.
15. K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is nearest neighbor meaningful?," in *Computer Science Database Theory ICDT'99*, *Lecture Notes in Computer Science* **Volume 1540/1999**, pp. 217–235, Springer, 1999.
16. L. Breiman, "Bagging predictors," *Machine Learning* **24**(2), pp. 123–140, 1996.
17. Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, pp. 148–156, 1996.
18. A. Asuncion and D. J. Newman, "UCI machine learning repository," 2007.
19. Apache Software Foundation, "Spamassasin public corpus," 2006.
20. A. Singhal, "Modern information retrieval: A brief overview," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* **24 (4)**, pp. 35–43, 2001.
21. M. E. Maron, "An historical note on the origins of probabilistic indexing," *Information Processing and Management* **44**, pp. 971–972, 2008.