Dublin Institute of Technology
ARROW@DIT

Conference papers

School of Computing

2005-01-01

Generating estimates of classification confidence for a case-based spam filter

Sarah Jane Delany
Dublin Institute of Technology, Sarahjane.Delany@dit.ie

Padraig Cunningham
Trinity College Dublin

Donal Coyle
Trinity College Dublin

Anton Zamolotskikh
Trinity College Dublin

Follow this and additional works at: http://arrow.dit.ie/scschcomcon

Part of the Physical Sciences and Mathematics Commons

Recommended Citation

Delany, Sarah Jane et al: Generating estimates of classification confidence for a case-based spam filter. Proceedings of the 6th. International Conference on Case-Based Reasoning (ICCBR'05), LNAI 3620, pp.170-190.

This Conference Paper is brought to you for free and open access by the School of Computing at ARROW@DIT. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie.

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License

Generating Estimates of Classification Confidence for a Case-Based Spam Filter

Sarah Jane Delany[1], Pádraig Cunningham[2], Dónal Doyle[2], and Anton Zamolotskikh[2]

[1] Dublin Institute of Technology,
Kevin Street, Dublin 8, Ireland
sarahjane.delany@comp.dit.ie
[2] University of Dublin, Trinity College,
Dublin 2, Ireland
{padraig.cunningham, donal.doyle, zamolota}@cs.tcd.ie

Abstract. Producing estimates of classification confidence is surprisingly difficult. One might expect that classifiers that can produce numeric classification scores (e.g. k-Nearest Neighbour, Naïve Bayes or Support Vector Machines) could readily produce confidence estimates based on thresholds. In fact, this proves not to be the case, probably because these are not probabilistic classifiers in the strict sense. The numeric scores coming from k-Nearest Neighbour, Naïve Bayes and Support Vector Machine classifiers are not well correlated with classification confidence. In this paper we describe a case-based spam filtering application that would benefit significantly from an ability to attach confidence predictions to positive classifications (i.e. messages classified as spam). We show that 'obvious' confidence metrics for a case-based classifier are not effective. We propose an ensemble-like solution that aggregates a collection of confidence metrics and show that this offers an effective solution in this spam filtering domain.

1 Introduction

One might expect that classifiers that produce numeric scores for class membership would deliver effective estimations of prediction confidence based on thresholds on these scores. Examples of classifiers that produce numeric scores in this manner are; Naïve Bayes, k-Nearest Neighbour [1], Neural Networks [2], Logistic Regression [3] and Support Vector Machines [4]. Our experience with these classifiers suggests that the numeric scores from Logistic Regression are predictive of confidence but those from Naive Bayes, Neural Networks, Support Vector Machines (SVM) and k-Nearest Neighbour (k-NN) are not. We demonstrate that this is the case for k-NN, Naïve Bayes and SVM in Section 3.

In this paper we are concerned with generating estimates of classification confidence for a case-based spam filter called ECUE (Email Classification Using Examples) [5]. ECUE has the advantage of being very effective at tracking concept drift but this requires the user to identify False Positives (FPs) and False

Negatives (FNs) so that they can be used to update the case-base. Identifying FNs is not a problem because they turn up in the Inbox (i.e. spam that has been allowed through the filter). Identifying FPs involves monitoring a spam folder to identify legitimate email that has been classified as spam. Our objective here is to be able to partition this class so that the user need only monitor a subset - the set for which the confidence is low.

A straightforward success criterion in this regard is the proportion of positives for which prediction confidence is high and the prediction is correct (clearly there cannot be any FPs in this set). A mechanism that could label more than 50% of the positive class (i.e. classified as spam) as confident and have no FPs in this set would be useful. The lower-confidence positives could be allowed into the Inbox carrying a *Maybe-Spam* marker in the header or placed in a *Maybe-Spam* folder that would be checked periodically.

In section 2 we provide a brief overview of research on estimating confidence. The basic indicators for confidence that can be used with k-NN are described in section 3 where we show that no single one of these measures is effective in estimating confidence. In section 4 we present some simple techniques for aggregating these basic indicators and present an evaluation on unseen data that shows a simple voting technique to be very effective. The paper concludes in section 5 with a summary.

2 Review

Cheetham and Price have recently emphasised the importance of being able to attach confidence values to predictions in CBR [6, 7]. This has been a research issue since the earliest days of expert systems research: it is part of the body of research on meta-level knowledge [8, 9], the view being that it is important for a system to 'know what it knows'. TEIRESIAS is a system in this spirit, it was designed to simply admit its ignorance instead of venturing risky advice [10].

More recently, the system SIROCCO from McLaren and Ashely [11] uses meta-rules to determine the system's confidence. Their system operates in an engineering ethics domain, in which incorrect suggestions could be considered sensitive and damaging. In this system, if any one of the meta-rules are fired then the system considers itself inadequate for the task. Their evaluation of SIROCCO shows that allowing the system to produce 'don't know' results reduces the number of incorrectly classified cases, with a small trade off whereby the number of correctly classified cases is reduced.

So while it is clear that it is useful to be able to produce estimates of confidence, it is also clear that that generating reliable estimates is not straightforward. Cheetham and Price [7] describe 12 measures of confidence that can be applicable for a k-NN classifier. Some of these indicators increase with confidence and some decrease. Since no single indicator is capable of producing a robust measure of confidence they explore the use of a decision tree, that is allowed to use all the measures, as a mechanism for aggregating all the available metrics. The authors show that, even using a decision tree to learn a good con-

fidence measure from historic data, it is difficult to avoid the situation where predictions labelled as confident prove to be incorrect. They also emphasise that the confidence estimation mechanism will need to be updated over time as the nature of the problems being solved can change.

Because of this we choose to concentrate on simpler aggregation mechanisms. We engineered all indicators so that they increased in value as confidence increased. This allowed us to consider additive and multiplicative mechanisms as well as various 'voting' alternatives.

2.1 Indirect Methods of Conveying Confidence

It is worth mentioning that there are other more indirect ways of conveying confidence to the user. Rather than conveying confidence as a term or a numeric score it can be conveyed by giving the user some insight into the problem domain. Confidence can be conveyed by presenting explanation cases [12] or by highlighting whether a feature has a negative or positive correlation with respect to the classification [13] or by highlighting features that contribute positively and negatively to the classification [14] Confidence may also be conveyed by using visualisation tools to highlight features that contribute to similarity and to differences [15].

3 Confidence Measures

This section describes a number of confidence measures that could be used to predict confidence in ECUE, a case-based spam filter. We concentrate on using measures appropriate for a k-NN classifier. We evaluate these measures on a number of spam datasets to assess their performance at predicting confidence.

The k-NN measures that we propose evaluating, which are described in Section 3.1, perform some calculation on a ranked list of neighbours of a target case. We do not use the basic classification score of the target case as ECUE uses unanimous voting in the classification process to bias the classifier away from FPs. Unanimous voting requires all the k nearest neighbours retrieved to be of classification *spam* in order for the target case to be classified as *spam*. Therefore there is no classification 'score', as such.

3.1 Proposed k-NN Confidence Measures

The objective of the k-NN measures is to identify those cases that are 'close' (i.e. with high similarity) to cases of the same class as the target case and are 'far' (i.e. low similarity) from cases of a different class. The closer a target case is to cases of a different class, the higher the chance that the target case is lying near or at the decision surface. Whereas the closer a case is to other cases of the same class, the higher the likelihood that it is further from the decision surface.

Similarity is determined by comparing features including the words and letters used in the body of the email and certain header fields including the subject, the 'from' address and addresses in the 'to' and 'cc' header fields [5].

For each k-NN confidence measure discussed in this section the same process occurs. Each target case is classified by ECUE as either spam or non-spam. For those target cases predicted to be spam a ranked list of neighbours of the target case is retrieved. This list of neighbours is a list of all the cases in the case-base ordered by distance from the target case. Those cases with classification equal to that of the target case (i.e. with classification spam) are considered to be *like* cases, while those cases with classification of nonspam are considered to be *unlike* cases. The measures can use

- the distance between a case and its nearest neighbours (let $NN_i(t)$ denote the ith nearest neighbour of case t) or,
- the distance between the target case t and its nearest like neighbours (let $NLN_i(t)$ denote the ith nearest *like* neighbour to case t) and/or
- the distance between a case and its nearest unlike neighbours (let $NUN_i(t)$ denote the ith nearest *unlike* neighbour to case t).

The number of neighbours used in each measure is adjustable and is independent of the number of neighbours used in the initial classification. All measures are constructed to produce a high score to indicate high confidence and a low score to indicate low confidence.

Avg NUN Index

The Average Nearest Unlike Neighbour Index (Avg NUN Index) is a measure of how close the first k NUNs are to the target case t as given in Equation 1.

$$AvgNUNIndex(t,k) = \frac{\sum_{i=1}^{k} IndexOfNUN_i(t)}{k} \tag{1}$$

where $IndexOfNUN_i(t)$ is the index of the ith nearest unlike neighbour of target case t, the index being the ordinal ranking of the case in the list of NNs.

This is illustrated in Figure 1 where NLNs are represented by circles, NUNs are represented by stars and target cases are represented by triangles. For $k = 1$, the index of the first NUN to target case T_1 is 5 whereas the index of the first NUN to target case T_2 is 2, indicating higher confidence in the classification of T_1 than T_2.

Similarity Ratio

The Similarity Ratio measure calculates the ratio of the similarity between the target case t and its k NLNs to the similarity between the target case and its k NUNs, as given in Equation 2.

$$SimRatio(t,k) = \frac{\sum_{i=1}^{k} Sim(t, NLN_i(t))}{\sum_{i=1}^{k} Sim(t, NUN_i(t))} \tag{2}$$

where $Sim(a,b)$ is the calculated similarity between cases a and b.

This is illustrated in Figure 2 where, for $k = 1$, the similarity between the target case T_1 and its NLN is much higher than the similarity between T_1 and

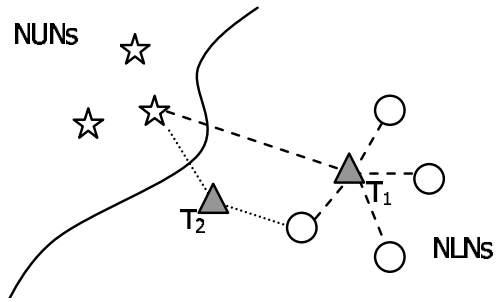

Fig. 1. Average NUN Index Confidence Measure

its NUN. Whereas the similarity between target case T_2 and its NLN is only marginally higher than the similarity between T_2 and its NUN. The ratio of these similarites for T_1 will give a higher result than that for T_2 indicating higher confidence in the classification of T_1 than T_2.

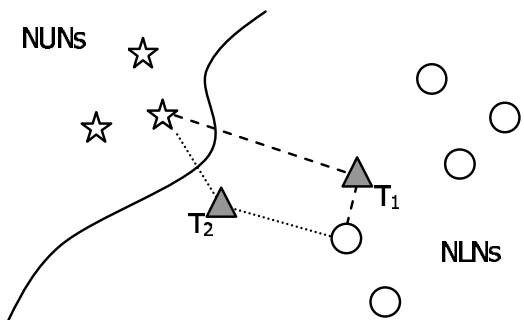

Fig. 2. Similarity Ratio Confidence Measure

Similarity Ratio Within K

The Similarity Ratio Within K is similar to the Similarity Ratio as described above except that, rather than consider the first k NLNs and the first k NUNs of a target case t, it only uses the NLNs and NUNs from the first k neighbours. It is defined in Equation 3.

$$SimRatio(t,k) = \frac{\sum_{i=1}^{k} Sim(t, NN_i(t)) 1(t, NN_i(t))}{1 + \sum_{i=1}^{k} Sim(t, NN_i(t))(1 - 1(t, NN_i(t)))} \tag{3}$$

where $Sim(a,b)$ is the calculated similarity between cases a and b and $1(a,b)$ returns one if the class of a is the same as the class of b or zero otherwise.

This measure will attempt to reward cases that have no NUNs within the first k neighbours, i.e. are in a cluster of k cases of the same class. This is illustrated

in Figure 3 where, considering $k = 3$, the target case T_1 has no NUNs within the first three neighbours whereas target case T_2 has two NUNs and one NLN. The Similarity Ratio Within K will be much larger for T_1 than that for T_2 indicating higher confidence in the classification of T_1 than T_2.

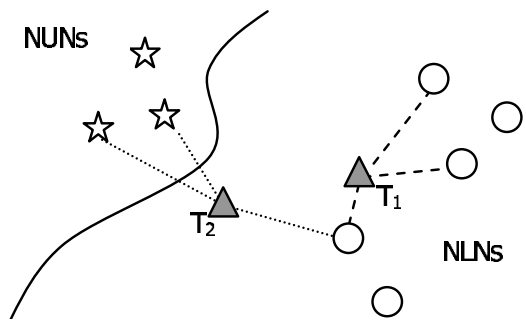

Fig. 3. Similarity Ratio Within K Confidence Measure

If a target case t has no NUNs then Equation 3 is effectively Equation 2 with the denominator set to one.

Sum of NN Similarites

The Sum of NN Similarities measure is the total similarity of the NLNs in the first k neighbours of the target case t, see Equation 4.

$$SumNNSim(t,k) = \sum_{i=1}^{k} 1(t, NN_i(t)) Sim(t, NN_i(t)) \tag{4}$$

where $Sim(a,b)$ is the calculated similarity between cases a and b and $1(a,b)$ returns one if the class of a is the same as the class of b or zero otherwise.

For target cases in a cluster of cases of similar class this number will be large. For cases which are closer to the decision surface and have NUNs within the first k neighbours, this measure will be smaller. In fact for target cases with no NUNs within the first k neighbours this measure will be equal to the value of the Similarity Ratio Within K. Although this measure does not reward such cases as strongly as the Similarity Ratio Within K does as the resulting measure for the sum of the NLNs is not reduced by the influence of the NUNs.

Average NN Similarity

The Average NN Similarity measure is the average similarity of the NLNs in the first k neighbours of the target case t, see Equation 5.

$$SumNNSim(t,k) = \frac{\sum_{i=1}^{k} 1(t, NN_i(t)) Sim(t, NN_i(t))}{\sum_{i=1}^{k} 1(t, NN_i(t))} \tag{5}$$

where $Sim(a, b)$ is the calculated similarity between cases a and b and $1(a, b)$ returns one if the class of a is the same as the class of b or zero otherwise.

3.2 Assessing k-NN Confidence Measure Performance

In order to assess the performance of these confidence measures we evaluated each of them on a number of spam datasets. Five datasets were used. Each consisted of legitimate and spam emails received by a single individual over a period of time. Each dataset represents a different period of time for a single individual. Two different individual's mail were used over all datasets. The legitimate emails in the datasets include a mixture of business, personal and mailing list emails. Case-bases were built from each of the five original datasets. Case representation details are available in [5, 16].

ECUE's case-base maintenance procedure to handle concept drift in spam filtering [17] has two components; an initial case-base editing stage and a case-base update protocol. In order for the evaluation to closely reflect the operation of ECUE, the case-base from each dataset was edited using the case editing procedure [18]. After editing the datasets averaged 700 emails in size with an average of 45% spam and 55% legitimate emails.

The evaluation involved performing a leave-one-out validation on each dataset for each measure. We evaluated each measure using k neighbours from $k = 1$ upto $k = 15$ and identified the confidence threshold, over all the k values, that gave us the highest proportion of correctly predicted spam emails when there were no incorrect predictions (i.e. FPs). This is illustrated in Figure 4.

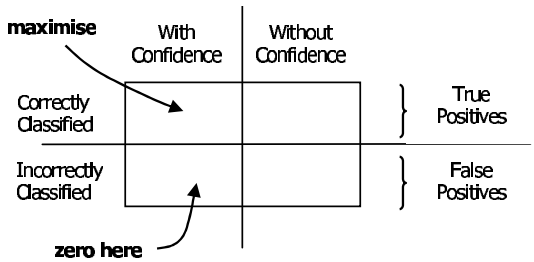

Fig. 4. Criteria used to identify the best confidence threshold level

This was achieved by recording the confidence measure results for each target case $c_i, i = 1 \ldots N$, that was classified by ECUE as spam. The results recorded included the number of neighbours k used in the measure, whether the target case was classified correctly or not and the measure calculated, m_{ik}. Setting the threshold t_k equal to the minimum value of m_{ik} for a given k and varying the threshold in small units ($t_k = t_k + .01$) up to the maximum value of m_{ik}, the number classified correctly with confidence (CC_k) and the number classified

incorrectly with confidence (CI_k) were calculated, where confidence exists for case c_i when $m_{ik} > t_k$.

The selected threshold value was the threshold t_k that maximised CC_k, the number of spam correctly predicted with high confidence when the number of incorrect predictions with high confidence was zero (i.e. $CI_k = 0$).

The results of this evaluation are presented in rows 1 to 5 of Table 1 (the other measures in rows 6 to 8 are described later). It details for each measure the highest percentage confidence that can be achieved on each dataset. This is the proportion of *spam* predictions that are made with high confidence. In all situations no highly confident incorrect predictions were made so no FPs are included in this proportion. In effect, this proportion of the spam can be ignored by the user, whereas the remaining percentage would have to be checked by the user.

Table 1. Best percentage confidence achievable for each dataset using different confidence measures

Confidence Measure	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Dataset 4**	**Dataset 5**	**Avg**
Avg NUN Index	23%	76%	75%	41%	44%	51.8%
Sim Ratio	46%	84%	50%	49%	16%	49.0%
Sim Ratio Within k	21%	29%	71%	91%	57%	54.8%
Sum NN Sim	21%	29%	68%	91%	58%	53.4%
Avg NN Sim	20%	29%	49%	91%	60%	49.8%
Naive Bayes	0%	94%	0%	83%	56%	46.4%
SVM	29%	100%	77%	81%	33%	63.8%
ACM	55.4%	85.4%	83.8%	93.7%	77.3%	79.1%

Looking at the proportion of spam predictions for which confidence is high across all datasets it is evident that no single measure achieves good percentage confidence across all datasets. If we define "good" performance as having confidence in at least 50% of the spam predictions, none of the measures achieve "good" performance on more than three of the five datasets. The best performing measure is the Similarity Ratio Within K which has good performance on three of the five datasets with an average performance across all datasets of 54.8% but with minimum performance of 21%.

3.3 Naïve Bayes and SVM Confidence Measures

Naïve Bayes is currently the machine learning technique of choice for spam filtering [19–23] although there has been a lot of interest recently in applying SVMs to the problem [23–27]. Naïve Bayes and SVM classifers produce numeric scores; Naïve Bayes produces a 'probability' of spam whereas an SVM produces a 'distance' from the hyperplane separating the spam and non spam classes. These scores can be used to predict confidence in the classifiers' prediction.

We examined confidence measures produced by Naïve Bayes on the five datasets. The implementation used is that described by Delany et al. [5]. The confidence threshold was identified as the highest numeric score returned by the classifier for a FP prediction. This ensured that no incorrectly classified spam emails were considered confident predictions. The 6th row of Table 1 gives the confidence predictions for the five datasets using the Naïve Bayes classifier. It is clear from the results that the Naïve Bayes numeric score cannot be used as a predictor of confidence. In two of the five datasets there are zero confident predictions as there are FPs with the maximum score.

We also evaluated using a SVM on the five datasets. The implementation used is a 2-norm soft-margin SVM as described in [4] with a dot product kernel function. The confidence threshold was identified as the highest postive result returned for nonspam email. This will ensure that no legitimate email will be confidently considered as spam. The 7th row of the Table 1 gives the confidence predictions for the five datasets using an SVM for classification. Although the average score across all datasets of 63.8% is higher than the best of the k-NN measures the SVM confidence measure does not realistically achieve any better overall performance as it also only achieves "good" performance on three of the five datasets but with slightly higher minimum performance of 29%. It is worth noting that the performance of dataset 2 is actually 99.7% but is reported as 100% due to rounding.

3.4 Implications for Predicting Confidence in Spam Filtering

To summarise, it appears that the confidence measures for k-NN, Naïve Bayes and SVMs presented here cannot consistently produce estimates of prediction confidence for spam. The average performance of the k-NN and the SVM measures shows promise however the lack of consistency across all datasets is an issue. The thresholds achieved for each k-NN measure across the five datasets also varies considerably. For example, considering the Similarity Ratio Within K measure which has the best of the k-NN measures performance, Table 2 shows the variation in the threshold across the five datasets.

Table 2. Demonstrating the variation in thresholds for the Similarity Within K Ratio confidence measure across the five datasets

Threshold	**Dataset 1**	**Dataset 2**	**Dataset 3**	**Dataset 4**	**Dataset 5**
k - num neighbours used	11	7	14	1	3
Value	991.07	574.08	717.04	58	214.1

It is important to note that the figures in Table 1 are very optimistic as the test data was used to set the threshold.

4 The Aggregated Confidence Measure

Since none of the individual measures discussed in Section 3 was consistently effective at predicting confidence we evaluated a number of aggregation approaches which involved combining the results from the individual measures. The aggregation approaches we considered included:

(i) Summing the results from each of the 5 individual measures evaluated at the same value of k and comparing the sum against a threshold;
(ii) Using the best threshold for each individual measure and indicating confidence if a certain number of the measures indicate confidence;
(iii) Using a fixed k across all measures and indicating confidence if a certain number of the measures indicate confidence.

We found that the simplest and most effective method of aggregating the results is to assign confidence to a prediction if any of the individual measures indicated that the prediction was confident as in (ii) above. We call this measure the Aggregated Confidence Measure (ACM). The algorithm for the ACM has two stages:

(i) calculation of the constituent measure threshold values in a pre-classification stage,
(ii) determination of the ACM during classification.

The pre-classification stage involves pre-processing of the case-base to identify the best threshold for each individual constituent measure. This is performed in the manner described in Section 3.2. A threshold consists of two values; the k value indicating the number of neighbours to use in the calculation and the actual threshold value above which the prediction is considered confident. These constituent measure thresholds are stored.

The ACM is then determined during classification for each target case that is classified as spam by ECUE. Using the appropriate threshold value of k, the actual score for each individual constituent measure is calcuated for the target case. The ACM specifies that if at least one of the calculated scores for the individual measures is equal to or greater than the stored threshold value for that measure, confidence is expressed in the prediction.

4.1 Assessment of ACM's Performance

We evaluated the ACM on the five datasets already used in Section 3. The results are presented in row 8 of Table 1. It is evident that the ACM is effective across all datasets with an average of 79% of the spam predictions being predicted with high confidence. The ACM also results in more than 50% of each dataset being predicted with high confidence. It is worth noting that the level of high confidence predictions for the ACM is also higher than the best individual measure's performance on each dataset (rows 1 to 5 of Table 1).

4.2 Evaluation on Unseen Data

One limitation of the evaluation performed in Section 4.1 is that the assessment was performed on the datasets which themselves were used to derive the confidence thresholds for the constituent confidence measures. In order to validate the ACM it is necessary to evaluate its performance on unseen data.

To do this we used ECUE along with two further datasets that have been used in concept drift evaluations of ECUE [17]. Each dataset is derived from an individual's email received over the period of approximately one year. The first 1000 emails (consisting of 500 spam and 500 legitimate emails) in each dataset were used as training data to build the initial case-base classifier and the remaining emails were left for testing. These datasets, 6 and 7, include eight and six months of test emails repectively. The monthly class distribution of the test emails is evident in rows 2 and 3 of Tables 3 and 4.

To evaluate the ACM on unseen data involved building confidence thresholds for the ACM constituent measures on the initial case-base and then classifying the remaining emails using the ACM to determine how confident the *spam* predictions are. In this way, the test emails were not used in the determination of the confidence thresholds in any way.

The test emails were presented in date order for classification. Since this email data is subject to concept drift, ECUE's case-base update policy was applied to allow the classifier to learn from the new types of spam and legitimate email presented. The update policy has a number of components; an immediate update of the case-base with any misclassified emails when a FP occurred, a daily update of the case-base with any other misclassifieds emails that occurred that day, and a monthly feature reselection process to allow the case representation to take any new predictive features into account. In order to keep the confidence thresholds in line with the updates to the case-base an update policy for the confidence thresholds was also applied. This policy had two components; the confidence thresholds were updated whenever a confident FP email occurred and also after a monthly feature reselect.

Tables 3 and 4 show the results of testing the performance of the ACM on unseen data using the two datasets 6 and 7. The tables present the accumulated monthly results for each dataset listing the total number and types of emails that were classified, the percentage of incorrect spam predictions (i.e. FPs) made (labeled *%FP classified*) and the percentage of incorrect spam predictions made with high confidence (labeled *%Confident FPs*). The table also gives the total percentage of spam predictions with high confidence (labeled *%Confidence*).

In both datasets predictions of confidence are high, averaging 85% in both cases with a lowest monthly level of 64%. This is the percentage of spam predictions that can be ignored by the user, the remaining spam predictions can either be flagged in the Inbox as *Maybe Spam* or placed in a separate *Maybe Spam* folder for the user to check.

However in some of the months the ACM has resulted in confident incorrect predictions. Although the actual numbers of emails are low (four emails for Dataset 6 and six emails for Dataset 7) the ideal situation is one where all

incorrect predictions have low confidence and will be flagged for the user to check. FPs flagged as confident will end up in the *spam* folder and may be missed by the user. Examining the confident FPs, three are emails from mailing lists and two are responses to Web registrations which users may not be too concerned with missing. The remaining five are important, some work related and one even a quotation in response to a online car hire request.

It is clear that we are approaching the limits of the accuracy of machine learning techniques in this domain. We see two possibilities for addressing these FPs. Close examination of such emails may identify domain specific characteristics that could be used as a feature or number of features in the case representation. Secondly, most deployed spam filtering solutions do not rely on one approach for filtering spam, they combine a number of techniques including white and black listing, rules, collaborative and learning approaches. Incorporating additional techniques into ECUE to add to its case-based approach could help in catching these outlier FPs.

Table 3. Performance of ACM on unseen data using Dataset 6

Month	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Overall**
Total emails classified	772	542	318	1014	967	1136	1370	1313	7382
Number of Spam	629	314	216	925	917	1065	1225	1205	6496
Number of Non Spam	93	228	102	89	50	71	145	108	886
%FPs classified	4.3%	2.6%	1.0%	1.1%	6.0%	1.4%	0.0%	1.9%	2.0%
%Confident FPs	0.0%	0.9%	0.0%	1.1%	0.0%	0.0%	0.0%	0.9%	0.5%
%Confidence	70%	87%	76%	94%	89%	73%	77%	99%	85%

Table 4. Performance of ACM on unseen data using Dataset 7

Month	**1**	**2**	**3**	**4**	**5**	**6**	**Overall**
Total emails classified	293	447	549	693	534	495	3011
Number of Spam	142	391	405	459	406	476	2279
Number of Non Spam	151	56	144	234	128	19	732
%FPs classified	0.7%	3.6%	3.5%	2.6%	1.6%	0.0%	2.2%
%Confident FPs	0.0%	3.6%	0.7%	0.4%	1.6%	0.0%	0.8%
%Confidence	95%	95%	87%	64%	89%	88%	85%

5 Conclusions

We have shown that confidence measures based on the numeric scores from Naïve Bayes, SVM or measures based on the k nearest neighbours for a case-based classifier are not consistent at predicting confidence in the spam filtering domain.

We have described an aggregation-based approach to combining individual k-NN confidence measures that shows great promise in confidently predicting spam. We evaluated this aggregated confidence measure by incorporating it into the classification process of a case-based spam filter and showed that it could successfully separate the spam predictions into two sets, those with high confidence of spam which can be ignored by the user and those with low confidence which should be periodically checked for False Positives. The high-confidence set included 85% of the predicted spam reducing the number of spam that the user needs to check.

References

1. Mitchell, T.: Machine Learning. McGraw Hill, New York (1997)
2. Fausett, L.: Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice Hall (1993)
3. Hosmer, D.W., Lemeshow, S.: Applied Logistic Regression. Wiley Series in Probability and Statistics. Wiley (2000)
4. Christianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods. Cambridge University Press (2000)
5. Delany, S., Cunningham, P., Coyle, L.: An assessment of case-based reasoning for spam filtering. Artificial Intelligence Review (to appear) (2005)
6. Cheetham, W.: Case-based reasoning with confidence. In Blanzieri, E., Portinale, L., eds.: 5th European Workshop on Case-Based Reasoning. Volume 1898 of LNCS., Springer (2000) 15–25
7. Cheetham, W., Price, J.: Measures of solution accuracy in case-based reasoning systems. In Funk, P., González-Calero, P., eds.: 7th European Conference on Case-Based Reasoning (ECCBR 2004). Volume 3155 of LNAI., Springer (2004) 106–118
8. Lenat, D., Davis, R., Doyle, J., Genesereth, M., Goldstein, I., Schrobe, H.: Reasoning about reasoning. In Hayes-Roth, F., Waterman, D.A., Lenat, D.B., eds.: Building Expert Systems. Addison-Wesley, London (1983) 219–239
9. Davis, R., Buchanan, B.: Meta level knowledge. In Hayes-Roth, F., Waterman, D.A., Lenat, D.B., eds.: Rule-Based Expert Systems. Addison-Wesley, London (1985) 507–530
10. Davis, R.: Expert systems: Where are we? and where do we go from here? AI Magazine **3** (1982) 3–22
11. McLaren, B.M., Ashley, K.D.: Helping a cbr program know what it knows. In Aha, D., Watson, I., eds.: 4th International Conference on Case-Based Reasoning (ICCBR-2001). Volume 2080 of LNAI., Springer (2001) 377–391
12. Doyle, D., Cunningham, P., Bridge, D., Rahman, Y.: Explanation oriented retrieval. In Funk, P., González-Calero, P.A., eds.: 7th European Conference on Case-Based Reasoning (ECCBR 2004). Volume 3155 of LNAI., Springer (2004) 157–168

13. Nugent, C., Cunningham, P.: A case-based explanation system for black-box systems. Artificial Intelligence Review (to appear) (2005)
14. McSherry, D.: Explaining the pros and cons of conclusions in cbr. In Funk, P., González-Calero, P., eds.: 7th European Conference on Case-Based Reasoning (ECCBR-2004). Volume 3155 of LNAI., Springer (2004) 317–330
15. Massie, S., Craw, S., Wiratunga, N.: A visualisation tool to explain case-base reasoning solutions for tablet formulation. In: 24th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence (AI-2004). LNCS, Springer (2004)
16. Delany, S., Cunningham, P., Coyle, L.: An assessment of case-based reasoning for spam filtering. Procs. of 15th Irish Conference on Artificial Intelligence and Cognitive Science (2004) 9–18
17. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L.: A case-based technique for tracking concept drift in spam filtering. In Macintosh, A., Ellis, R., Allen, T., eds.: Applications and Innovations in Intelligent Systems XII, Procs. of AI 2004, Springer (2004) 3–16
18. Delany, S.J., Cunningham, P.: An analysis of case-based editing in a spam filtering system. In Funk, P., P.González-Calero, eds.: 7th European Conference on Case-Based Reasoning (ECCBR 2004). Volume 3155 of LNAI., Springer (2004) 128–141
19. P.Pantel, Lin, D.: Spamcop: A spam classification and organisation program. In: Procs of Workshop for Text Categorisation, AAAI-98. (1998) 95–98
20. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A bayesian approach to filtering junk email. In: Procs of Workshop for Text Categorisation, AAAI-98. (1998) 55–62
21. Androutsopoulos, I., J.Koutsias, Chandrinos, G., Paliouras, G., Spyropoulos, C.: An evaluation of naive bayesian anti-spam filtering. In Potamias, G., Moustakis, V., van Someren, M., eds.: Procs of Workshop on Machine Learning in the New Information Age, ECML 2000. (2000) 9–17
22. Schneider, K.: A comparison of event models for näive bayes anti-spam e-mail filtering. In: 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL'03). (2003) 307–314
23. Zhang, L., Zhu, J., Yao, T.: An evaluation of statistical spam filtering techniques. ACM Transactions on Asian Language Information Processing (TALIP) **3** (2004) 243–269
24. Drucker, H., Wu, D., Vapnik, V.: Support vector machines for spam categorisation. IEEE Transactions on Neural Networks **10** (1999) 1048–1055
25. Androutsopoulos, I., Paliouras, G., Michelakis, E.: Learning to filter unsolicited commercial email. Technical Report 2004/02, NCSR "Demokritos" (2000)
26. Kolcz, A., Alspector, J.: Svm-based filtering of email spam with content-specific misclassification costs. In: TextDM'2001 (IEEE ICDM-2001 Workshop on Text Mining), IEEE (2001) 123–130
27. Michelakis, E., Androutsopoulos, I., Paliouras, G., Sakkis, G., Stamatopoulos, P.: Filtron: A learning-based anti-spam filter. In: 1st Conference on Email and Anti-Spam (CEAS 2004). (2004)