2009-01-01

Sampling with Confidence: Using k-NN Confidence Measures in Active Learning

Rong Hu
Dublin Institute of Technology, rong.hu@dit.ie

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

Brian MacNamee
Dublin Institute of Technology, brian.macnamee@dit.ie

Sampling with Confidence: Using *k*-NN Confidence Measures in Active Learning

Rong Hu, Sarah Jane Delany and Brian Mac Namee

Dublin Institute of Technology, Dublin, Ireland
rong.hu@dit.ie,sarahjane.delany@dit.ie,brian.macnamee@dit.ie

Abstract. Active learning is a process through which classifiers can be built from collections of unlabelled examples through the cooperation of a human oracle who can label a small number of examples selected as *most informative*. Typically the most informative examples are selected through *uncertainty sampling* based on classification scores. However, previous work has shown that, contrary to expectations, there is not a direct relationship between classification scores and classification confidence. Fortunately, there exists a collection of particularly effective techniques for building measures of classification confidence from the similarity information generated by *k*-NN classifiers. This paper investigates using these confidence measures in a new active learning sampling selection strategy, and shows how the performance of this strategy is better than one based on uncertainty sampling using classification scores.

1 Introduction

Active Learning (AL) [1] attempts to overcome the problem that in supervised learning labelled datasets can be difficult or expensive to obtain. AL attempts to build labelled datasets by selecting only the *most informative* examples in a larger unlabelled example set for labelling by an *oracle*, typically a human expert. The most common selection strategy for picking these most informative examples is *uncertainty sampling* [2] in which examples are selected based on the certainty with which a classifier can classify them.

The typical approach to uncertainty sampling is to use the output of a ranking classifier that produces numeric *classification scores* (e.g. *k*-Nearest Neighbour, Naïve Bayes or Support Vector Machines) as a measure of *classification confidence*. However, Delany et al. [3] have shown that there is not a direct relationship between classification scores and classification confidence. This suggests that AL selection strategies that measure certainty using factors other than classification scores would be more effective. Delany et al. [3] show that an aggregate of five basic confidence measures used with *k*-NN classifiers are particularly effective in estimating classification confidence. In this paper we investigate an AL selection strategy based on these confidence measures, and evaluate whether this performs better than a selection strategy based on classification scores.

Section 2 will discuss AL in more detail and provide examples of how AL has been used in Case-Based Reasoning (CBR). Section 3 will then discuss the

confidence measures that will be used in our selection strategy. Section 4 will describe our overall AL approach including the details of how the confidence measures are integrated into the selection process. This *confidence-based selection strategy* has been evaluated against a strategy based on classification scores using a number of text datasets and the results of these evaluations will be presented and discussed in Section 5. Finally, we conclude and outline our intended directions for future work in Section 6.

2 AL and CBR

The principle aim of AL is to build quality classifiers using as few labelled training examples as possible. The most common AL scenario is *pool-based* AL [2, 4] which assumes that the learner has access to a large pool of unlabelled examples from the beginning of the process and this is the scenario considered in this work.

The pool-based AL process begins by selecting a small number of examples from the pool, that the oracle is asked to label to form the initial *labelled set*, or *case base*. The labelled set is used to build a classifier which in turn is used to calculate the *informativeness* of each example remaining in the pool. The informativeness of an example is a measure of how useful to the training process it would be to solicit the oracle for a label for that example. The most informative examples from the pool are then labelled by the oracle, removed from the pool, and added to the labelled set. A new classifier is then built using the labelled set and the process iterates until a stopping criteria is reached — for example the oracle exceeds a label budget, or labelling further examples is not deemed sufficiently informative.

The predominant research issue in pool-based AL is determining the best selection strategy for choosing those examples most informative to the training process. Uncertainty sampling, first proposed by Lewis and Gale [2], is the most widely used approach. Uncertainty sampling uses ranking classifiers that associate a certainty score with each classification. The certainty score, $P(C|e)$, indicates the certainty of the system that example e belongs to class C. Certainty scores fall into the range $[0, 1]$ where 0 indicates that the system is certain that the example does not belong to the class in question, and 1 indicates that it is certain that it does. At each iteration of the AL process the certainty scores of each example are computed and those for which classifications are least certain (i.e. those with scores closest to 0.5) are selected for labelling. The philosophy behind this approach is that a better classifier can be built by reducing the uncertainty in the dataset. The advantages of the uncertainty sampling approach include its simplicity and fast execution speed.

Other selection strategies include *version space reduction* [1] in which examples that best reduce the version space associated with a classifier are selected; Query-By-Committee (QBC) [5] in which the examples that give rise to the most disagreement in an ensemble of classifiers are selected; the use of Expectation-

Maximization (EM) [6]; and the inclusion of density information to select those examples in most densely populated regions of the example space [7].

Although just about any classifier can be used in the AL process, the CBR approach to classification is particularly attractive as certainty scores are easily calculated, and the repeated classifier retraining required in AL is especially efficient — new examples are simply added to the case base. Two of the earliest examples of using CBR and AL together were by Hasenjager & Ritter [8] who contrasted local learning approaches against global ones; and Lindenbaun et al. [9] who developed AL strategies for nearest neighbour classifiers. More recent examples of the use of CBR and AL together include their combination for the semantic labelling of text [10]; solving problems in drug development [11]; creating case retention strategies for CBR [12]; and supervised network intrusion detection [13].

Earlier work by Li et al. [14] proposed a *confidence-based AL* approach to image segmentation which calibrates the classification scores of SVM classifiers to classification confidence [15]. The overall benefits of using classifiers properly calibrated to produce class-membership probabilities is discussed in [16].

3 Confidence Measures

To attach confidence to classification scores Delany et al. [3] proposed five basic confidence measures that can be used with k-NN classifiers and showed that an aggregate of these is particularly effective. The use of aggregate measures is also supported by the work of Cheetham & Price [17] who presented a similar result, using different measures.

The objective of the k-NN measures is to assign higher confidence to those examples that are 'close' (i.e. with high similarity) to examples of its predicted class, and are 'far' (i.e. low similarity) from examples of a different class. The closer a target example is to examples of a different class, the higher the chance that the target example is lying near or at the decision surface. Whereas the closer an example is to other examples of the same class, the higher the likelihood that it is further from the decision surface. All the k-NN measures perform some calculation on a ranked list of neighbours of a target example using a combination of:

- the distance between an example and its nearest neighbours ($NN_i(t)$ denotes the ith nearest neighbour of example t),
- the distance between the target example t and its nearest like neighbours ($NLN_i(t)$ denotes the ith nearest *like* neighbour to example t),
- the distance between an example and its nearest unlike neighbours ($NUN_i(t)$ denotes the ith nearest *unlike* neighbour to example t).

Preliminary experiments using the five measures proposed in [3] showed a high correlation between three of them, and so we chose to use the three of the five that are least correlated in our evaluations. Full details on each measure can be found in [3].

Average NUN Index (M1) The Average Nearest Unlike Neighbour Index (Avg NUN Index) is a measure of how close the first k NUNs are to the target example t as given in Equation 1.

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

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

Similarity Ratio (M2) The Similarity Ratio measure calculates the ratio of the similarity between the target example t and its k NLNs to the similarity between the target example and its k NUNs, as given in Equation 2.

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

where $Sim(a,b)$ is the similarity between examples a and b and ϵ is a smoothing value to allow for situations where an example may have no NLNs or NUNs ($\epsilon = 0.0001$ is used in all of our evaluations).

Similarity Ratio Within K (M3) 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 example t, it uses only the NLNs and NUNs from the first k neighbours. It is defined in Equation 3.

$$SimRatioK(t,k) = \frac{\sum_{i=1}^{k} Sim(t, NN_i(t))\delta_{t,NN_i(t)}}{\epsilon + \sum_{i=1}^{k} Sim(t, NN_i(t))(1 - \delta_{t,NN_i(t)})} \quad (3)$$

where $Sim(a,b)$ is as above, δ_{ab} is Kronecker's delta where $\delta_{ab} = 1$ if the class of a is the same as the class of b and 0 otherwise, and ϵ is a smoothing value to allow for situations where an example may have no NUNs ($\epsilon = 0.0001$ is used).

4 Approach

The important aspects of the AL process are: forming the initial case base, building a classifier to label all examples in the pool, and selecting examples for labelling by the oracle. This section will describe our approach to each of these (further details are available in [18]).

4.1 Initial Case Base Selection and Classifier

The AL process begins with a small set of examples labelled by the oracle which is the initial case base. While this selection can be performed at random, it offers an opportunity to prime the AL process through informed selection. Previous work has shown that using clustering to select the initial case base gives better

results than random selection [19] . However, this can lead to highly inconsistent results over many trials as clustering is quite unstable, especially when dealing with high dimensional textual data. For this reason, we use the *furthest-first* initialisation algorithm [20] which is deterministic and will always return the same initial case base for a given dataset.

At every iteration of the AL process all of the unlabelled examples remaining in the pool are classified using a classifier trained on the examples labelled by the oracle so far. In our system the classifier used to do this is a k-NN classifier using distance weighted voting [21] with $k = 5$.

4.2 The Confidence-Based Selection Strategy

Before any of the confidence measures described in Section 3 can be used to calculate classification confidence it is necessary to identify for each measure a confidence threshold value for each of the possible classes. Predictions with confidence values higher than the predicted class's threshold are considered *confident*, while those with values below are considered *non-confident*. The threshold value for a particular class is that value that results in the highest proportion of correctly predicted examples of a particular class when there were no incorrect predictions. The confidence thresholds are referred to as $thres_{ij}$ for each confidence measure M_i $(i = 1 \ldots n)$, and each class $j = 1 \ldots c$. Specific details on the approach used for setting the threshold level for a class are described in [3].

Our ACM Selection (ACMS) strategy aggregates the three confidence measures used into a new selection strategy. First each example e_k in the pool is classified using the initial case base and the value for each confidence measure m_{ik} is calculated. Based on the predicted class of the example the appropriate threshold value is checked for each of the measures. If any one of the measures indicates confidence, i.e. $m_{ik} > thres_{ij}$ for any $i = 1 \ldots n$ and $j =$ *the predicted class*, then we consider that the example has been classified with confidence, and it gets added to the *confident set*. Otherwise, it gets added to the *non-confident set*.

A single $rank(e_k)$ value is associated with each e_k example. For an example e_k classified with confidence, $rank(e_k)$ is assigned the value that indicates most confidence, i.e. $rank(e_k) = max(m_{ik})$ for those M_i's that indicate confidence; while the one used for an example in the non-confident set should be the m_{ik} that indicates least confidence (i.e. $rank(e_k) = min(m_{ik})$ for those M_i's that do not indicate confidence). Different strategies for combining confidence measures were considered in preliminary experiments which showed the min/max combination to be consistently best.

In order to be able to compare m_{ik} across different confidence measures, the values of m_{ik} for each M_i are normalised using statistical normalisation after performing a log transformation to correct those with skewed distributions.

Once all pool examples have been classified, the one that the classifier is least confident of is the example in the non-confident set that has the smallest $rank(e_k)$ value. If the non-confident set is empty, the least confident example is the one in the confident set with the smallest $rank(e_k)$ value. This is the example

that is presented to the oracle for labelling before the process repeats until the stopping criteria is met. The algorithm for our ACMS strategy is presented in Algorithm 1.

5 Evaluation

The two objectives to the evaluations described here were to confirm the superiority of using an aggregate confidence measure over using single confidence measures; and to compare the performance of our ACMS approach with an uncertainty sampling approach based on classification scores.

In order to conduct a comprehensive analysis, we tested various algorithms on seven datasets: a spam dataset [22]; four binary classification textual datasets derived from the 20-Newsgroup collection[1]; and two binary classification datasets from the Reuters collection[2]. The properties of each dataset and the average accuracy achieved in five iterations of 10-fold cross validation using a 5-NN classifier are shown in Table 1 (accuracies are included as an indication of the difficulty of each classification problem). Each dataset was pre-processed to remove stop-words and stemmed using Porter stemming.

To evaluate the system, we simulated the labelling process and compared the results with the actual labels in each dataset. The accuracy of the labelling is used to evaluate the performance of the system, calculated as $Accuracy = C/N$ where N is the number of examples in the dataset (including the examples in the initial case base) and C is the number of correctly labelled examples. Both manually and automatically labelled examples are included in this calculation to avoid the accuracy figure becoming unstable in the latter stages of the process. The accuracy is recorded after each manual labelling.

At present we use a simple stopping criterion that allows the human oracle to only provide a specified number of labels, a *label budget*. We set the label budget to 110 which includes 10 initial labels and 100 during the AL process.

We evaluated the performance of sampling selection strategies using each individual confidence measure and using the aggregation of the measures on all of the datasets. Illustrative results on two datasets are shown in Figure 1. The results indicate that ACMS is at least as good as but generally dominates the individual measures. Furthermore, we found that the ACMS strategy is more stable than using individual measures.

Figure 2 shows the results of comparing the ACMS strategy with the more typical Uncertainty Sampling (US) strategy using classification scores. A Random Sampling (RS) strategy, which randomly picks the example to label, is also included as a baseline. The accuracy graph for the ACMS strategy dominates the graph for the RS strategy in all cases, and the graph for the US strategy for five (WinXwin, Comp, Vehicle, Reuters, Spam) of the seven datasets. Interestingly, across all ACMS experiments the average *effectiveness* — how often

[1] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html
[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Input: An initial labelled case base $\mathcal{CB}$, an unlabelled pool $\mathcal{P}$ of p examples, a k-NN classifier $\mathcal{C}$ for classes $1 \dots c$, a stopping criterion $\mathcal{S}$, a batch size b, a set of confidence measures M_i, $i = 1 \dots n$

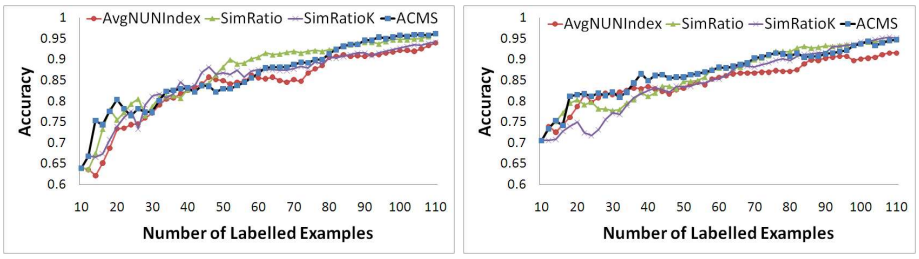
Output: A labelled case base

while *S is not met* **do**
  foreach *confidence measure* $M_i, i = 1 \dots n$ **do**
    Identify the threshold: find $thres_{ij}$ and k_{ij}, for $j = 1 \dots c$;
  end
  foreach *example* $e_k \in \mathcal{P}$ **do**
    $ConfSet = \emptyset$, $NonConfSet = \emptyset$, $Selected = \emptyset$;
    Classify e_k using the classifier $\mathcal{C}$;
    Calculate m_{ik} using k_{ij} for $i = 1 \dots n$ and $j =$ predicted class of e_k;
    if $m_{ik} > thres_{ij}$ *for any* $i = 1 \dots n$ *and* $j =$ *predicted class of* e_k **then**
      $ConfSet = ConfSet + e_k$;
      Set the ranking score: $rank(e_k) = max(m_{ik})$;
    else
      $NonConfSet = NonConfSet + e_k$;
      Set the ranking score: $rank(e_k) = min(m_{ik})$;
    end
  end
  foreach $l, l = 1 \dots b$ **do**
    if $NonConfSet == \emptyset$ **then**
      $Selected = Selected + e$ where
      $rank(e) = min(rank(e_k)), e_k \in ConfSet$;
    else
      $Selected = Selected + e$ where
      $rank(e) = min(rank(e_k)), e_k \in NonConfSet$;
    end
  end
  Label each $e_l \in Selected$;
  $\mathcal{CB} = \mathcal{CB} \cup Selected$, $\mathcal{P} = \mathcal{P}/Selected$;
end

Algorithm 1: The algorithm for the Aggregated Confidence Measure Selection (ACMS) strategy

Table 1. Benchmark Datasets.

Dataset	**Task**	**Examples**	**Features**	**Accuracy**
WinXwin	comp.os.ms-windows.misc vs. comp.windows.x	496	8557	91.14%
Comp	comp.sys.ibm.pc.hardware vs. comp.sys.mac.hardware	500	7044	85.56%
Talk	talk.religion.misc vs. alt.atheism	500	9000	93.92%
Vehicle	rec.autos vs. rec.motorcycles	500	8059	92.96%
Reuters	acq vs. earn	500	3692	89.56%
RCV1	g151 vs. g158	500	6135	95.36%
Spam	spam vs. non-spam	500	18888	96.80%



(a) Talk Dataset (b) Vehicle Dataset

Fig. 1. Comparison of Individual Confidence Measures and the ACM as the Sampling Selection Strategy

the $rank(e_k)$ given to a case by ACMS is determined by a particular confidence measure — of M1, M2 and M3 are 38.87% 34.57% and 26.56% respectively.

6 Conclusions and Future Work

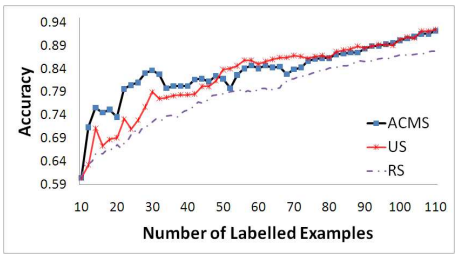
In this paper, we propose a new selection strategy for active learning using k-NN based confidence measures. The experimental results show that an aggregated confidence measure is more effective than single confidence measures. We also show that ACMS generally outperforms the more typical uncertainty sampling approach using classification scores. Although the algorithm is computationally expensive, the use of k-NN classifier makes it possible to cache and re-use case similarities making ACMS computationally feasible, even for large datasets. Furthermore, a larger batch size b can be used to reduce the computational load.

There are three main areas we intend to explore in the future. Firstly, the furthest-first method may include outliers in the initial case base which may limit the exploitation capability of the AL process. To solve this problem, more sophisticated initial case base selection strategies will be considered. However, the stability problems with clustering textual data must be overcome.

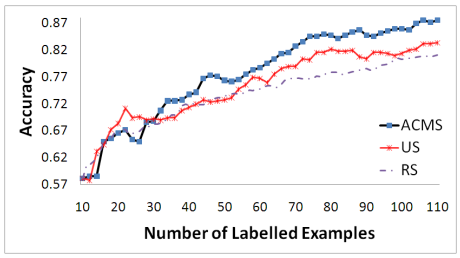
Secondly, ACMS focuses on refining the decision boundary. However, there is a balance to be achieved between this and the exploration of new regions in the decision space that the current classifier may not perform well on. We will consider using additional information, such as density information to allow our AL process to explore more while maintaining good performance.

Finally, the work described here has focussed on binary classification, but we intend to extend this to multi-class situations in the near future.

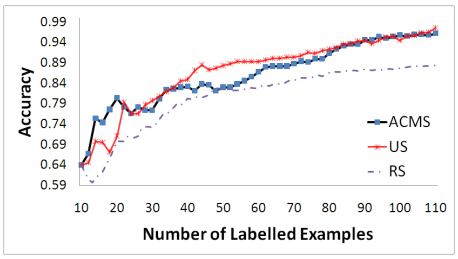
Acknowledgments. This material is based upon works supported by the Science Foundation Ireland under Grant No. 07/RFP/CMSF718.



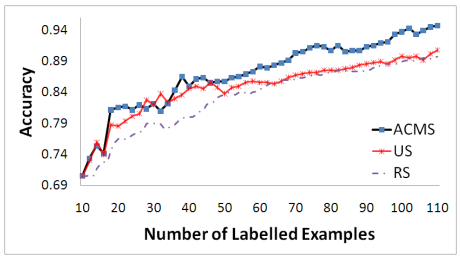
(a) WinXwin Dataset



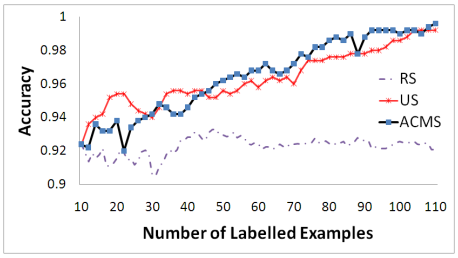
(b) Comp Dataset



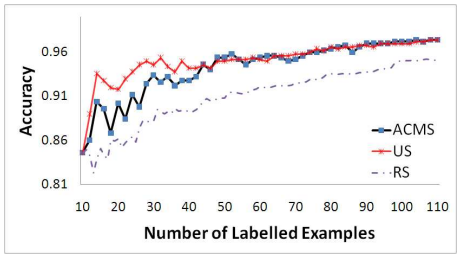
(c) Talk Dataset



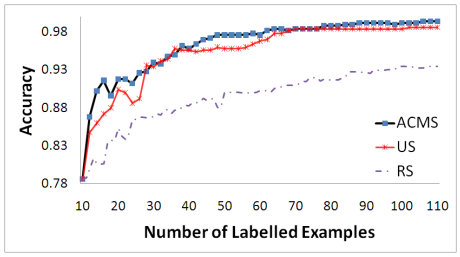
(d) Vehicle Dataset



(e) Reuters Dataset



(f) RCV1 Dataset



(g) Spam Dataset

Fig. 2. Comparison of ACMS, US and RS selection strategies

References

1. Tong, S.: Active Learning:Theory and applications. PhD thesis, Computer science department, Stanford University (August 2001)
2. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Proc 17th annual International ACM SIGIR conference on Research and Development in Information Retrieval, Springer-Verlag NY (1994) 3–12
3. Delany, S.J., Cunningham, P., Doyle, D.: Generating estimates of classification confidence for a case-based spam filter. In: Proc of ICCBR '05. Volume 3620 of LNAI., Springer (2005) 170–190
4. McCallum, A.K., Nigam, K.: Employing EM and pool-based active learning for text classification. In: Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann (1998)
5. H.S.Seung, M.Opper, H.Sompolinsky: Query by committee. In: In Proceedings of the Fifth Workshop on Computational Learning Theory. (1992) 287–294
6. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Learning to classify text from labeled and unlabeled documents. In: Proc of AAAI '98. (1998) 792–799
7. Xu, Z., Yu, K., Tresp, V., Xu, X., Wang, J. In: Representative Sampling for Text Classification Using Support Vector Machines. Springer (2003) 11
8. Hasenjager, M., Ritter, H.: Active learning with local models. In: Neural Processing Letters. Volume 7. (1998)
9. Lindenbaum, M., Markovitch, S., Rusakov, D.: Selective sampling for nearest neighbor classifiers. In: Proceedings of AAAI '99. (1999) 366–371
10. Mustafaraj, E., Hoof, M., Freisleben, B.: Learning semantic annotations for textual cases. In: In Textual Case-based Reasoning Workshop at the 6th ICCBR. (2005)
11. Cebron, N., Berthold, M.R.: An adaptive multi objective selection strategy for active learning. Technical report, Universität Konstanz (2007)
12. Ontañón, S., Plaza, E.: Collaborative Case Retention Strategies for CBR Agents. In: Case-Based Reasoning Research and Development. Springer (2003)
13. Li, Y., Guo, L.: An active learning based TCM-KNN algorithm for supervised network intrusion detection. Computers and Security **26** (2007) 459–467
14. Li, M., Sethi, I.K.: Confidence-based active learning. IEEE Trans. Pattern Anal. Mach. Intell. **28** (2006) 1251–1261
15. Ma, A., Patel, N., Li, M., Sethi, I. In: Confidence Based Active Learning for Whole Object Image Segmentation. Springer Berlin / Heidelberg (2006) 753–760
16. Cohen, I., Goldszmidt, M.: Properties and benefits of calibrated classifiers. In: 8th European Conference on Principles and Practice of Knowledge Discovery in Databases. (2004) 125–136
17. Cheetham, W., Price, J.: Measures of solution accuracy in case-based reasoning systems. In: Proc of ECCBR '04. (2004) 106–118
18. Hu, R., Mac Namee, B., Delany, S.J.: Sweetening the dataset: Using active learning to label unlabelled datasets. In: Proceedings of the the 19th Irish Conference on Artificial Intelligence and Cognitive Science (AICS '08). (2008)
19. Kang, J., Ryu, K., Kwon, H.: Using Cluster-Based Sampling to Select Initial Training Set for Active Learning in Text Classification. In: Advances in Knowledge Discovery and Data Mining. Volume 3056. Springer (2004) 384–388
20. Greene, D.: A State-of-the-Art Toolkit for Document Clustering. PhD thesis, University of Dublin, Trinity College (2006)
21. Mitchell, T.: Machine Learning. McGraw Hill (1997)
22. Delany, S.J., Cunningham, P., Tsymbal, A., Coyle, L. In: A Case-Based Technique for Tracking Concept Drift in Spam Filtering. Springer London (2005) 3–16