How to Select Information That Matters

A Comparative Study on Active Learning Strategies for Classification

Christian Beyer
University Magdeburg
Universitätsplatz 2, Building 29
39016 Magdeburg, Germany
christian.beyer@st.ovgu.de

Georg Krempl
Knowledge Management &
Discovery, Univ. Magdeburg
Universitätsplatz 2, Building 29
39016 Magdeburg, Germany
georg.krempl@ovgu.de

Vincent Lemaire
Orange Labs
2 avenue Pierre Marzin
22300 Lannion, France
vincent.lemaire@orange.com

ABSTRACT

Facing ever increasing volumes of data but limited human annotation capabilities, active learning strategies for selecting the most informative labels gain in importance. However, the choice of an appropriate active learning strategy itself is a complex task that requires to consider different criteria such as the informativeness of the selected labels, the versatility with respect to classification algorithms, or the processing speed. This raises the question, which combinations of active learning strategies and classification algorithms are the most promising to apply. A general answer to this question, without application-specific, label-intensive experiments on each dataset, is highly desirable, as active learning is applied in situations with limited labelled data. Therefore, this paper studies several combinations of different active learning strategies and classification algorithms and evaluates them in a series of comparative experiments.

CCS Concepts

•Theory of computation → Active learning;

Keywords

Active Learning; Selective Sampling; Uncertainty Sampling; Probabilistic Active Learning

1. INTRODUCTION

While the volumes of data are constantly increasing [9], human annotation and supervision capacities remain limited. This raises the need for approaches that help in the efficient allocation of these capacities [15]. Active machine learning [22] provides such approaches for determining and selecting the most valuable information. In classification tasks, this corresponds to selecting the instance from a set of candidates, whose label is expected to improve a classifier's performance the most [23]. Given the large number of approaches that have been proposed in literature, the choice of the most appropriate active learning strategy constitutes itself a complex task: multiple criteria such as the informativeness of the selected labels, the versatility of the approach with respect to classification algorithms, or the processing speed of the approach need to be considered.

Active learning is applied in situations with very limited initial labelled data. Thus, knowing the overall most promising combinations of active learning strategies and classification algorithms without performing application-specific, label-intensive experiments on each novel dataset is highly desirable. This paper addresses this question by providing results of an experimental performance comparison of several combinations of popular classification algorithms and active learning strategies. In Section 2, related surveys are reviewed before discussing selected active learning strategies. These strategies are then experimentally evaluated in Section 3, before concluding in Section 4.

2. ACTIVE LEARNING APPROACHES

This paper addresses the *pool-based* [23, 6] active learning scenario for *binary* classifiers, where an active classifier has access to a pool of unlabelled instances $\mathcal{U} = \{(x, .)\}$. Repeatedly, *the best instance* $(x^*, .) \in \mathcal{U}$ is selected, its label y^* is requested from an oracle, and it is moved from $\mathcal{U}$ to the set of labelled instanced $\mathcal{L} = \{(x, y)\}$ to retrain the classifier. In particular, this paper focuses on a sequential labelling scenario, in contrast to batch-based active learning where multiple instances are labelled in one iteration [10]. Various existing approaches for this scenario are surveyed in [22, 6, 23, 8]. The technical report [22], the machine learning encyclopedia entry [6] on active learning, and more recently the textbook [23] provide an introduction to active learning, as well as a good overview on various families of active learning approaches. While comparing theoretical aspects of the different approaches, they do not include an empirical evaluation. Recently, [8] surveys different approaches based on uncertainty sampling and instance correlation and provide a categorisation of different approaches. However, the performance analysis in that review is limited to runtime evaluations, thus leaving the question on the classification performance of different approaches open. An experimental classification performance evaluation and comparison of some approaches was done in the active learning challenge, published in [11]. It is remarked therein that a key to success in active learning is handling the trade-off between *exploration* and *exploitation*: the former samples in regions with yet little collected information, the latter investigates re-

i-KNOW '15, October 21 - 23, 2015, Graz, Austria

ISBN 978-1-4503-3721-2/15/10…$15.00

DOI: http://dx.doi.org/10.1145/2809563.2809594

gions where the current model suspects the decision boundary. According to [11, page iv], the overall winners use combinations of random and uncertainty sampling to tackle this trade-off.

This comparative study's focus are *fast* approaches that are *usable with any classification technique*. Building on the results above, we compare random sampling, uncertainty sampling, and a combination of both that tackles exploration-exploration. In addition to these *popular* approaches, we include the *very recently* proposed probabilistic active learning approach, which implicitly balances exploration-exploration. We now briefly review these approaches, before continuing with the experimental evaluation in the next chapter.

2.1 Random Sampling

A simple and fast baseline is *random sampling*, where instances are selected at random with equal probability. Despite the simplicity of this *purely explorative* strategy, it has been shown to be difficult to be beaten consistently [2] and is one of the most popular active learning baselines [11].

2.2 Uncertainty Sampling

A very popular active learning strategy is *uncertainty sampling* [17], which is frequently used as baseline (e.g. in the active learning competition [11]). This is a purely exploitative strategy that relies on the current model to compute so-called *uncertainty measures*. These serve as proxies for a candidate's impact on the classification performance, and the candidate with the highest uncertainty is selected for labelling. In the seminal work of [17], a probabilistic classifier is used on a candidate to compute the posterior of its most likely class. The absolute difference between this posterior estimate and 0.5 is used as uncertainty measure (lower values denoting higher uncertainty). The formula for picking x^*_{LC} is the following according to [22]:

$$x^*_{LC} = \underset{x}{argmax}\ (1 - P_\theta(\hat{y} \mid x)) \tag{1}$$

x^*_{LC} is the instance from the pool of unlabelled data D_u which our model θ is least confident in while $\hat{y}$ is the class for which the model calculated the highest posterior estimate so $\hat{y} = \underset{y}{argmax}\ P_\theta(y \mid x)$. In addition to this confidence-based uncertainty measure, other common measures [23] are entropy or the margin between a candidate and the decision boundary. However, [22] notes that for *binary* classification problems classifiers the measures margin, confidence and entropy result in the same ranking and querying of instances.

This strategy is easy to implement and computationally efficient, having an asymptotic time complexity of $O(|\mathcal{U}|)$. Thus, it is also usable in time critical applications, or in big data scenarios with large numbers of unlabelled instances, or on fast data streams [27]. Nevertheless, a known disadvantage [25] of uncertainty sampling is that these proxies do not consider the number of similar instances on which the posterior estimates are made or the decision boundaries are drawn. The reported results of empirical evaluations are somewhat inconclusive, with some authors (e.g. [4, 20, 13]) reporting even worse performance on some data sets than random sampling. Its major problems are that it can get stuck in regions with high Bayesian error, especially when the data is not linearly separable. Additionally, as this strategy queries instances that are close to the current decision boundary, it is prone to missing subconcepts if the initial decision boundary is unfavourable for the data. Furthermore, it can also tend to query outliers which are not representative for the underlying distribution. Figure 1 illustrates some of the problems while the work of [26] and [25] discuss the issue of querying outliers. Following next is a short description of a mixed strategy that combines random and uncertainty sampling.

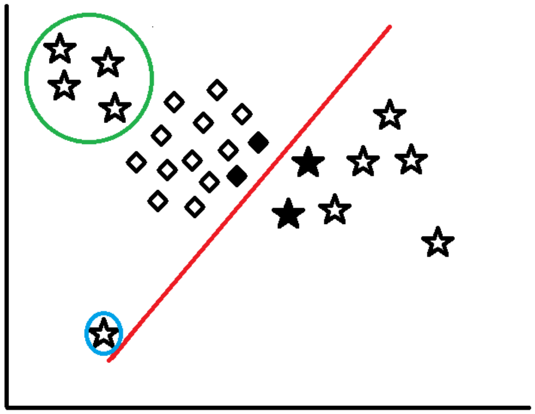

Figure 1: This figure shows a configuration during the active learning process on a two-class problem. The red line is the current decision boundary and the coloured stars and squares are the labelled instances. The stars on the top left are a subconcept which will probably be missed by uncertainty sampling because those instances are far away from the decision boundary which means the classifier is very confident in their prediction. The star with the blue circle on the other hand is an outlier that is very close to the current decision boundary and therefore highly likely to be queried for labelling.

2.3 Semi-Random Sampling

The combination of uncertainty and random sampling to combine exploitation and exploration has been suggested for example in [16, 11, 27]. Most recently, [27] uses a mix of random and uncertainty sampling on streams to tackle the problem of missing exploration with uncertainty measures. This is especially useful in stream-based active learning where concepts and thus the optimal decision boundary might change over time. The authors speculate that in a static scenario it is likely that uncertainty sampling beats the mixed strategies, as the decision boundary does not change over time. We investigate this hypothesis by studying the performance of a mixed strategy for pool-based active learning, which switches between uncertainty and random sampling. This strategy alternately applies random sampling and uncertainty sampling, beginning with the initial instance being selected randomly from the unlabelled pool D_u. This strategy has the same asymptotic time complexity as uncertainty sampling, but is faster by a constant factor due to using random selection half of the time.

2.4 Probabilistic Active Learning

Probabilistic active learning is a novel approach [14] that directly optimises a performance measure like accuracy, using statistically sound methods to guide the degree of exploitation and exploration. In this aspect it is comparable to error reduction approaches (proposed in [19]), while still having linear complexity like the fast uncertainty methods. For binary classification with Parzen Window classifiers, it was already shown that probabilistic active learning achieves comparable or superior performance than error reduction.

Probabilistic active learning builds on the smoothness assumption commonly used in semi-supervised learning [5], which suggests that the influence of an instance on the classification process is the highest in its neighbourhood:

> *Semi-supervised smoothness assumption*: If two points x_1, x_2 in a high-density region are close, then so should be the corresponding outputs y_1, y_2.

Therefore, the method proposed in [14] considers within the neighbourhood of an instance the number of labelled instances n and the share of positive labels therein $\hat{p} = \frac{n_+}{n}$. These two values are the necessary label statistics $ls = (n, \hat{p})$, which should be provided by the classifier being used. As the real posterior p of that neighbourhood and the label realisation y of the instance under consideration are unknown, they are modelled as hidden variables. The so-called probabilistic gain is calculated as the expectation over all possible realisations of p and y of the gain in classification performance. This gain is then weighted with the density in an instance's neighbourhood, considering the union of the labelled and unlabelled pool $D_u \cup D_l$, in order to prefer dense regions and avoid outliers. This probabilistic gain calculation models the true posterior p within the neighbourhood as being Beta-distributed, and the label realisation y as being Bernoulli-distributed with p as an input. Thus, the number of positive instances in the neighbourhood $n_+ = n \cdot \hat{p}$ is binomially distributed.

For accuracy or misclassification loss, a closed-form solution for computing the probabilistic gain is given in [13], which is called optimised probabilistic active learning (OPAL). The gain G can be written as:

$$G_{\mathrm{OPAL}}(n, \hat{p}, \tau, m) = \frac{(n+1)}{m} \cdot \left(\begin{array}{c} n \\ n \cdot \hat{p} \end{array} \right) \cdot \tag{2}$$

$$\left(\mathrm{I}_{ML}(n, \hat{p}, \tau, 0, 0) - \sum_{k=0}^{m} \mathrm{I}_{ML}(n, \hat{p}, \tau, m, k) \right) \tag{3}$$

Here, τ is the cost of a false positive (normalised such that the costs of a false positive and a false negative add up to one), m denotes how many labels can be purchased in a given neighbourhood, and $I_{ML}(n, \hat{p}, \tau, m, k)$ is a function that is proportional to the expected misclassification loss in case k positive labels were among the m purchased ones:

$$\mathrm{I}_{ML}(n, \hat{p}, \tau, m, k) = \left(\begin{array}{c} m \\ k \end{array} \right) \cdot \tag{4}$$

$$\begin{cases} (1-\tau) \cdot \frac{\Gamma(1-k+m+n-n\hat{p})\Gamma(2+k+n\hat{p})}{\Gamma(3+m+n)} & \frac{n\hat{p}+k}{n+m} < \tau \\ (\tau-\tau^2) \cdot \frac{\Gamma(1-k+m+n-n\hat{p})\Gamma(1+k+n\hat{p})}{\Gamma(2+m+n)} & \frac{n\hat{p}+k}{n+m} = \tau \\ \tau \cdot \frac{\Gamma(2-k+m+n-n\hat{p})\Gamma(1+k+n\hat{p})}{\Gamma(3+m+n)} & \frac{n\hat{p}+k}{n+m} > \tau \end{cases} \tag{5}$$

Here, $\Gamma(z)$ is Legendre's gamma function (see e.g. [18, p. 206]).

For computing the probabilistic gain, the label statistics of an instance's neighbourhood are required, which consist of total number of labels (n) and the share of positives therein ($\hat{p}$). These statistics need to be estimated. In [13, 14], it is argued that using estimates provided by a probabilistic classifier might be favourable to using kernel frequency estimates as substitutes. For investigating this experimentally, different ways of computing the label statistics for different classifiers need to be specified.

When using kernel frequency estimates as substitutes, [14] propose the following formula that employs Gaussian kernels with a bandwidth of σ:

$$LC(x, \mathcal{L}) \approx \sum_{x_i \in \mathcal{L}} \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2} \right) \tag{6}$$

The total number of labels is then $n = LC(x, \mathcal{L})$, where $\mathcal{L}$ is the set of all labelled instances, and the the share of positives is $\hat{p} = LC(x, \mathcal{L}_+)/LC(x, \mathcal{L})$, where $\mathcal{L}_+$ is the subset of labelled positive instances.

For *Parzen-Window Classifiers* [4], which use kernel density estimates for computing an instance's posterior probabilities, the kernel frequency estimates above for $\hat{p}$ are identical to the classifier's posterior estimates. However, for *Naive Bayes Classifiers* these frequency estimates differ from the posterior estimates, due to the conditional independence assumed when computing the latter. Therefore, the classifier's estimates should be used directly for $\hat{p}$. For *k-Nearest Neighbour Classifiers*, these posterior estimates are obtained by the number of positives among an instance's k nearest neighbours. In analogy, for *Tree-Based Classifiers* such as Hoeffding Trees [7], the probabilistic estimates are obtained from the summary statistics in an instance's leaf, i.e. by simply dividing the number of positives by the total number of labels processed in that leaf.

In the classification algorithms discussed above, a label influences solely a particular region in the feature space. However, for some classifiers this does not hold. For example, in *Logistic Regression Classifiers* an instance might alter the decision on instances that are far away. Thus, even though Logistic Regression Classifiers provide probabilistic estimates that might be used for $\hat{p}$, they might be not suited for probabilistic active learning.

3. EXPERIMENTAL COMPARISON

Motivated by the relationship between the active learning strategies described in Section 2, the following three hypotheses guide the experimental evaluation:

1. Probabilistic active learning outperforms random, semi-random and uncertainty sampling.

2. The performance of probabilistic active learning drops if the label statistics are calculated independently of the classifier being used.

3. Semi-random sampling does not outperform random and uncertainty sampling at the same time.

The first hypothesis is motivated by the capability of probabilistic active learning to balance exploration and exploitation by computing the expected improvement in classification performance in an instance's neighbourhood, rather than using a heuristic approach. However, this relies on good estimates of the labelled information in an instance's neighbourhood, which are provided by the label statistics. These estimates depend on the classifier, thus computing them independently from the classifier is expected to deteriorate the performance, motivating the second hypothesis. The third hypothesis is motivated by the speculation in [27] that mixed strategies might be inferior in a pool-based setting with static concepts (as in our setting). According to this hypothesis, the performance of semi-random sampling should be between that of random and uncertainty sampling.

For testing these hypotheses, we follow the standard active learning assumptions, discussed and motivated in [22]:

1. All labels cost the same.

2. The labels that are bought are always correct.

3. The classifier learns incrementally on the actively selected labels, without any other change.

3.1 Experimental Setup

Active learning works on the trade-off between minimising the number of labels and maximising classification performance. For a single experiment, this trade-off is commonly visualised using learning curves, which depict the classifier's performance at different amounts of labelled instances. However, for a multitude of combinations of active learning approaches and datasets (as in this comparative study), a multitude of curves need to be compared. For matters of space and readability, different approaches for aggregating this information were used in literature. One proposed solution is to compare the area under the learning curve [11] but this method loses information about dominance at the different stages and might be misleading when learning curves intersect. Therefore, we use the approach suggested in [13] of pairwise comparisons at specific points in the learning process, in order to see which strategy dominates or is dominated by another strategy at which point in the learning process. Furthermore, in order to improve reliability of the results, we use n-fold-cross-validation to divide the datasets into different partitions of test and training sets. The experimental setup used for the comparison of active learning strategies is summarised by the following workflow:

1. Employ the selected strategies with the selected classifiers and datasets.

2. Compare the accuracy of two competing strategies after a specific number of instances were labelled and create two performance vectors for that point of comparison by collecting the achieved performance from all the folds of the 10-fold-cross-validation and do that for 10 random seeds. This gives us two vectors of the length 100 as there are 10 folds for each of the 10 seeds equalling a 100-fold-cross-validation.

3. Test if the performance vector of one strategy is significantly better or worse using a two-sided Wilcoxon test with a significance level of 0.05.

4. Repeat steps 2 and 3 for all classifiers on the individual datasets and also over all datasets at the same time which gives us a summary of how the strategies perform for a specific classifier over all datasets. The chosen comparison points are the performances obtained after labelling 20 and 40 instances. Accuracy is selected as performance measure.

5. Check if the results of step 4 are in line with the hypotheses or contradict them.

Table 1: Specifications of the data sets that were used for the experiments

Data Set	Instances	Attributes	Pr(+)
Seeds	210	7	33%
Vertebral	310	6	32%
Haberman	306	3	73%
Checkerboard1	308	2	44%
Checkerboard2	392	2	49%

3.1.1 Datasets

For the experiments the following real-world datasets from the UCI machine learning repository [1] are used: haberman, seeds, vertebral. Additionally, two synthetic datasets are included, namely checkerboard1 and checkerboard2 [4, 13]. The datasets are preprocessed such that there are no missing or invalid values and normalised such that all attribute values are between zero and one. The specifications of the data sets can be seen in Table 1. All the datasets are randomised and divided into ten folds, which are then used in the cross-validation of all active learning strategies. Since the datasets are small and the learning process converges quickly, the budget is set to 40 instances.

3.1.2 Algorithms

The compared active learning approaches are random sampling (uniform selection probability), semi-random and uncertainty sampling (both using confidence as uncertainty measure), and probabilistic active learning (using accuracy with $\tau = 0.5$ as performance measure) which were introduced in Sections 2.1 till 2.4.

All active learning strategies are evaluated on the same set of (incremental) classifiers. Those classifiers, implemented in MOA [3] and WEKA [12], are Hoeffding trees, Naive Bayes, logistic regression, k-nearest-neighbour and a Parzen-Window classifier which was implemented by the authors and is described in [4]. All algorithms were run on a desktop computer (Intel i5-760 with 2.8GHz and 8GB RAM).

The label statistics are once calculated by using the probabilistic classifier's posterior estimate for the values of the share of positives ($\hat{p}$) in a neighbourhood. Furthermore, to evaluate the effect of calculating the label statistics independently of the classifier, estimates based on kernel frequency estimates (as in [13]) over the labels are used.

3.2 Results

Based on the three hypotheses stated above, we now summarise our findings in the next subsections. Tables 3 and 2 provide the complete results of the experimental evaluation.

Table 3 shows the performance comparison after 20 and 40 labels over all datasets for different pairs of active learning strategies. The numbers are the percentages of wins of the strategy in the row versus the strategies in the columns, excluding ties. Thus, symmetric values sum up to one. Significantly better results of a two-sided Wilcoxon test with a significance level of 0.05 are denoted with a '*', significantly worse ones with a '-'. The active learning strategies are denoted with Pal (probabilistic active learning), Conf (confidence-based uncertainty sampling), Ran (random sampling), and Semi (semi-random sampling). For the columns on the left, the posterior estimates $\hat{p}$ come from the probabilistic classifier, while for the columns on the right they are calculated independently of the classifier by using kernel frequency estimates. In both cases, the number of labels n is calculated by kernel frequency estimates.

Table 2 summarises for different classifiers the effect on Pal's performance of using independently calculated posterior estimates against estimates takes from the probabilistic classifier. That is, the values correspond to the number of wins (excluding ties) of Pal with independently calculated posterior estimates (by using kernel frequency estimates) against Pal with estimates taken directly from the probabilistic classifier. A '*' shows that the performance is significantly better and a '-' shows that it is significantly worse using a two-sided Wilcoxon test with a significance level of 0.05. One can see that in the majority of cases calculating both parameters independently leads to a significantly worse classifier performance.

3.2.1 Probabilistic Active Learning Is Superior

In order to assess this statement, Table 3 provides the results for different classifiers. For a Parzen Window classifier (top-most cells), probabilistic active learning outperforms the other strategies significantly over all datasets, both after 20 and 40 acquired labels. This classifier's posterior estimates are kernel frequency estimates, thus there is no difference between its left and right subtables.

For Hoeffding Trees, this does only hold when posterior estimates by the classifier are used (64.26%, 64.92%, 62.7% at 20 labels, and 63.19%, 69%, 66.55% at 40 labels against confidence-based uncertainty sampling, random sampling, and semi-random sampling, respectively). When using independently calculated posterior estimates for probabilistic active learning, its performance is neither significantly better nor significantly worse than that of other approaches.

For Naive Bayes with posterior estimates by the classifier, Pal is again always significantly better. For Naive Bayes with kernel frequency estimates for the posterior, Pal is significantly better than random while not significantly worse than any other strategy.

For k-Nearest Neighbour and Logistic Regression, probabilistic active learning is not better: with k-NN it is significantly worse than uncertainty sampling or semi-random-sampling, but not significantly worse than random sampling. With logistic regression, results are inconclusive, but probabilistic active learning performs in some constellations significantly worse than uncertainty or random sampling. The reason for the weak performance of probabilistic active learning in combination with Logistic Regression is that here the smoothness assumption is violated, as an instance might influence the decision boundary at locations that are far away from its coordinates. The problem with k-Nearest Neighbour is a different one: here, the number of labels that are considered by the classifier is constantly set to three. Thus, the value n used in the label statistics is misleading the active learner. Overall, hypothesis one is confirmed for Parzen Window, Hoeffding Tree, and Naive Bayes, but not for k-Nearest Neighbour and Logistic Regression Classifiers.

3.2.2 Independently Calculated Label Statistics Reduce the Performance

The results discussed above already indicate an important relationship between the label statistics and the performance of the probabilistic active learning approach. To assess this relationship further, and to test the second hypothesis that classifier-independent calculation of these label statistics reduces the performance, Table 2 shows the results of a pairwise comparison between probabilistic active learning with and without independently computed posterior estimates. It depicts the percentage of cases where the performance with independent estimates was greater than the performance with estimates coming from the classifier. For example, the Naive Bayes classifier with independent estimates outperformed its counterpart with dependent estimates in 47.62% of the cases after 10 labels were bought and only in 25,76% of the cases after 40 labels were bought which is significantly worse being indicated by the '-' sign. A '*' ! would indicate that it performed better in most of the cases and that the result can be deemed significant.

Interestingly, the results depend on the learning stage: after processing the first ten labels (comparison point $CP = 10$), there is not yet a difference in performance between the two ways of calculating the label statistic's $\hat{p}$ (except for 3-Nearest Neighbour). In the later learning stages ($CP = 20, 30, 40$), this changes, and using independently estimated values for $\hat{p}$ significantly reduces performance for Hoeffding-Trees, Naive Bayes, and Logistic Regression. For 3-Nearest Neighbour, the results are different, but on this particular type of classifier the probabilistic active learning approach is not recommendable anyway.

One should note that this evaluation was limited to the effect of independent posterior estimates for $\hat{p}$, while always independently calculated estimates for the number of labels n were used. The situation of using for both values (for n and $\hat{p}$) kernel frequency estimates corresponds to using two classifiers, namely a Parzen-Window classifier for instance selection, and the chosen classifier for prediction. This is the typical scenario of label reusability as introduced by [24]. Summarising, the second hypothesis is confirmed for Hoeffding Trees, Naive Bayes, and Logistic Regression Classifiers.

Table 2: Effect of Independent Label Statistics Calculation

Labels	H-Tree	Naive B.	Log. Reg.	3-NN
CP=10	54.33%	47.62%	49.82%	34.92%-
CP=20	39.3%-	38.78%-	37.67%-	54.32%
CP=30	32.17%-	28.35%-	30.93%-	48.67%
CP=40	30.4%-	25.76%-	37.04%-	65.82%*

3.2.3 Semi-Random Sampling is not better than both Random and Uncertainty Sampling

The results in Table 3 confirm hypothesis three that that semi-random-sampling is with none of the classifiers consistently better (or worse) than both, random sampling and uncertainty sampling. That is, it is never at the same time dominating (or dominated by) both strategies. This supports the suggestions by [27] that a mixed strategy is inferior in a static setting because either uncertainty sampling will perform well or random will perform well and semi-random will end up in the middle of the two. However, this does not mean that a semi-random strategy is inferior to random or uncertainty sampling in every setting. For some configurations, semi-random sampling is slightly better than both, but in those cases the difference is never significant. Thus, in a real-world application where hold-out performance tests are difficult, semi-random sampling might help to avoid the worst-case performance. Nevertheless, for most classifier types probabilistic active learning seems to be the better choice, as it outperforms in general all three other methods when the label statistics are provided by the used classifier.

4. CONCLUSION

In this paper, the performance of popular active learning strategies in combination with different classification algorithms has been studied. These combinations were experimentally evaluated using 100-fold cross validation over several different real-world and synthetic datasets. The results confirm the finding of previous studies that neither pure exploration nor pure exploitation strategies perform consistently well, making the handling of the trade-off between *exploration* and *exploitation* a key challenge. In addition, the results show that the recently proposed probabilistic active learning approach significantly outperforms uncertainty-sampling-based strategies when used with Bayes, Naive Bayes or Decision-Tree Classifiers, but works not well on k-Nearest Neighbour or Logistic Regression Classifiers. Furthermore, it is shown that using a probabilistic classifier's estimates for the label statistics is in most cases better than using estimates that were calculated independently of the classifier. Finally, the results confirm the recently stated conjecture [27] that a hybrid between random and uncertainty sampling does not outperform both strategies at the same time in a pool-based setting.

While several combinations of active learning and classification approaches have been evaluated in this paper, this comparative study is by no means complete. Future work will focus on evaluating further combinations, as well as investigating further ways of computing better label statistics for some classification algorithms. Furthermore, comparisons for other active learning settings like user-based visually-supported active learning [21] would be insightful.

5. ACKNOWLEDGMENTS

We would like to thank Daniel Kottke and Myra Spiliopoulou.

6. REFERENCES

[1] A. Asuncion and D. J. Newman. UCI machine learning repository, 2015.

[2] J. Attenberg, P. Melville, F. Provost, and M. Saar-Tsechansky. *Selective Data Acquisition for Machine Learning*, chapter 5. CRC Press, Inc., 2011.

[3] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. Moa: Massive online analysis. *The Journal of Machine Learning Research*, 11:1601–1604, 2010.

[4] O. Chapelle. Active learning for parzen window classifier. In *Proc. of the 10th Int. Workshop on AI and Statistics*, pages 49–56, 2005.

[5] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-supervised Learning.* MIT Press, 2006.

[6] D. Cohn. Active learning. In C. Sammut and G. I. Webb, editors, *Encyclopedia of Machine Learning*, pages 10–14. Springer, 2010.

[7] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proc. of the 6th ACM SIGKDD int. conf. on Knowledge discovery and data mining (KDD00)*, pages 71–80. ACM, 2000.

[8] Y. Fu, X. Zhu, and B. Li. A survey on instance selection for active learning. *Knowledge and Information Systems*, 35(2):249–283, 2012.

[9] J. Gantz and D. Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east, December 2012.

[10] Y. Guo and D. Schuurmans. Discriminative batch mode active learning. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems (NIPS2007)*, pages 593–600, 2007.

[11] I. Guyon, G. Cawley, G. Dror, V. Lemaire, and A. Statnikov, editors. *Active Learning Challenge*, volume 6 of *Challenges in Machine Learning.* Microtome Publishing, 2011.

[12] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[13] G. Krempl, D. Kottke, and V. Lemaire. Optimised probabilistic active learning (OPAL) for fast, non-myopic, cost-sensitive active classification. *Machine Learning*, 2015.

[14] G. Krempl, D. Kottke, and M. Spiliopoulou. Probabilistic active learning: Towards combining versatility, optimality and efficiency. In S. Dzeroski, P. Panov, D. Kocev, and L. Todorovski, editors, *Proc. of the 17th Int. Conf. on Discovery Science (DS)*, volume 8777 of *LNCS*, pages 168–179. Springer, 2014.

[15] G. Krempl, I. Zliobaitė, D. Brzeziński, E. Hüllermeier, M. Last, V. Lemaire, T. Noack, A. Shaker, S. Sievi, M. Spiliopoulou, and J. Stefanowski. Open challenges for data stream mining research. *SIGKDD Explorations*, 16(1):1–10, 2014.

[16] L. Lan, H. Shi, Z. Wang, and S. Vucetic. Active learning based on parzen window. *Journal of Machine Learning Research*, 16:99–112, 2011.

[17] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proc. of the 17th annual int. ACM SIGIR conf. on Research and development in information retrieval*, SIGIR '94, pages 3–12, 1994.

[18] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in Fortran 77: The Art of Scientific Computing.* Cambridge University Press, 2 edition, 1992.

[19] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proc. of the 18th Int. Conf. on Machine Learning, ICML 2001*, pages 441–448, 2001.

[20] A. I. Schein and L. H. Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007.

[21] C. Seifert and M. Granitzer. User-based active learning. In *Proc. of 10th Int. Conf. on Data Mining Workshops (ICDMW2010)*, pages 418–425, 2010.

[22] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison, Madison, Wisconsin, USA, 2009.

[23] B. Settles. *Active Learning.* Number 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.

[24] K. Tomanek and K. Morik. Inspecting sample reusability for active learning. In I. Guyon, G. C. Cawley, G. Dror, V. Lemaire, and A. R. Statnikov, editors, *AISTATS workshop on Active Learning and*

Experimental Design, volume 16, pages 169–181. JMLR.org, 2011.

[25] J. Zhu, H. Wang, B. K. Tsou, and M. Y. Ma. Active learning with sampling by uncertainty and density for data annotations. *IEEE Trans. on Audio, Speech & Language Processing*, 18(6):1323–1331, 2010.

[26] J. Zhu, H. Wang, T. Yao, and B. K. Tsou. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In D. Scott and H. Uszkoreit, editors, *22nd Int. Conf. on Computational Linguistics (COLING 2008)*, pages 1137–1144, 2008.

[27] I. Zliobaitė, A. Bifet, B. Pfahringer, and G. Holmes. Active learning with drifting streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, PP(99), 2013.

Table 3: Part A: Pairwise performance comparison after 20 and 40 labels over all datasets. It shows how often the classifier using the strategy from the row outperformed the same classifier using the strategies in the columns. For example, a Parzen-Window classifier using probabilistic active learning outperformed confidence-based uncertainty sampling significantly in 69.48%, random sampling in 73,22% and semi-random sampling in 71.98% of the cases (*continues on the next page*).

Parzen-Window Classifier

Posterior from Classifier, 20 labels					**Posterior from KFE, 20 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	69.48%*	73.22%*	71.98%*	pal	0%	69.48%*	73.22%*	71.98%*
conf	30.52%-	0%	47.62%-	43.75%-	conf	30.52%-	0%	47.62%-	43.75%-
ran	26.78%-	52.38%*	0%	47.82%	ran	26.78%-	52.38%*	0%	47.82%
semi	28.02%-	56.25%*	52.18%	0%	semi	28.02%-	56.25%*	52.18%	0%
Posterior from Classifier, 40 labels					**Posterior from KFE, 40 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	69.09%*	68.19%*	66.32%*	pal	0%	69.09%*	68.19%*	66.32%*
conf	30.91%-	0%	38.3%-	37.13%-	conf	30.91%-	0%	38.3%-	37.13%-
ran	31.81%-	61.7%*	0%	51.93%	ran	31.81%-	61.7%*	0%	51.93%
semi	33.68%-	62.87%*	48.07%	0%	semi	33.68%-	62.87%*	48.07%	0%

Hoeffding-Tree Classifier

Posterior from Classifier, 20 labels					**Posterior from KFE, 20 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	64.26%*	64.92%*	62.7%*	pal	0%	51.66%	53.82%	49.1%
conf	35.74%-	0%	51.46%	48.92%	conf	48.34%	0%	51.68%	45.17%
ran	35.08%-	48.54%	0%	49.86%	ran	46.18%	48.32%	0%	44.48%
semi	37.3%-	51.08%	50.14%	0%	semi	50.9%	54.83%	55.52%	0%
Posterior from Classifier, 40 labels					**Posterior from KFE, 40 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	63.19%*	69%*	66.55%*	pal	0%	53.06%	56.71%	50.16%
conf	36.81%-	0%	52.88%	51.24%	conf	46.94%	0%	52.12%	48.48%
ran	31%-	47.12%	0%	45.48%	ran	43.29%	47.88%	0%	45.42%-
semi	33.45%-	48.76%	54.52%	0%	semi	49.84%	51.52%	54.58%*	0%

Table 3: Part B: Pairwise performance comparison for Naive Bayes, K-Nearest Neighbour, and Logistic Regression Classifiers (*continuation from the previous page*).

Naive Bayes Classifier

Posterior from Classifier, 20 labels					**Posterior from KFE, 20 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	58.97%*	64.66%*	60.7%*	pal	0%	51.25%	56.1%*	55.4%*
conf	41.03%-	0%	54.57%*	51.35%	conf	48.75%	0%	52.32%	55.41%*
ran	35.34%-	45.43%-	0%	47.26%	ran	43.9%-	47.68%	0%	51.26%
semi	39.3%-	48.65%	52.74%	0%	semi	44.6%-	44.59%-	48.74%	0%
Posterior from Classifier, 40 labels					**Posterior from KFE, 40 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	63.17%*	74.5%*	67.24%*	pal	0%	54.55%	56.9%*	47.93%
conf	36.83%-	0%	60.92%*	53.65%	conf	45.45%	0%	56.51%	48.99%
ran	25.5%-	39.08%-	0%	41.18%-	ran	43.1%-	43.49%	0%	42.94%-
semi	32.76%-	46.35%	58.82%*	0%	semi	52.07%	51.01%	57.06%*	0%

K-Nearest Neighbour (K=3) Classifier

Posterior from Classifier, 20 labels					**Posterior from KFE, 20 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	24.18%-	59.4%	30.34%-	pal	0%	27.53%-	50%	25%-
conf	75.82%*	0%	76.02%*	65.32%*	conf	72.47%*	0%	69.52%*	59.46%
ran	40.6%	23.98%-	0%	32.74%-	ran	50%	30.48%-	0%	30.81%-
semi	69.66%*	34.68%-	67.26%*	0%	semi	75%*	40.54%	69.19%*	0%
Posterior from Classifier, 40 labels					**Posterior from KFE, 40 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	25%-	68.59%*	43.27%-	pal	0%	18.64%-	46.25%	27.55%-
conf	75%*	0%	80.57%*	80.75%*	conf	81.36%*	0%	78.24%*	72.89%*
ran	31.41%-	19.43%-	0%	35.05%-	ran	53.75%	21.76%-	0%	30.5%-
semi	56.73%*	19.25%-	64.95%*	0%	semi	72.45%*	27.11%-	69.5%*	0%

Logistic Regression Classifier

Posterior from Classifier, 20 labels					**Posterior from KFE, 20 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	40.57%	37.5%-	45.31%	pal	0%	51.46%	46.05%	49.27%
conf	59.43%	0%	50.54%	55.75%*	conf	48.54%	0%	44.09%	47.43%
ran	62.5%*	49.46%	0%	56.52%*	ran	53.95%	55.91%	0%	51.63%
semi	54.69%	44.25%-	43.48%-	0%	semi	50.73%	52.57%	48.37%	0%
Posterior from Classifier, 40 labels					**Posterior from KFE, 40 labels**				
StrategyName	pal	conf	ran	semi	StrategyName	pal	conf	ran	semi
pal	0%	41.21%-	42.01%	43.53%	pal	0%	60.13%*	55.84%	58.13%*
conf	58.79%*	0%	52.66%	54.78%	conf	39.87%-	0%	44.03%	49.68%
ran	57.99%	47.34%	0%	48.78%	ran	44.16%	55.97%	0%	53.59%
semi	56.47%	45.22%	51.22%	0%	semi	41.88%-	50.32%	46.41%	0%