Georg Krempl Vincent Lemaire
Robi Polikar Bernhard Sick
Daniel Kottke Adrian Calma (Eds.)

IAL@ECML PKDD 2017

Workshop and Tutorial on Interactive Adaptive Learning

The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2017)

Skopje, Macedonia, September 18, 2017

Proceedings

Preface

Science, technology, and commerce increasingly recognize the importance of machine learning approaches for data-intensive, evidence-based decision making.

This is accompanied by increasing numbers of machine learning applications and volumes of data. Nevertheless, the capacities of processing systems or human supervisors or domain experts remain limited in real-world applications. Furthermore, many applications require fast reaction to new situations, which means that first predictive models need to be available even if little data is yet available. Therefore approaches are needed that optimize the whole learning process, including the interaction with human supervisors, processing systems, and data of various kind and at different timings: techniques for estimating the impact of additional resources (e.g. data) on the learning progress; techniques for the active selection of the information processed or queried; techniques for reusing knowledge across time, domains, or tasks, by identifying similarities and adaptation to changes between them; techniques for making use of different types of information, such as labeled or unlabeled data, constraints or domain knowledge. Such techniques are studied for example in the fields of adaptive, active, semi-supervised, and transfer learning. However, this is mostly done in separate lines of research, while combinations thereof in interactive and adaptive machine learning systems that are capable of operating under various constraints, and thereby address the immanent real-world challenges of volume, velocity and variability of data and data mining systems, are rarely reported. Therefore, this combined tutorial and workshop aims to bring together researchers and practitioners from these different areas, and to stimulate research in interactive and adaptive machine learning systems as a whole.

This workshop aims at discussing techniques and approaches for optimizing the whole learning process, including the interaction with human supervisors, processing systems, and includes adaptive, active, semi-supervised, and transfer learning techniques, and combinations thereof in interactive and adaptive machine learning systems. Our objective is to bridge the communities researching and developing these techniques and systems in machine learning and data mining. Therefore we welcome contributions that present a novel problem setting, propose a novel approach, or report experience with the practical deployment of such a system and raise unsolved questions to the research community.

All in all, we accepted five regular papers (7 papers submitted) and 3 short papers (4 submitted) to be published in these workshop proceedings. The authors discuss approaches, identify challenges and gaps between active learning research and meaningful applications, as well as define new application-relevant research directions.

We thank the authors for their submissions and the program committee for their hard work.

September 2017

Georg Krempl, Vincent Lemaire, Robi Polikar
Bernhard Sick, Daniel Kottke, Adrian Calma

Organizing Committee

Georg Krempl, Otto von Guericke University
Vincent Lemaire, Orange Labs France
Robi Polikar, Rowan University
Bernhard Sick, University of Kassel
Daniel Kottke, University of Kassel
Adrian Calma, University of Kassel

Program Committee

Michael Beigl, KIT
Giacomo Boracchi, Politecnico di Milano
Bartosz Krawczyk, Virginia Commonwealth University
Mark Embrechts, Rensselaer Polytechnic Institute
Michael Granitzer, University of Passau
Barbara Hammer, University of Bielefeld
Henner Heck, University of Kassel
Vera Hofer, University of Graz
George Kachergis, Radboud University
Christian Müller-Schloer, University of Hannover
Christin Seifert, TU Dresden
Ammar Shaker, University of Paderborn
Jasmina Smailovic, Jožef Stefan Institute
Myra Spiliopoulou, Otto von Guericke University
Jurek Stefanowski, University of Poznan
Dirk Tasche, Swiss Financial Market Supervisory Authority FINMA
Martin Znidarsic, Jožef Stefan Institute

Table of Contents

Invited Talk : Ensemble Learning from Data Streams with Active and Semi-Supervised Approaches

Bartosz Krawczyk

Department of Computer Science
Virginia Commonwealth University, Richmond, VA
bkrawczyk@vcu.edu

Abstract. Developing efficient classifiers which are able to cope with big and streaming data, especially with the presence of the so-called concept drift is currently one of the primary directions among the machine learning community. This presentation will be devoted to the importance of ensemble learning methods for handling drifting and online data. It has been shown that a collective decision can increase classification accuracy due to mutually complementary competencies of each base learner. This premise is true if the set consists of diverse and mutually complementary classifiers. For non-stationary environments, diversity may also be viewed as a changing context which makes them an excellent tool for handling data shifts. The main focus of the lecture will be given to using these mentioned advantages of ensemble learning for data stream mining on a budget. As streaming data is characterized by both massive volume and velocity one cannot assume unlimited access to class labels. Instead methods that allow to reduce the number of label queries should be sought after. Recent trends in combining active and semi-supervised learning with ensemble solutions, such as online Query by Committee or Self-Labeling Committees, will be presented. Additionally, this talk will offer discussion on emerging challenges and future directions in this area.

Challenges of Reliable, Realistic and Comparable Active Learning Evaluation

Daniel Kottke[1], Adrian Calma[1], Denis Huseljic[1],
Georg Krempl[2], and Bernhard Sick[1]

[1)] University of Kassel
Wilhelmshöher Allee 73, 34112 Kassel, Germany
{daniel.kottke,adrian.calma,bsick}@uni-kassel.de

[2)] Otto-von-Guericke University Magdeburg
Universitätsplatz 2, 39106 Magdeburg, Germany
georg.krempl@ovgu.de

Abstract. Active learning has the potential to save costs by intelligent use of resources in form of some expert's knowledge. Nevertheless, these methods are still not established in real-world applications as they can not be evaluated properly in the specific scenario because evaluation data is missing. In this article, we provide a summary of different evaluation methodologies by discussing them in terms of being reproducible, comparable, and realistic. A pilot study which compares the results of different exhaustive evaluations suggests a lack in repetitions in many articles. Furthermore, we aim to start a discussion on a gold standard evaluation setup for active learning that ensures comparability without reimplementing algorithms.

Keywords: Evaluation, Active Learning, Classification, Semi-supervised Learning, Data Mining

1 Introduction

The field of machine *active learning* (AL) investigates how a learning algorithm can learn to solve problems (e.g., classification or regression problems) more effectively by exploiting interactions with humans (e.g., experts in a specific application field) or simulation systems which are abstractly modeled as an *oracle* [1] (Fig. 1). In many application domains, it is unproblematic to collect unlabeled data, but gathering labels may be complicated, time-consuming, or costly [18]. Furthermore, AL is based on the assumption that by allowing the *learner* to be curios (i.e., it is allowed to choose the data from which it learns), it may learn faster [39].

Pool-based AL [29] usually starts with an initially empty or very sparsely labeled set of samples, a large pool of unlabeled samples (candidates), and iteratively queries for new labels from instances of the candidate pool by "asking the right questions". For example, in every learning cycle the oracle is asked to provide labels for the most "informative" samples based on a *selection strategy*.

Thereby, it aims to improve the performance of the learning model as fast as possible. After the labels are added, the knowledge model is updated.

In this article, we focus on three critical aspects of AL evaluation which are underrepresented in current AL research:

- **Reliable Evaluation:** Reliable evaluation results require a robust and reproducible evaluation methodology. Hence, the methodology should be described in detail and should be robust to varying seeds or shuffled data.
- **Realistic Evaluation:** Evaluating an AL algorithm in a lab setting (the lack of labels is just simulated) is not realistic. Often, implications for the real world do not hold. Hence, AL methods are not very common in industrial applications. We will discuss the challenges of a real-world application.
- **Comparable Evaluation:** Current evaluation methodologies vary a lot regarding its evaluation type, performance measure, number of repetitions, etc. Ideally, presented results are directly comparable with others. Hence, this article aims to initiate a discussion for a standardized AL evaluation gold standard.

The article starts with a general overview of components taking part in an AL cycle (Sec. 2). Next, we discuss aspects of reliable evaluation (Sec. 3) and compare two methodologies in a pilot study (Sec. 4). In Sec. 5, we present unrealistic assumptions for real-world applications. Finally, we conclude the work and propose an outlook on how comparable evaluation could be made possible.

2 Active Learning in Classification Tasks

The learning cycle of AL (see Fig. 1) consists of three main components: In pool-based AL for classification tasks, we have a selection strategy, an oracle, and a classifier. The selection strategy selects the instances from the candidate pool to be labeled by the oracle such that the classifier can learn a well-suited model. This procedure repeats until a stopping criterion is reached. In AL evaluation, we normally investigate the performance of the selection strategy. Using an omniscient oracle and a pre-trained classifier, we can assure that performance

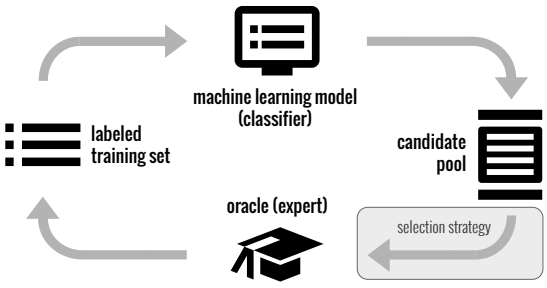


Fig. 1. Pool-based active learning cycle [39]

differences are solely induced by the selection of training instances from the candidate pool. Changing the classifier (or the parameters of the classifier) within different AL systems might lead to falsified results because of the high interdependence between the three components.

Comparing multiple classifiers in combination with AL, the selection strategy should be fixed. Comparing both, classifiers and selection strategies, one should run every combination. Unfortunately, some selection strategies solely work with specific classifiers or classifier types. Hence, it is not possible to compare these selection strategies with their individual classifiers as performance differences could be explained by the qualities of the classifiers and not the selection strategy. To face this problem, we could learn *multiple* classifiers on the selected samples. According to [42], this is subsumed under the term *label reusability*. The authors propose to use the specific classifier for the active selection (selector) and train additional classifiers for prediction (consumer). Although the authors of [42] show that the suitability of selector-consumer pairings cannot be estimated independently of the AL problem, we propose to run each selector also as a consumer for evaluation.

3 Aspects of Reliable Evaluation

Reliable evaluation is robust and reproducible. Robustness in evaluation means that changing seeds or the order of data points does not effect the results. In this section, we will point out different aspects and discuss what is done in literature.

3.1 Repetitions and Hold-Out Evaluation

In AL, we are facing classification tasks with very few training instances. When classifiers try to generalize from only a few training samples, their performance might be very sensitive to small changes. Also, the performance probably varies a lot depending on the concrete choice of instances to be labeled. Hence, lots of repetitions are needed to get a reliable trend of the performance. In Fig. 2, we clarify the nomenclature of different sets that might take part in AL.

In recent active learning articles, the number of repetitions varies between one single training-evaluation set [49] to 100 different partitionings [26]. Therefore, some authors use a k-fold cross validation [2, 5, 31] with solely one execution [31, 38] or multiple ones [2, 5]. Executing a k-fold cross validation multiple times

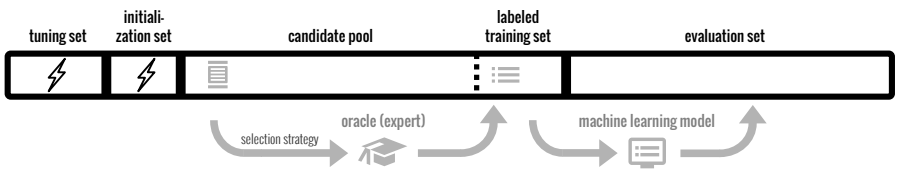


Fig. 2. Different sets used in literature for active learning.

requires different seeds among the repetitions. Others [8, 21, 30, 46] use a simple split with a fixed percentage (varying between 50% and 67%) for the candidate pool and the rest, respectively, for the evaluation set. To get rid of random effects, this is repeated multiple times.

In Sec. 4, we present a pre-study that shows the drawbacks of a single k-fold cross validation and shows the importance of multiple repetitions.

3.2 Performance Measures

Active Learning is a dynamic process which improves its model by successively adding labels to instances from the candidate pool. The aim of AL algorithms is to achieve a high performance which improves as fast as possible. Hence, we have two objectives [27, 39]:

1. achieve a high performance level (learn a good classifier) and
2. learn as fast as possible (save cost induced by annotations).

Applying Common Performance Measures to AL:

Depending on the learning problem, several performance measures [36] have been used. Usually, accuracy or error [2, 6] are used for problems with balanced misclassification cost and class priors. For unbalanced data, measures like cost, F1-Score, G-mean, Area under the Receiver Operating Characteristic-Curve (AUROC) [17, 20] (see [21, 22, 30, 48]) or H-measure [19] are more sophisticated. Usually, these performance measures are then plotted over time (resp. the number of acquired labels), which is then called learning curve (e.g., see Fig. 3).

As mentioned in the previous subsection, the results from multiple executions should be included in the evaluation by plotting standard deviations or ideally quartiles. An evaluation of means could also include the mean standard error or mean quartiles which can be determined using bootstrapping [15]. Note that quartiles are more exact as the distribution of performances given the number of acquired labels is unlikely normally distributed because these random variables are bounded (most of the time between 0 and 1).

The comparison of learning curves remains difficult as it is unclear how to combine the two objectives from above. The easiest option is to present the result for different points in time (e.g., early stage, mid stage, saturated stage) [26, 37]. Having fixed these time points, one can use comparison methods like in usual classification tasks. Note that most often, these time points and the total number of label acquisitions (when to stop learning) are chosen by the authors which could bias the results. We recommend not to stop learning before most of the AL algorithms have converged, and if possible, to also include the performance of a classifier learned on all instances as a baseline.

In reliable evaluation, statistical testing plays a essential role. Nevertheless, one should be reminded that statistical test only show if the results may also be explained by random artifacts [33], and do not show the real superiority of one's method. Nuzzo [33] claims that results should not only be reported by their statistical significance but also their effect size. Typically, statistical tests

(like the t-test or the Wilcoxon signed rank test [47]) assume to have i.i.d. random variables. Hence, the compared performance values should be drawn from the different training-evaluation combinations and not from different time points because these performance values are highly correlated and therefore *not* independent. One also could argue that even the performances across the repetitions are not independent because training and/or evaluation sets might overlap. Many use a t-test for comparing the tendencies of the mean between two algorithms [8, 21]. Due to the assumption of the mean being normally distributed, it might be better to use a parameter-free test like the Wilcoxon signed rank test [8, 22, 26, 41]. To test if an algorithm is significantly better across datasets, the Wilcoxon signed rank test might also be a good choice. An alternative to statistical testing is to present the number of won/lost trials using a simple pairwise comparison between the performances of two algorithms [26].

Active Learning Specific Performance Measures:

There also exist approaches to summarize the shape of the performance curve: The easiest approach sums up all the performance values for each time point. Often, this is called area under the learning curve [38] (also denoted as AUC[1]). This measure is proportional to the mean and hence dependent on the length of the AL process (i.e., the number of acquisitions which is often chosen manually).

More convenient is the deficiency score proposed by Yanik et al. [50]. This is determined by calculating the area between the maximal performance line and the actual learning curve which they call α for algorithm A and β for algorithm B. The deficiency of A with respect to B is then calculated using the following equation:

$$\text{deficiency}(A, B) = \frac{\alpha}{\alpha + \beta} \tag{1}$$

Another measure to calculate how fast the AL algorithm learns (2nd objective) is the Data Utilization Rate (DUR) by Reitmaier et al. [38]. They first compute the target accuracy defined as the mean (considering the performances between 80% and 100% of the total number of acquired labels) from the random strategy. The DUR is then the minimum number of samples needed by each strategy to reach this target accuracy divided by the number of samples needed by random.

3.3 Initialization of Active Learning

Some papers propose to initialize their AL cycle with some labels to be compatible to state-of-the-art implementations or as an essential part of their algorithm. The number of initialization labels varies between no label at all and 10% [30]. This choice is highly dependent on the dataset and the proposed algorithm. Unfortunately, it is often not described, how the specific values have been determined (or tuned), although this is essential for the method to succeed or fail.

[1] We do not recommend the abbrev. AUC because it can be mixed up with AUROC

The number of initial labels is relatively small when initialization is done due to compatibility issues [7, 13, 25, 37]. In some SVM implementations, the classifiers need one instance per class to predict labels. Hence, some authors added a fixed number of instances per class [43, 49, 50, 37] although this is not possible in real applications as the class labels are unknown in advance. This is even more relevant in datasets with unequal class priors as finding an instance of the minority class is especially difficult [16].

In [30, 48], the initialization step is used to have a representative sample for the dataset to find a broad decision boundary. Later, an uncertainty based method is used to refine the boundary and improve the performance. In this case, the number of samples used for initialization is critical for the active learning process. Especially, when the number of initial samples is varied across the datasets [30], one should mention how this number has been tuned.

For transparent evaluation of the selection strategy, we propose that algorithms with an initialization phase should be seen as a two step selection strategy. In the first step, labeling candidates are chosen according to an initialization strategy (e.g., random) which is stopped by a comprehensible stopping criterion. Then, the real active learning method can proceed. As this initialization phase is now part of the active learning algorithm it should be somehow evaluated (e.g., regarding robustness) and included in the learning curves [30, 37].

3.4 Parameter Tuning

Tuning parameters for classifiers is very difficult with only a few labels available. Unfortunately, these tuning procedures are often not described in great detail. Yanik et al. [50] used a grid search approach in an 5 fold cross validation after each label acquisition to tune the parameters of the SVM. Similarly, Tuia et al. [43] tune their parameters for their SVM. Both do not describe, on which data this is executed. Using a hold out tuning set [13, 27] is not valid in AL unless these additional labels are comprehensibly selected and included in the evaluation (i.e., considering them in the number of acquired labels in the learning curve). As in passive classification tasks, it is strictly forbidden to tune the parameters using the evaluation instances.

One could also argue that parameters should be adapted during learning as the number of training instances is increased by AL which affects the capability of generalization. This means, we either use a pre-trained mediocre classifier because parameters are tuned for a specific labeling situation, or we re-calibrate the parameters during learning which means that classifiers become different across selection methods which also biases the results.

Another way is to use standard parameter with normalized features (e.g. z-normalized) [25].

3.5 Proposing an AL Evaluation Methodology

In order to achieve reliable results across selection strategies, we propose the following methodology for AL evaluation:

- Use exactly the same robust classifier for every AL method when comparing and try to sync the parameters of these classifiers.
- Capture the effect of different AL methods on multiple datasets using at least 50 repetitions.
- Start with an initially unlabeled set. If you need initial training instances, sample randomly and explain how to determine the number of samples.
- Use either a clear defined stopping criterion or enough label acquisitions (sample until convergence).
- Show learning curves (incl. quartiles) with reasonable performance measures.
- Present pairwise differences in terms of significance and effect size (Wilcoxon signed rank test).

4 Pilot Study: Influence of the Number of Repetitions

The major challenge of AL evaluation is to measure the effect of improvement although the variance of results might be high: Especially in the early learning stages ($1\% - 10\%$ of the data are labeled), the classification performance varies a lot. This is where the differences across AL methods are highest. Hence, experiments have to be repeated multiple times to yield reliable results as mentioned before. In this section, we provide an exemplary evaluation methodology using a 5-fold cross validation.

For these experiments, we solely used one dataset from the UCI machine learning repository, named Mammographic Mass [3]. We chose this dataset as it is a typical representative for an AL dataset regarding the number of instances and features. For classification, we decided to use a robust classifier based on Gaussian kernel density estimation, namely a Parzen Window Classifier (PWC). Here, we only have one parameter: the bandwidth. In a pre-processing step, all categorical data has been dichotomized and all features are linearly transformed into $[0, 1]$ space. Hence, we use a standard bandwidth for the Gaussian kernel of the PWC of 0.2 as this seems to be reasonable. The AL algorithms are: Optimized Probabilistic AL [26], uncertainty sampling (Uncer) [29], an optimized version of expected error reduction from Chapelle (EER) [11], and random (Rand).

In 5-fold cross validation, we split the dataset D into 5 separate subsets ($D = D_1 \cup \ldots \cup D_5, D_i \cap D_j = \emptyset, i \neq j$) to build disjoint candidates and evaluation sets $(\mathcal{T}_i, \mathcal{E}_i)$. In this subsection, we applied AL 5 times on four of the subsets and evaluated the trained classifier on the left out subset.

Performing solely one complete 5-fold cross validation, as shown in Fig. 3, the performances might vary a lot. Furthermore, the ranking of the final performance (after 60 labels have been acquired) changes completely. The left evaluation shows OPAL being the best, followed by Expected Error Reduction, Random, and Uncertainty Sampling. Using another seed (right plot), the ranking is different: First OPAL, then Random, Uncertainty Sampling, and Expected Error

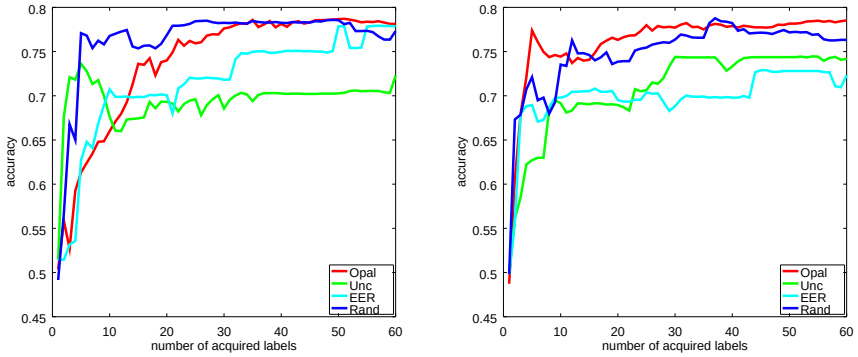


Fig. 3. Results of a 5-fold cross validation: two executions with different seeds of a complete 5-fold cross validation.

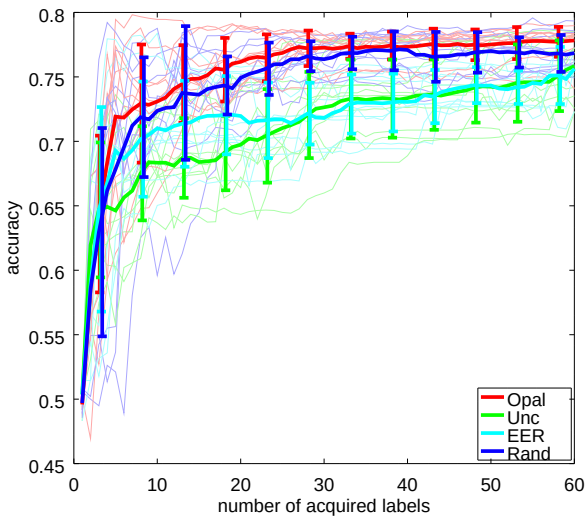


Fig. 4. Mean results of 10 times repeated 5-fold cross validations

Reduction. This clearly shows that a 5-fold cross validation evaluation for these AL methods on this dataset using a PWC is not sufficient. Similar experiments (not shown due to space restrictions) show that it is also true for other datasets and other classifiers. Repeating this 5-fold cross validation 10 times as shown in Fig. 4, provides much more stable results that are also comparable to the ones from the following experiment.

5 Challenges of realistic evaluation

Publications from companies such as Microsoft [24, 35], IBM [32], or Mitsubishi [23] show the growing interest in AL and its practical usefulness. AL has been successfully applied to solve problems such as on-road vehicle detection [40] or in recommender systems [28]. Unfortunately, these systems are highly specialized and often cannot be easily used for related problems.

In contrast to lab experiments, real active learning approaches only have one shot to learn. Hence, not the mean performance of multiple repetitions is of interest but the pairwise comparisons of the different methods. Because of high variances, it is still difficult to ensure a certain improvement of performance of one selection algorithm against others. This is the reason for many researchers arguing that random sampling is still a powerful baseline [10].

One of the main challenges to apply active learning in practice is to know when to stop querying for new label information. By now, in real-world applications, the AL process stops when a given "labeling budget" has been consumed. For example, in [40] the performance of the investigated AL approaches is done after a *fixed number* of queried samples. But, this may be a waste of resources, both in terms of time and money. Thus, the active learner should be able to asses its own performance. Here, different problems occur: a) collecting a separate evaluation dataset by randomly sampling instances is expensive, b) the collected data can not be used for performance estimation due to the sampling bias [12]. Some research work has been done to analyze when to stop the AL process besides estimating the performance directly [14, 34, 45]. It has been shown that it is possible to identify when a learning process might be saturated, but none provides information about the real classification performance.

In *dedicated collaborative interactive learning (D-CIL)* [9], different realistic applications for AL have been outlined. It addresses AL processes that are *interactive* – the information flows from humans to the active learner and vice versa, *collaborative* – multiple domain experts collaborate, and *dedicated* – a small number of benevolent domain experts interact with the active learner in order to support the selection process. Even though the oracles are impersonated by benevolent domain experts, they are still prone to error. Their labeling performance may depend on the labeler's experience, form of the day, or the complexity degree of the learning problem. In case of an *opportunistic* active learner [4], the oracles are not necessarily embodied by benevolent domain experts. Similar smart systems, simulation systems, or own sensors of the learning

system may assemble together or separately the oracle. Furthermore, there is high heterogeneity between these oracles, and their number is not fixed.

To summarize, AL research is mostly based on the following (limiting) assumptions [9]: a) the classification problem is well-defined (i.e., the number of classes and features are known in advance), b) labeled samples are available at the beginning of the learning process, c) uniform labeling cost (i.e., identical labeling costs for all samples), d) the oracle is omnipresent and omniscient, e) there exists a ground truth, based on which the performance of the active learner is evaluated. However, these assumptions often do not hold in real-world applications. Although, a large variety of specialized solutions is given which solve single problems, there is further work necessary to apply methods in a real-world setting. Here, a central aspect is the lack of comparability across different approaches which is a critical point for practitioners to apply AL in their specific domain.

6 Conclusion and Outlook

In this article, we summarized various challenges of AL evaluation with regard to being reliable, realistic, and comparable. Some of these appear naturally by the problem's definition, others are defined through the demands of real-world applications. We proposed an evaluation methodology to initialize a discussion on a gold standard for AL evaluation and provided preliminary results in a pilot study which shows the importance of many repetitions in AL which hopefully leads to comparable results without repeating whole experiments. Nevertheless, it is essential to report all details of evaluation to be able to reproduce the results of a paper. Those details have been discussed in this paper.

As future work, we plan to extend this literature overview and refine our proposed methodology. Additionally, we aim at providing a large comparison of different methodologies showing the effect of each component for different selection strategies. In this paper, we excluded the whole discussion of online algorithms and methods for evolving datastreams. Providing a valid evaluation framework for one-shot AL, is one of the goals of future research.

Our vision is to develop an evaluation system, enabling researchers and practitioners to collaborate. This system will provide a web-based user interface like OpenML [44] showing detailed information about different AL methods and their specific characteristics in relation to different tasks. In that way, we aim to standardize AL evaluation in order to simplify the steps towards practical solutions and fair comparison.

References

1. Aggarwal, C.C., Kong, X., Gu, Q., Han, J., Yu, P.S.: Active learning: A survey. In: Aggarwal, C.C. (ed.) Data Classification: Algorithms and Applications, pp. 571–606. CRC Press (2014)
2. Aldogan, D., Yaslan, Y.: A comparison study on ensemble strategies and feature sets for sentiment analysis. Lecture Notes in Electrical Engineering 363, 359–370 (2016)
3. Asuncion, A., Newman, D.J.: UCI machine learning repository (2015), `http://archive.ics.uci.edu/ml/`
4. Bahle, G., Calma, A., Leimeister, J.M., Lukowicz, P., Oeste-Reiß, S., Reitmaier, T., Schmidt, A., Sick, B., Stumme, G., Zweig, K.: Lifelong learning and collaboration of smart technical systems in open-ended environments – Opportunistic Collaborative Interactive Learning. In: International Conference on Autonomic Computing. IEEE, Würzburg, Germany (2017)
5. Bilgic, M., Getoor, L.: Active learning for networked data. Computer 411(29-30), 2712–2728 (2010)
6. Bouguelia, M.R., Belaïd, Y., Belaïd, A.: An adaptive streaming active learning strategy based on instance weighting. Pattern Recognition Letters 70, 38–44 (2016)
7. Brinker, K.: Incorporating diversity in active learning with support vector machines. In: Proceedings of the 20th International Conference on Machine Learning (ICML). pp. 59–66 (2003)
8. Cai, W., Zhang, Y., Zhou, S., Wang, W., Ding, C., Gu, X.: Active learning for support vector machines with maximum model change. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases. vol. 8724 (2014)
9. Calma, A., Leimeister, J.M., Lukowicz, P., Oeste-Rei, S., Reitmaier, T., Schmidt, A., Sick, B., Stumme, G., Zweig, A.K.: From active learning to dedicated collaborative interactive learning. In: Varbanescu, A.L. (ed.) 29th International Conference on Architecture of Computing Systems, Workshop Proceedings. pp. 1–8. VDI Verlag, Nuremberg, Germany (2016)
10. Cawley, G.C.: Baseline methods for active learning. In: Active Learning and Experimental Design Workshop in Conjunction with AISTATS 2010. pp. 47–57 (2011)
11. Chapelle, O.: Active learning for parzen window classifier. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics. pp. 49–56 (2005)
12. Dasgupta, S., Hsu, D.: Hierarchical sampling for active learning. In: Proceedings of the 25th International Conference on Machine learning. pp. 208–215. ACM (2008)
13. Demir, B., Persello, C., Bruzzone, L.: Batch-mode active-learning methods for the interactive classification of remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 49(3), 1014–1031 (2011)
14. Dimitrakakis, C., Savu-Krohn, C.: Cost-Minimising Strategies for Data Labelling: Optimal Stopping and Active Learning, pp. 96–111. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
15. Efron, B.: Bootstrap methods: another look at the jackknife. The annals of Statistics pp. 1–26 (1979)
16. Ertekin, S., Huang, J., Bottou, L., Giles, L.: Learning on the border: Active learning in imbalanced data classification. In: Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management. pp. 127–136. CIKM '07, ACM, New York, NY, USA (2007)

17. Flach, P., Hernandez-Orallo, J., Ferri, C.: A coherent interpretation of AUC as a measure of aggregated classification performance. In: Getoor, L., Scheffer, T. (eds.) Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA. pp. 657–664. ACM, New York, NY, USA (2011)
18. Fu, Y., Zhu, X., Li, B.: A survey on instance selection for active learning. Knowledge and Information Systems 35(2), 249–283 (2013)
19. Hand, D.J.: Measuring classifier performance: a coherent alternative to the area under the roc curve. Machine Learning 77(1), 103–123 (2009)
20. Hu, B.G., Dong, W.M.: A study on cost behaviors of binary classification measures in class-imbalanced problems. arXiv preprint arXiv:1403.7100 (2014)
21. Huang, K.h., Lin, H.t.: A novel uncertainty sampling algorithm for cost-sensitive multiclass active learning. In: 2016 IEEE 16th International Conference on Data Mining (ICDM) (2016)
22. Huang, S.j., Jin, R., Zhou, Z.H.: Active learning by querying informative and representative examples. In: NIPS'10 Proceedings of the 23rd International Conference on Neural Information Processing Systems. pp. 892–900 (2010)
23. Joshi, A.J., Porikli, F., Papanikolopoulos, N.P.: Scalable active learning for multiclass image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(11), 2259–2273 (2012)
24. Kapoor, A., Horvitz, E., Basu, S.: Selective supervision: Guiding supervised learning with decision-theoretic active learning. In: Veloso, M.M. (ed.) Proceedings of the 20th International Joint Conference on Artifical Intelligence. pp. 877–882. Morgan Kaufmann Publishers Inc. (2007)
25. Kottke, D., Krempl, G., Lang, D., Teschner, J., Spiliopoulou, M.: Multi-class probabilistic active learning. In: ECAI. Frontiers in Artificial Intelligence and Applications, vol. 285, pp. 586–594. IOS Press (2016)
26. Krempl, G., Kottke, D., Lemaire, V.: Optimised probabilistic active learning (OPAL) for fast, non-myopic, cost-sensitive active classification. Machine Learning pp. 1–28 (2015)
27. Krempl, G., Kottke, D., Spiliopoulou, M.: Probabilistic active learning: Towards combining versatility, optimality and efficiency. In: Proceedings of the 17th International Conference on Discovery Science (DS), Bled. Lecture Notes in Computer Science, Springer (2014)
28. Lamche, B., Trottmann, U., Wörndl, W.: Active Learning Strategies for Exploratory Mobile Recommender Systems. In: Proceedings of the Fourth Workshop on Context-Awareness in Retrieval and Recommendation. pp. 10–17. Amsterdam, Niederlande (2014)
29. Lewis, D., Gale, W.: A sequential algorithm for training text classifiers. In: Conference on Research and Development in Information Retrieval. pp. 3–12. ACM/Springer, New York, NY (1994)
30. Li, X., Guo, Y.: Active learning with multi-label svm classification. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (2013)
31. Longstaff, B., Reddy, S., Estrin, D.: Improving activity classification for health applications on mobile devices using active and semi-supervised learning. Proceedings of the 4th International ICST Conference on Pervasive Computing Technologies for Healthcare (2010)
32. Melville, P., Sindhwani, V.: Active dual supervision: Reducing the cost of annotating examples and features. In: Workshop on Active Learning for Natural Language Processing. pp. 49–57. Boulder, CO (2009)
33. Nuzzo, R.: Statistical errors. Nature 506(7487), 150 (2014)

34. Olsson, F., Tomanek, K.: An intrinsic stopping criterion for committee-based active learning. In: Conference on Computational Natural Language Learning. pp. 138–146. Boulder, CO (2009)
35. Paquet, U., Gael, J.V., Stern, D., Kasneci, G., Herbrich, R., Graepel, T.: Vuvuzelas & active learning for online classification. In: Workshop on Computational Social Science and the Wisdom of Crowds. pp. 1–5. Whistler, BC (2010)
36. Parker, C.: An analysis of performance measures for binary classifiers. In: Proceedings of the 11th IEEE International Conference on Data Mining (ICDM). pp. 517–526. IEEE (2011)
37. Pasolli, E., Melgani, F.: Active learning methods for electrocardiographic signal classification. IEEE Transactions on Information Technology in Biomedicine 14(6), 1405–16 (2010)
38. Reitmaier, T., Sick, B.: Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4DS. In: Information Sciences - Informatics and Computer Science Intelligent Systems Applications. vol. 230, pp. 106–131 (2013)
39. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin, Department of Computer Science (2009)
40. Sivaraman, S., Trivedi, M.M.: Active learning for on-road vehicle detection: a comparative study. Machine Vision and Applications pp. 1–13 (2011)
41. Son, Y., Lee, J.: Active learning using transductive sparse bayesian regression. Information Sciences 374, 240–254 (2016)
42. Tomanek, K., Morik, K.: Inspecting sample reusability for active learning. In: Guyon, I., Cawley, G.C., Dror, G., Lemaire, V., Statnikov, A.R. (eds.) Workshop on Active Learning and Experimental Design. JMLR Proceedings, vol. 16, pp. 169–181 (2011)
43. Tuia, D., Volpi, M., Copa, L., Kanevski, M., Munoz-Mari, J.: A survey of active learning algorithms for supervised remote sensing image classification. IEEE Journal of Selected Topics in Signal Processing 5(3), 606–617 (2011)
44. Vanschoren, J., van Rijn, J.N., Bischl, B., Torgo, L.: Openml: Networked science in machine learning. SIGKDD Explorations 15(2), 49–60 (2013)
45. Vlachos, A.: A stopping criterion for active learning. Computer Speech & Language 22(3), 295–312 (2008)
46. Wang, J., Park, E.: Active learning for penalized logistic regression via sequential experimental design. Neurocomputing 222, 183–190 (2017)
47. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics bulletin 1(6), 80–83 (1945)
48. Yan, Y., Rosales, R., Fung, G., Dy, J.G.: Active learning from crowds. Proceedings of the 28th International Conference on Machine Learning pp. 1161–1168 (2011)
49. Yang, Y., Ma, Z., Nie, F., Chang, X., Hauptmann, A.G.: Multi-class active learning by uncertainty sampling with diversity maximization. International Journal of Computer Vision 113(2), 113–127 (2014)
50. Yanik, E., Sezgin, T.M.: Active learning for sketch recognition. Computers and Graphics (Pergamon) 52, 93–105 (2015)

Interactive Anonymization for Privacy aware Machine Learning

Bernd Malle[1 2], Peter Kieseberg[1 2], Andreas Holzinger[1]

[1] Holzinger Group HCI-KDD
Institute for Medical Informatics, Statistics & Documentation
Medical University Graz, Austria
b.malle@hci-kdd.org
[2] SBA Research gGmbH, Favoritenstrae 16, 1040 Wien
PKieseberg@sba-research.org

Abstract. Privacy aware Machine Learning is the discipline of applying Machine Learning techniques in such a way as to protect and retain personal identities during the process. This is most easily achieved by first anonymizing a dataset before releasing it for the purpose of data mining or knowledge extraction. Starting in June 2018, this will also remain the sole legally permitted way within the EU to release data without granting people involved the *right to be forgotten*, i.e. the right to have their data deleted on request. To governments, organizations and corporations, this represents a serious impediment to research operations, since any anonymization results in a certain degree of reduced data utility. In this paper we propose applying human background knowledge via interactive Machine Learning to the process of anonymization; this is done by eliciting human preferences for preserving some attribute values over others in the light of specific tasks. Our experiments show that human knowledge can yield measurably better classification results than a rigid automatic approach. However, the impact of interactive learning in the field of anonymization will largely depend on the experimental setup, such as an appropriate choice of application domain as well as suitable test subjects.

Keywords: Machine Learning, Privacy aware ML, interactive ML, Knowledge Bases, Anonymization, k-Anonymity, SaNGreeA, Information Loss, Weight Vectors

1 Introduction and Motivation

In many sectors of today's data-driven economies technical progress is dependent on data mining, knowledge extraction from diverse sources, as well as the analysis of personal information. Especially the latter constitutes a vital building-block for business intelligence and the provision of personalized services, which are practically demanded by modern society. Often, the insights necessary for enabling organizations to provide these goods require publication, linkage, and systematic analysis of personal data sets from heterogeneous sources, exposing

those data to the risk of leakage, with repercussions ranging from mild inconvenience (exposure of a social profile) to potentially catastrophic ramifications (leakage of health information to an employer).

Living up to those challenges, governments around the world are contemplating or already enacting new laws concerning the handling of personal data. For instance, under the new European General Data Protection Regulations (*GDPR*) taking effect on June 1st, 2018, customers are given a *right to be forgotten*, meaning that an organization is obligated to remove a customer's personal data upon request. An exception to this rule is only granted to organizations which anonymize data before analyzing them in any wholesale, automated fashion. This brings us to the field of Privacy aware machine learning (PaML), e.g. the application of ML algorithms only on previously anonymized data. Such anonymization can be provided by perturbing data (e.g. introduction noise into numerical values or *differential privacy* [4]) or *k-anonymity* [17] (clustering of data into equivalence groups), which has since become the industry standard.

The original requirement of *k-anonymity* has since been extended by the concepts of *l-diversity* [11] (where every cluster must contain at least l diverse sensitive values), *t-closeness* [9] (demanding that the local distribution over sensitive values must not diverge from its global distribution by more than a threshold of t) as well as *delta-presence* [15] (which incorporates the background knowledge of a potential attacker). Although all of those concepts are interesting in their own right, for the sake of comparing interactive ML algorithms to their fully automatic counterpart, we only took *k-anonymity* into consideration.

Based on our previous works on this topic [13] [12], in which we conducted a comparison study of binary classification performance on perturbed (selective deletion) vs. wholesale anonymized data, in this paper we introduce the notion of interactive Machine Learning for (k-)anonymization.

2 k-Anonymity

Given the original tabular concept of anonymization, we will usually encounter three different categories of attributes within a given dataset:

- **Personal identifiers** are data items which directly identify a person without having to cross-reference or further analyze them. Examples are email address or social security number (SSN). As personal identifiers are immediately dangerous, this category of data is usually removed.
- **Sensitive data,** also called 'payload', represents information that is crucial for further data mining or research purposes. Examples for this category would be disease classification, drug intake or personal income. This data shall be preserved in the anonymized dataset and can therefore not be deleted or generalized.
- **Quasi identifiers (QI's)**, are data which in themselves do not directly reveal the identity of a person, but might be used in aggregate to reconstruct it. For instance, [18] reported in 2002 that the identity of 87% of U.S. citizens

could be uncovered via just the 3 attributes *zip code*, *gender* and *date of birth*. Despite this danger, QI's may contain vital information to research applications (like ZIP code in a disease spread study); they are therefore generalized to an acceptable compromise between privacy (data loss) and information content (data utility).

Based on this categorization *k-anonymity* [16] was introduced as a formal concept of privacy, in which a record is released only if its quasi-identifiers are indistinguishable from at least $k-1$ other entities in the dataset. This can be imagined like a clustering of data into so-called *equivalence groups* of at least size k, with all internal QI's being generalized to the exact same level.

Generalization in this setting means an abstraction of attribute value: e.g. given two ZIP codes of '8010' and '8045', we could first generalize to '80**', then incorporate another data point showing ZIP '8500' by generalizing the cluster to '8***', and finally merging with any other ZIP code to the highest level of 'all', also denoted as '*'.

3 interactive Machine Learning

Interactive ML algorithms adjust their inner workings by continuously interacting with an outside *oracle*, drawing positive / negative reinforcement from this interaction [7]. Such systems are especially useful for highly-personalized predictions or decision support [8]; moreover many real-world problems exhibit (super)exponential algorithmic runtime; in such cases human brains dwarf machines at approximating solutions and learning from very small samples, thus enabling us to 'intuit' solutions efficiently [6].

By incorporating humans as oracles into this process, we can elicit background knowledge regarding specific use cases unknown to automatic algorithms [19]. This however is highly dependent on the users' experience in a certain field as well as data / classification complexity; domain experts can of course be expected to contribute more valuable decision points than laymen; likewise, a low-dimensional dataset and simple classification tasks will result in higher quality human responses than convoluted problem sets.

While the authors of [14] propose a system that interacts with a user in order to set a certain k-factor and subsequently provides a report on information loss and Kurtosis of QI distributions, the algorithm is not *interactive* by our definition in that it does not influence the inner workings of the algorithm during the learning phase. This is also true in case of the Cornell Anonymization Toolkit (Cat) [20], which conducts a complete anonymization run and only afterwards lets the user decide if they are satisfied with the results. In contrast, our approach alters algorithmic parameters upon every (batch of) human decisions, letting the algorithm adapt in real-time.

[10] describe an approach incorporating humans into the anonymization process by allowing them to set constraints on attribute generalization; moreover they construct generalization hierarchies involving domain-specific ontologies.

Although this technique marks a departure from wholesale automatic anonymization, it still lacks the dynamic human-computer interaction of our approach.

Apart from the field of privacy, interactive ML is present in a wide spectrum of applications, from bordering medical fields like protein interactions / clusterings [1] via on-demand group-creation in social networks [2] to even teaching algorithms suitable mappings from gestures to music-generating parameters [5].

4 Experiments

The following sections will describe our experiment in detail, encompassing the general iML setting, chosen data set, anonymization algorithm used as well as a description of the overall processing pipeline employed to obtain the final results as presented.

4.1 General setting

The basic idea of our experiment was to compare different weight vectors representing attribute (quasi-identifier) importance during anonymization: Let's say that a doctor needs ro release a dataset for the purpose of studying disease-spread; in this case 'ZIP code' information is probably (but not necessarily) of much greater importance then 'occupation' or 'race'. However, if a skin cancer study is to be performed, 'race' information might be of utmost importance, whereas 'ZIP code' might be negligible.

In our experiment, the task was to classify a people dataset on the target attributes *income*, *education level* and *marital status*. Therefore, we tested an *equal* weight vector setting against two others obtained from human experiments: 1) *bias* in which the user just specified which attributes they thought would be important for a specific classification by moving sliders, and 2) *iML* in which the user was tasked to decide a series of clustering possibilities by moving a data row to one of two partly anonymized clusters presented, thereby conveying which attributes were more important to preserve than others (Figure 1). Only the last method constitutes an interactive learning approach by introducing an oracle into the process.

4.2 Data

We chose the adults dataset from the UCI Machine Learning repository which was generated from US census data from 1994 and contains approximately 50k entries in it's original; this data-set is used by many anonymization researchers and therefore constitutes a quasi-standard. After initial preprocessing we chose the first 500 complete data rows as our iML experimental data to be presented to users. After obtaining bias / iml weights from the experiment, we chose the first 3k entries of the original data as the basis for producing 775 new, anonymized data sets. Although 3k rows might seem overly frugal on our part, we have asserted via random deletion of original data points that classifier performance remains stable for as little as 1.5k rows. Of the original attributes (data columns)

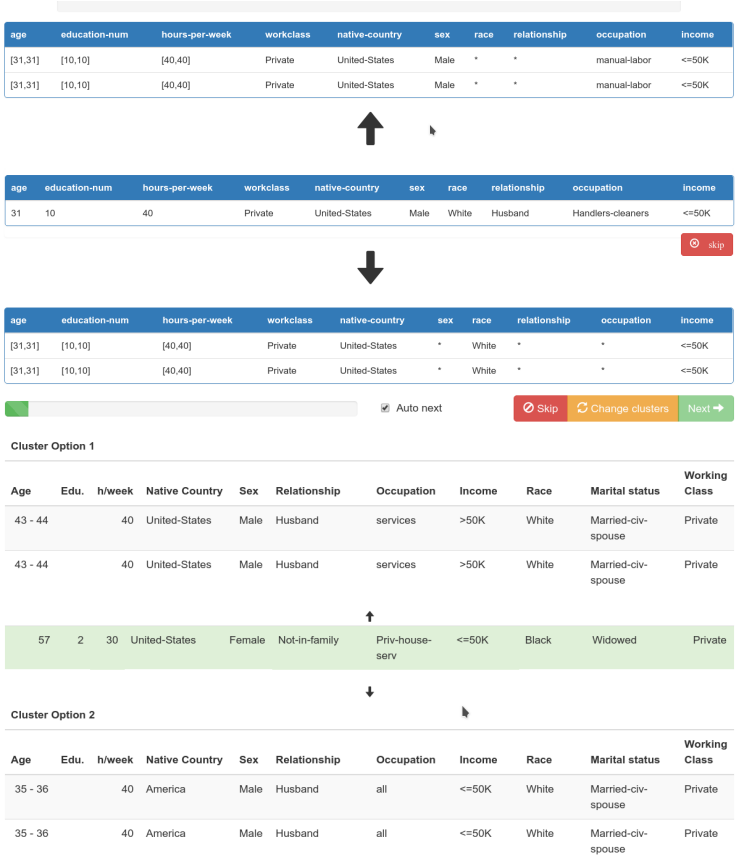


Fig. 1. Two different implementations of the iML interface design.

provided 4 were deleted: 'capital-gain' & 'capital-loss' (both were too skewed to be useful for humans), 'fnlwgt' (a mere weighting factor) as well as 'education' which is also represented by 'education_num'.

4.3 Anonymization Algorithm

In order to conduct our experiments, it was necessary to choose an algorithm which would enable us to easily hook into its internal logic - we therefore chose a greedy clustering algorithm called *SaNGreeA* (Social network greedy clustering) which was introduced by [3] and implemented it in JavaScript. This enabled us to execute it within a browser environment during our iML experiments as well as server-side for batch-execution of all derived datasets afterwards. As a greedy clustering algorithm SaNGreeA's runtime lies in $O(n^2)$ - which we were willing to accept in exchange for it's white-box internals.

Besides its capacity to anonymize graph structures (which we did not utilize during this work), it is a relatively simple algorithm considering *General information loss* - or GIL - during anonymization. This GIL can be interpreted by the sum of information loss occurring during generalization of continuous (range) as well as hierarchical attributes:

$$\mathrm{GIL}(cl) = |cl| \cdot (\sum_{j=1}^{s} \frac{size(gen(cl)[N_j])}{size(min_{x \epsilon N}(X[N_j]), max_{x \epsilon N}(X[N_j]))} + \sum_{j=1}^{t} \frac{height(\Lambda(gen(cl)[C_j]))}{height(H_{C_j})})$$

where:
- $|cl|$ denotes the cluster cl's cardinality;
- $size([i1, i2])$ is the size of the interval $[i1, i2]$, i.e., $(i2 - i1)$;
- $\Lambda(w), w \epsilon H_{C_j}$ is the sub-hierarchy of H_{C_j} rooted in w;
- $height(H_{C_j})$ denotes the height of the tree hierarchy H_{C_j};

The following formulas then give the total / normalized GIL, respectively:

$$\mathrm{GIL}(G, S) = \sum_{j=1}^{v} \mathrm{GIL}(cl_j) \quad \text{and} \quad \mathrm{NGIL}(G, S) = \frac{\mathrm{GIL}(G, S)}{n \cdot (s + t)}$$

The algorithm starts by picking a (random or pre-defined) data row as its first cluster, then iteratively picking best candidates for merging by minimizing GIL until the cluster reaches size k, at which point a new data point is chosen as the initiator for the next cluster; this process continues until all data points are merged into clusters, satisfying the k-anonymity criterion for the given dataset.

4.4 Processing pipeline for obtaining results

Once our iML experiments had yielded enough weight vectors, we had to generate a whole new set of anonymized datasets on which we subsequently applied 4 classifiers on each of the 3 target attributes (columns) described; therefore we designed the following processing pipeline:

1. Taking the first 5k rows of the original, preprocessed dataset as input and applying k-anonymization with a k-factor range of [5, 10, 20, 50, 100, 200] and 129 different weight vectors (equal, bias, iml) from our experiments on it, we produced 774 anonymized datasets (775 including the original).
2. We executed 4 classifiers on all of the datasets and compared their F1 score; the reason for selecting multiple algorithms was to explore if anonymization would yield different behaviors on different mathematical approaches for classification. The four algorithms used were *linear SVC* (as a representative of Support Vector Machines), *logistic regression* (gradient descent),

gradient boosting (ensemble, boosting) as well as *random forest* (ensemble, bagging). While reading the datasets pertaining to the classification target of *education*, the 14 different education levels present within the adult dataset were grouped into 4 categories 'pre-high-school', 'high school', '<=bachelors' and 'advanced studies'.

3. For each combination of classification target (*income*, *marital status*, *education*) and weight category (*equal*, *bias*, *iml*) we averaged the respective results. Results were plotted per target, as this allows better comparison between different classifiers. The leftmost point in all plots designates the original, un-anonymized dataset.

5 Results & Discussion

As per the results in our previous work on PaML [13] [12] we generally expected $1/x$ shaped curves for classifier performance as factors of k are increasing. These expectations held to only a small degree; moreover for targets *education* as well as *income* there was no clear winner amongst the weight categories, with some achieving better or worse depending on a specific factor of k.

We got the smoothest results for the *marital status* target, with human bias winning consistently over equal weights as well as human interaction (Figure 2). We interpret this as stemming from the fact that there is a significant correlation between the attributes 'marital-status' and 'relationship' in the dataset, which led users to consciously overvalue the latter when prompted for their bias. It is not completely clear why the iML results were not able to keep up in this case, but since this seems to be a general phenomenon throughout our results, we will discuss this in a later paragraph.

On classification target *education*, bias still mostly outperforms iML-obtained attribute weights, with equal weights slightly winning out at very high factors of k (Figure 3). Although we assume that apparently important clues towards education might be misleading (like income or working hours), this cannot explain the difference between bias- and iML-based results. It has to be noted however, that results on this target are distinctly inferior to those of the other scenarios which might diminish the gap's significance.

Only on target *income* did we observe a partly reversed order between human bias and iML - however at the cost of both being usually inferior to a simple setting with equal attribute weights (Figure 4). This is especially surprising because *income* was the only binary classification task in our experiments, which should have given humans a slight advantage over the algorithm. On the other hand, human bias seems most susceptible to falling prey to certain stereotypes in the area of money (w.r.t. gender, race, marital status...), which would explain the reversal of results.

As for the failure of iML to significantly outperform both the equal weight setting and especially human bias, we conjecture that our experimental setup has produced those effects: Since we wanted our users to conduct their experiment in real-time but needed a simple implementation of an anonymization algorithm to

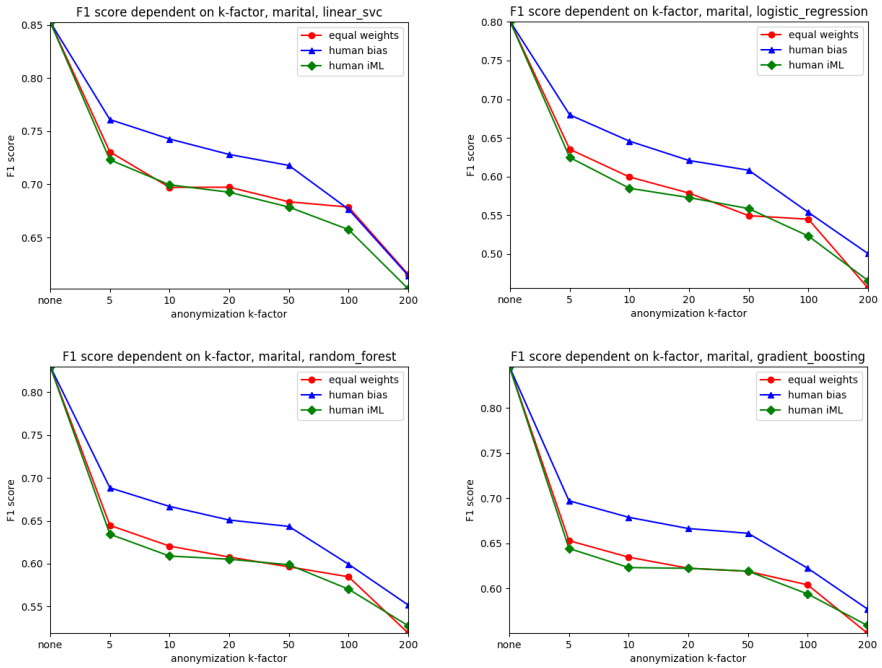


Fig. 2. Results on target *marital status* - human bias wins consistently over both equal weights and human interaction with the algorithm.

enable this interaction (which resulted in an $O(n^2)$ algorithmic runtime), we had to limit ourselves to just a tiny subset of data (500 rows, merely 1% of the original dataset). This choice apparently resulted in generalizations proceeding far too quickly, reaching suppression ('all') levels prematurely, thereby denying our users sensible clustering choices. On the other hand, the effect could also stem from users not really trying to contribute to the experiments in a meaningful way; this effect could only be mitigated by selecting more serious users or choosing some less serious (more social?) application domain.

Overall, we were also surprised that a seemingly absurd k-factor of 200 would still yield comparably good results (and in some cases even improve performance..).

6 Open problems & future challenges

As iML for anonymization is still a fledgling sub-area in the larger fields of privacy as well as Machine Learning, there are certainly innumerable possibilities for even basic progress & development. The following list is only a tiny subset of possible research venues we deem suitable for our own future work:

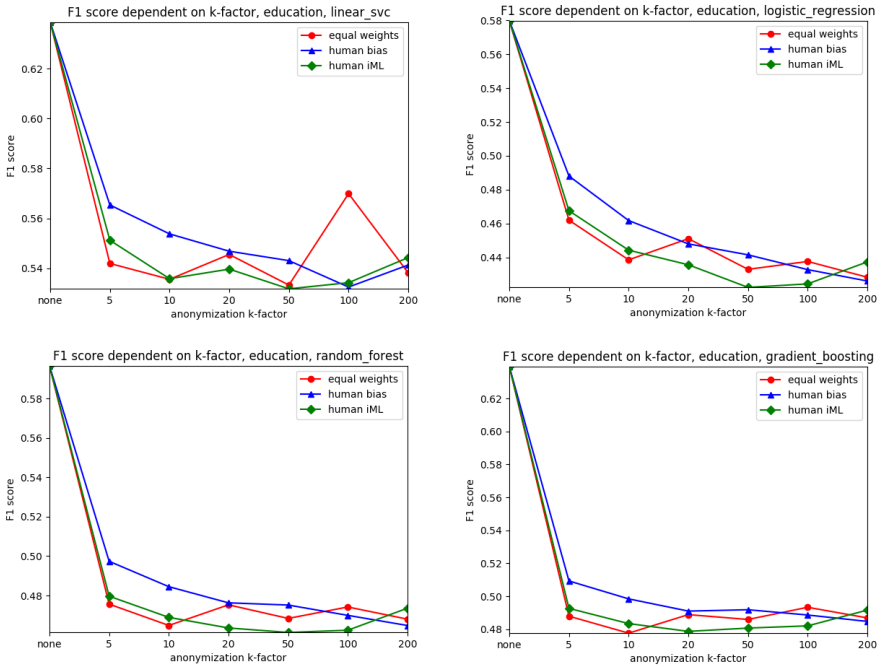


Fig. 3. Results on target *education* - we still see human bias performing slightly better than equal weights / iML in most cases of k, but not as consequently as above.

- **Explain the unexpected behavior** of linear SVC on the *income* target at high levels of k; probably by performing comparison studies on synthetically generated datasets.
- **Faster algorithm.** Repeat the experiments with a faster algorithmic implementation so that we can use thousands of data points even in real time within a Browser: this would lead to more relaxed generalizations, allowing the user to make better interactive choices, thus presumably improving results by quite some margin.
- **Expert domain, domain experts.** Choosing an expert domain like cancer studies in combination with proper experts like medical professionals, we would expect both human bias as well as iML results to significantly outperform a pre-defined weight vector.
- **Different setting.** On the other hand, a more 'gamified' setting such as recommendations within a social network could motivate amateur users to get more immersed into the experiment, yielding better results even for mundane application tasks.
- **Different data formats.** As Artificial Intelligence is slowly reaching maturity, it is now also applied to non- and semi-structured data like audio/video

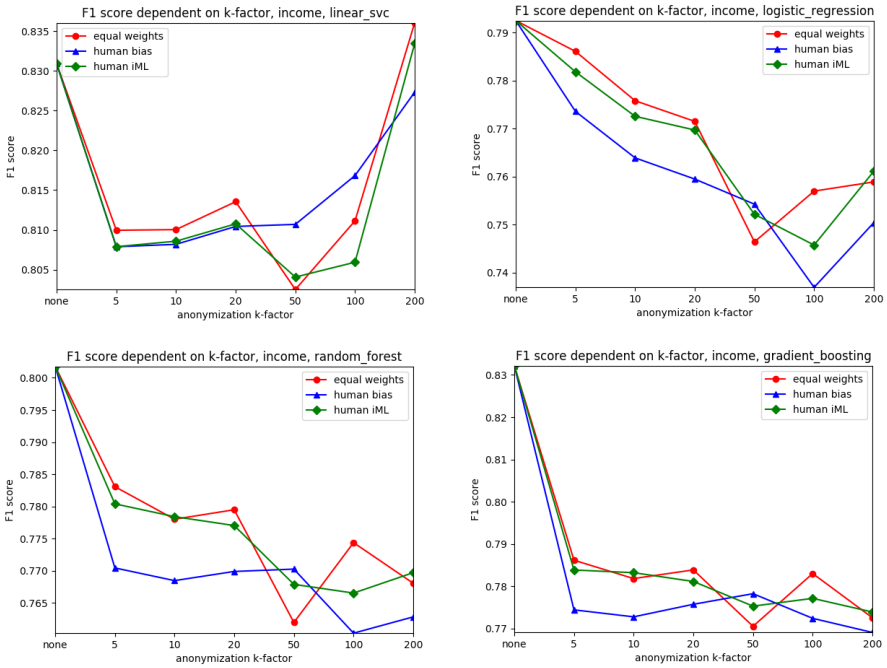


Fig. 4. Results on target *income* - only in this scenario do we see iML-based results generally outperforming bias (except linear SVC), nevertheless incapable of outperforming the rigidly equal setting.

or even *omics data. Since images are clearly relevant for medical research, and humans extremely efficient at processing them, studying interactive ML on visual data promises great scientific revenue.

7 Conclusion

Based on the emerging necessity of Privacy aware data processing, in this work we presented a fundamental approach of bringing human knowledge to bear on the task of anonymization via interactive Machine Learning. We devised an experiment involving clustering of data points with respect to human preference for attribute preservation and tested the resulting parameters on classification of anonymized people data into classes of *marital status*, *education* and *income*. Our preliminary results show that human bias can definitely contribute to even mundane application areas, whereas more complex or convoluted tasks may require trained professionals or better data preparation (dimensionality reduction etc.). We also described our insights regarding technical details for iML experiments and closed by outlining promising future research venues.

References

1. Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, 35(4):105–120, 2014.
2. Saleema Amershi, James Fogarty, and Daniel Weld. ReGroup: interactive machine learning for on-demand group creation in social networks. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, page 21, 2012.
3. Alina Campan and Traian Marius Truta. Data and structural k-anonymity in social networks. In *Privacy, Security, and Trust in KDD*, pages 33–54. Springer, 2009.
4. Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.
5. R. Fiebrink, D. Trueman, and P.R. Cook. A metainstrument for interactive, on-the-fly machine learning. *Proc. NIME*, 2:3, 2009.
6. A Holzinger, M Plass, K Holzinger, GC Crisan, CM Pintea, and V Palade. Towards interactive machine learning (iml): Applying ant colony algorithms to solve the traveling salesman problem with the human-in-the-loop approach. In *IFIP International Cross Domain Conference and Workshop (CD-ARES)*, pages 81–95. Springer, Heidelberg, Berlin, New York, 2016.
7. Andreas Holzinger. Interactive machine learning for health informatics: When do we need the human-in-the-loop? *Springer Brain Informatics (BRIN)*, 3(2):119–131, 2016.
8. Peter Kieseberg, Bernd Malle, Peter Frhwirt, Edgar Weippl, and Andreas Holzinger. A tamper-proof audit and control system for the doctor in the loop. *Brain Informatics*, pages 1–11, 2016.
9. Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE 23rd International Conference on Data Engineering, ICDE 2007*, pages 106–115. IEEE, 2007.
10. Brian C.S. Loh and Patrick H.H. Then. Ontology-enhanced interactive anonymization in domain-driven data mining outsourcing. *Proceedings - 2nd International Symposium on Data, Privacy, and E-Commerce, ISDPE 2010*, (June):9–14, 2010.
11. Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):1–52, 2007.
12. Bernd Malle, Peter Kieseberg, and Andreas Holzinger. Do not disturb? classifier behavior on perturbed datasets. In *Machine Learning and Knowledge Extraction, IFIP CD-MAKE, Lecture Notes in Computer Science LNCS Volume 10410*, pages 155–173. Springer, Cham, 2017.
13. Bernd Malle, Peter Kieseberg, Edgar Weippl, and Andreas Holzinger. The right to be forgotten: towards machine learning on perturbed knowledge bases. In *International Conference on Availability, Reliability, and Security*, pages 251–266. Springer, 2016.
14. Carlos Moque, Alexandra Pomares, and Rafael Gonzalez. AnonymousData.co: A Proposal for Interactive Anonymization of Electronic Medical Records. *Procedia Technology*, 5:743–752, 2012.
15. M. E. Nergiz and C. Clifton. delta-presence without complete world knowledge. *IEEE Transactions on Knowledge and Data Engineering*, 22(6):868–883, 2010.

16. Pierangela Samarati. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.
17. Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):571–588, 2002.
18. Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
19. MALCOLM WARE, EIBE FRANK, GEOFFREY HOLMES, MARK HALL, and IAN H WITTEN. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001.
20. Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Interactive anonymization of sensitive data. *Proceedings of the 35th SIGMOD international conference on Management of data - SIGMOD '09*, page 1051, 2009.

Transfer learning for time series anomaly detection

Vincent Vercruyssen, Wannes Meert, and Jesse Davis

Dept. of Computer Science, KU Leuven, Belgium
firstname.lastname@cs.kuleuven.be

Abstract. Currently, time series anomaly detection is attracting significant interest. This is especially true in industry, where companies continuously monitor all aspects of production processes using various sensors. In this context, methods that automatically detect anomalous behavior in the collected data could have a large impact. Unfortunately, for a variety of reasons, it is often difficult to collect large labeled data sets for anomaly detection problems. Typically, only a few data sets will contain labeled data, and each of these will only have a very small number of labeled examples. This makes it difficult to treat anomaly detection as a supervised learning problem. In this paper, we explore using transfer learning in a time-series anomaly detection setting. Our algorithm attempts to transfer labeled examples from a source domain to a target domain where no labels are available. The approach leverages the insight that anomalies are infrequent and unexpected to decide whether or not to transfer a labeled instance to the target domain. Once the transfer is complete, we construct a nearest-neighbor classifier in the target domain, with dynamic time warping as the similarity measure. An experimental evaluation on a number of real-world data sets shows that the overall approach is promising, and that it outperforms unsupervised anomaly detection in the target domain.

Keywords: transfer learning; anomaly detection; time series

1 Introduction

Time series data frequently arise in many different scientific and industrial contexts. For instance, companies use a variety of sensors to continuously monitor equipment and natural resources. One relevant use case is developing algorithms that can automatically identify time series that show anomalous behavior. Ideally, anomaly detection could be posed as a supervised learning problem. However, these algorithms require large amounts of labeled training data. Unfortunately, such data is often not available as obtaining expert labels is time-consuming and expensive. Typically, only a small number of labels are known for a limited number of data sets. For example, if a company monitors several similar machines, they may only label events (e.g., shutdown, maintenance...) for a small subset of them.

Transfer learning is an area of research focused on methods that are able to extract information (e.g., labels, knowledge, etc.) from a data set and reapply it in another, different data set. Specifically, the goal of transfer learning is to improve performance on the target domain by leveraging information from a related data set called the source domain [10]. In this paper, we adopt the paradigm of transfer learning for anomaly detection. In our setting, we assume that labeled examples are only available in the source domains, and that there no labeled examples in the target domain. In the example, we utilize the label information available for machine A to help constructing an anomaly detection algorithm for machine B, where no labeled points are available for machine B.

In this paper we study transfer learning in the context of time-series anomaly detection, which has received less attention in transfer learning [1, 6, 10]. Our approach attempts to transfer instances from the source domain to the target domain. It is based on two important and common insights about anomalous data points, namely that they are infrequent and unexpected. We leverage these insights to propose two different ways to identify which source instances should be transferred to the target domain. Finally, we make predictions in the target domain by using 1-nearest neighbors classifier where the transferred instances are the only labeled data points in the target domain. We experimentally evaluate our approach on a large data set adapted from a real-world data set and find that it outperforms an unsupervised approach.

2 Problem statement

We can formally define the task we address in this paper as follows:

Given: One or multiple source domains D_S with source domain data $\{X_S, Y_S\}$, and a target domain D_T with target domain data $\{X_T, Y_T\}$, where the instances $x \in X$ are time series and the labels $y \in Y$ are $\in \{\texttt{anomaly}, \texttt{normal}\}$. Additionally, only partial label information is available in the source domains, and no label information in the target domain.

Do: Learn a model for anomaly detection $f_T(\cdot)$ in the target domain D_T using the knowledge in D_S, and $D_S \neq D_T$.

Both the source and target domain instances are time series. Thus each instance $x = \{(t_1, v_1), \ldots, (t_n, v_n)\}$, where t_i is a time stamp and v_i is a single measurement of the variable of interest v at time t_i. The problem has the following characteristics:

- The joint distributions of source and target domain data, denoted by $p_S(X, Y)$ and $p_T(X, Y)$, are not necessarily equal.
- No labels are known for the target domain, thus $Y_T = \varnothing$. In the source domain, (partial) label information is available.
- The same variable v is monitored in the source and target domain, under possibly different conditions (e.g., the same machine in different factories).
- The number of samples in the D_S and D_T are denoted respectively by $n_S = |X_S|$ and $n_T = |X_T|$, and no restrictions are imposed on them.

- Each time series in D_S or D_T has the same length d.
- The source and target domain instances are randomly sampled from the true underlying distribution.

3 Context and related work

Several flavors of transfer learning distinguish themselves in the way knowledge is transferred between source and target domain. In this paper we employ instance-based transfer learning. The idea is to transfer specific (labeled) instances from the source domain to the target domain in order to improve learning a target predictive function $f_T(\cdot)$ [6]. In the case of anomaly detection, the target function is a classifier that aims to distinguish normal instances from anomalous instances. However, care needs to be taken when selecting which instances to transfer, because transferring all instances could result in degraded performance in the target domain (i.e., negative transfer) [8]. A popular solution is to define a weight for each transferred instance based on the similarity of the source and target domain. The latter is characterized either by the similarity of the marginal probability distributions $p_S(X)$ and $p_T(X)$, and/or the similarity of conditional probability distributions $p_S(Y|X)$ and $p_T(Y|X)$. Various ways of calculating these weights have been proposed [3, 6, 10]. However, the problem outlined in this paper states that $Y_T = \varnothing$, which is a realistic assumption given that in practice labeling is expensive. Hence, we cannot easily calculate $p_T(Y|X)$. Furthermore, even if the marginal distributions are different, it can still be beneficial to transfer specific instances. Consider the following. Since the target task is anomaly detection, one cares for a classifier that robustly characterizes normal behavior. By adding a diverse set of anomalies to the training data of the classifier, the learned decision surfaces will be more restricted, ensuring a decrease of type 2 errors when detecting anomalies in new, unseen data.

The subject of instance-based transfer learning for time series has received less attention in literature. Spiegel recently proposed a mechanism for learning a target classifier using set of unlabeled time series in various source domains, without assuming that source and target domain follow the same generative distribution or even have the same class labels [7]. However, they require a limited set of labels in the target domain, whereas we have $Y_T = \varnothing$.

4 Methodology

In order to learn the model for anomaly detection $f_T(\cdot)$ in the target domain, we transfer labeled instances from different source domains. To avoid situations of negative transfer (e.g., transferring an instance with the label `anomaly` that maps to a normal instance in the target domain), a decision function decides whether to transfer an instance or not. First, we outline the intuitions behind the decision function based on two commonly known characteristics of anomalous instances (Sec. 4.1). Then, we propose two distinct decision functions (Sec. 4.2 and 4.3). Finally, we describe a method for supervised anomaly detection in the target domain based on the transferred instances (Sec. 4.4).

4.1 Instance-based transfer learning for anomaly detection

The literature frequently makes two important observations about anomalous data:

Observation 1 *Anomalies occur infrequently [2].*

Observation 2 *If a model of normal behavior is learned, then anomalies constitute all unexpected behavior that falls outside the boundaries of normal behavior. This implies that it is impossible to predefine every type of anomaly.*

From the first observation we derive the following property:

Property 1 *Given a labeled instance $(x_S, y_S) \in D_S$ and $y_S =$ `normal`. If the probability of the instance under the true target domain distribution $p_T(x_S)$ is high (i.e., the instance is likely to be sampled from the target domain), then the probability that the true label of the instance in the target domain is* `normal`*, $p_T(y_S =$ `normal`$|x_S)$ is also high.*

The second observation allows us to derive the reverse property:

Property 2 *Given a labeled instance $(x_S, y_S) \in D_S$ and $y_S =$ `anomaly`. If the probability of the instance under the true target domain distribution $p_T(x_S)$ is low, then the probability that the true label of the instance in the target domain is* `anomaly`*, $p_T(y_S =$ `anomaly`$|x_S)$ is high.*

Notice that in the latter property the time series x_S can have any form, while this is not true for the first property, where the form is restricted by the distribution of the target domain data. Given a labeled instance $(x_S, y_S) \in D_S$ that we want to transfer to the target domain, **Property 1** and **Property 2** allow us to make a decision whether to transfer or not. We can formally define a weight associated with x_S which will be high when the transfer makes sense, and low when it will likely cause negative transfer.

$$w_S = \begin{cases} p_T(x_S) & \text{if } y_S = \texttt{normal} \\ 1 - p_T(x_S) & \text{if } y_S = \texttt{anomaly} \end{cases} \tag{1}$$

However, since each time series x_S can be considered as a vector of length d in $\mathbb{R}^d$ (i.e., it consists of a series of numeric values for continuous variable v), the probability of observing exactly x_S under the target domain distribution must be 0. Instead, we calculate the probability of observing a small interval around x_S, such that:

$$p_T(x_S) = \lim_{\Delta I \to 0} \int_{\Delta I} p_T(x_S) dx \tag{2}$$

where ΔI is an infinitesimally small region around x_S in the target domain. This probability is equal to the true density function over the target domain $f_T(x_S)$. Given that the true target domain density is unknown, we need to estimate it

from the data X_T. It is shown that this estimate $\hat{f_T}(x_S)$ can be calculated as follows [4]:

$$\hat{f_T}(x_S) = \frac{1}{n_T} \frac{1}{(h_{n_T})^d} \sum_{i=1}^{n_T} K\left(\frac{x_S - x_i}{(h_{n_T})^d}\right) \tag{3}$$

where $K(x)$ is the window function or kernel in the d-dimensional space and $\int_{\mathbb{R}^d} K(x)dx = 1$. The parameter $h_{n_T} > 0$ is the *bandwidth* corresponding to the width of the kernel, and depends on the number of observations n_T. The estimate $\hat{f_T}(x_S)$ converges to the true density $f_T(x_S)$ when there is an infinite number of observations, $n_T \to \infty$, under the assumption that the data X_T are randomly sampled from the true underlying distribution.

4.2 Density-based transfer decision function

For guaranteeing convergence of $\hat{f_T}(x_S)$ to the true density function, the sample size must increase exponentially with the length d of the time series data. The reasoning is clear; high-dimensional spaces are sparsely populated by the available data, making it hard to produce accurate estimates. However, this is often infeasible in practice (gathering data is expensive). For longer time series d is automatically high, that is, if we treat the time series as a vector in $\mathbb{R}^d$. As a practical solution, we propose to reduce the length d of the time series x_S by dividing it into l equal-length subsequences, each with length $m < d$. For every subsequence s in x_S, the density is estimated using Eq. 3 with a Gaussian kernel:

$$\hat{f}_{T,m}(s) = \frac{1}{n_T} \frac{1}{(h_{n_T}\sqrt{2\pi})^m} \sum_{i=1}^{n_T} \exp\left(-\frac{1}{2}\left(\frac{s - s_i}{h_{n_T}}\right)^2\right) \tag{4}$$

where h_{n_T} is the standard deviation of the Gaussian, and s_i are the subsequences of the instances in X_T. The Gaussian kernel ensures that instead of simply counting similar subsequences, the count is weighted for each subsequence s_i based on the kernelized distance to s_S.

Estimating the densities for the subsequences yields more accurate estimates given the reduced dimensionality, but simultaneously results in $l = m/d$ estimates for each time series x_S. Hence, we have to adjust Eq. 1 to reflect this new situation. We only show the case in which the label $y_S = \texttt{normal}$ as the reverse case is straightforward:

$$w_S = \frac{1}{Z_{max} - Z_{min}} \left(\sum_{i=1}^{l} \hat{f_{T,m}}(s_i) - Z_{min}\right) \tag{5}$$

$$Z_{max} = \max_{x_T \in \{X_T \cup x_S\}} \sum_{s_j \in x_T} \hat{f_{T,m}}(s_j) \tag{6}$$

The sum of the density estimates in the subsequences is normalized using min-max normalization, such that $w_S \in [0, 1]$. Z_{min} is calculated similarly as Z_{max} in Eq. 6, but taking the minimum instead of maximum. By setting a threshold on the final weights, we make a decision on whether to transfer or not.

4.3 Cluster-based transfer decision function

Our second proposed decision function is also based on the intuitions outlined in Sec. 4.1. First, the target domain data X_T are clustered using k-means clustering. Second, the resulting set of clusters C over X_T is divided into a set of large clusters, and a set of small clusters according to the following definition [5]:

Definition 1. *Given a dataset X_T with n_T instances, a set of ordered clusters $C = \{C_1, ..., C_k\}$ such that $|C_1| \geq |C_2| \geq ... \geq |C_k|$, and two numeric parameters α and β, the boundary b between large and small clusters is defined such that either of the following conditions holds:*

$$\sum_{i=1}^{b} |C_i| \geq n_T \times \alpha \tag{7}$$

$$\frac{|C_b|}{|C_{b+1}|} \geq \beta \tag{8}$$

$LC = \{C_i | i \leq b\}$ and $SC = \{C_i | i > b\}$ are respectively the set of large and small clusters, and $LC \cup SC = C$.

Furthermore, we define the radius of a cluster as $r_i = \max_{x_j \in C_i} \|x_j - c_i\|^2$. Lastly, a decision is made whether or not to transfer a labeled instance x_S from the source domain. Intuitively, and in line with **Observation 1 and 2**, anomalies in X_T should fall in small clusters, while large clusters contain the normal instances. Transferred labeled instances from the source domain should adhere to the same intuitions. Each transferred instance is assigned to a cluster $C_i \in C$ such that $\|x_S - c_i\|^2$ is minimized. An instance is only transferred in two cases. First, if the instance has label `normal` and is assigned to a cluster C_i such that $C_i \in LC$ and the distance of the instance to the cluster center is less or equal to the radius of the cluster. Second, if the instance has label `anomaly` and fulfills either of two conditions: the instance is assigned to a cluster C_i such that $C_i \notin LC$, or it is assigned to a cluster C_i such that $C_i \in LC$ and the distance of the instance to the cluster center is larger than the radius of the cluster. In all other cases there is no transfer.

4.4 Supervised anomaly detection in a set of time series

After transferring instances from one or multiple source domains to the target domain using the decision functions in Sec. 4.2 and 4.3, we can construct a classifier in the target domain to detect anomalies. Ignoring the unlabeled target domain data, we only use the set of labeled data $L = \{(x_i, y_i)\}_{i=1}^{n_A}$, n_A being the number of instances transferred. It has been shown that one-nearest-neighbor (1NN) classifier with dynamic time warping (DTW) or Euclidean distance is a strong candidate for time series classification [9]. To that end, we construct a 1NN-DTW classifier on top of L to predict the labels of unseen instances.

5 Experimental evaluation

In this section we aim to answer the following research questions:

– Do the proposed decision functions for instance-based time series transfer succeed in transferring useful knowledge between source and target domain.

First, we introduce the unsupervised baseline method to which we will compare the 1NN-DTW method with instance transfer (Sec. 5.1). Then, we discuss the data, the experimental setup, and the results (Sec. 5.2).

5.1 Unsupervised anomaly detection in a set of time series

Without instance transfer, the target domain consists of a set of unlabeled time series data $U = \{(x_i)\}_{i=1}^{n_T}$. Based on the anomaly detection approach outlined in Kha et al., we introduce a straightforward unsupervised algorithm for anomaly detection that will serve as a baseline [5]. The algorithm calculates the *cluster based local outlier factor* (CBLOF) for each series in U.

Definition 2. *Given a set of large LC and small clusters SC defined over U (as per definition 1), the CBLOF of an instance $x_i \in U$, belonging to cluster C_i, is calculated as:*

$$CBLOF(x_i) = \begin{cases} |C_i| \times D(x_i, c_i) & \text{if } C_i \in LC \\ |C_i| \times \min_{c_j \in LC} D(x_i, c_j) & \text{if } C_i \in SC \end{cases} \tag{9}$$

Then, anomalies are characterized by a high CBLOF.

5.2 Experiments

Data. Due to the lack of readily available benchmarks for the problem outlined in Sec. 2, we experimentally evaluate on a real-world data set obtained from a large company. The provided data detail resource usage continuously tracked over a period of approximately two years. Since the usage is highly dependent on the time of day, we can generate 24 (hourly) data sets by grouping the usage data by hour. Each data set contains about 850 different time series. For a limited number of these series in each set we possess expert labels indicating either `normal` or `anomaly`.

Experimental setup. In turn, we treat each of the 24 data sets as the target domain and the remaining data sets as source domains. We consider transferring from a single source or multiple sources. Any labeled examples in the target domain are set aside and serve as the test set. First, the proposed decision functions are used to transfer instances from either a single source domain or multiple source domains combined to the target domain. Then, we train both the unsupervised CBLOF (Sec. 5.1), and supervised 1NN-DTW anomaly detection model that uses the labeled instances transferred to the target domain

(Sec. 4.4). Finally, both models predict the labels of the test set, and we report classification accuracy. For the density-based approach, we set the threshold on the final weights to 0.5. For the cluster-based approach we selected $\alpha = 0.95$, $\beta = 4$, and the number of clusters 10.

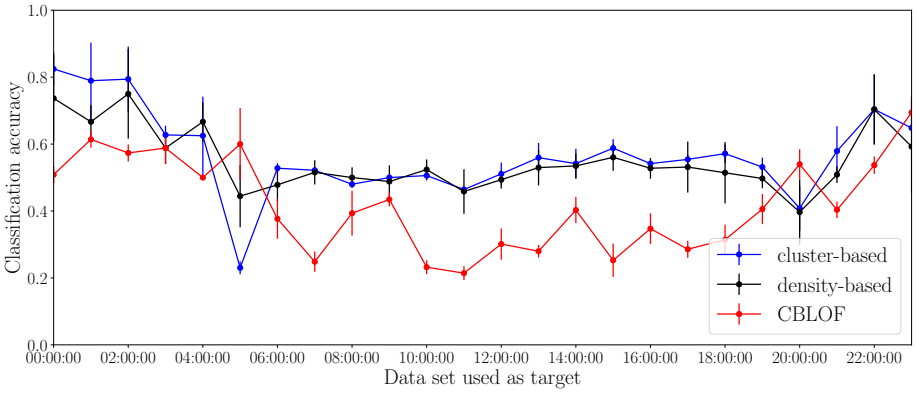


Fig. 1: The graph plots the mean classification accuracy and the standard deviation for each of the 24 (hourly) data sets. These statistics are calculated after considering 7 randomly chosen data sets as source domains, and performing the analysis for each combination of source and target. The plot indicates both transfer approaches with 1NN-DTW perform quite similarly, while outperforming the unsupervised method in 21 of the 24 data sets.

Evaluation. A limited excerpt of the experimental results is reported in Table 1. Figure 1 plots the full experimental results in a condensed manner. From the results we derive the following observations. First, instance transfer with 1NN-DTW outperforms the unsupervised CBLOF algorithm in 21 of the 24 data sets. Clearly, this indicates that the instances that are transferred by both decision functions, are useful in detecting anomalies. Second, the transfer works both between similar and dissimilar domains. To see this, one must know that in our real-world data set resource usage during the night is very different from usage during the day. As a result, the data sets at 00:00 and 01:00 are fairly similar for example, while data sets at 21:00 and 15:00 are highly different. From Table 1 it is clear that this distinction has little impact on the performance of the 1NN-DTW model. Third, the cluster-based decision function performs at least as well as the density-based variant. This is apparent from Figure 1.

6 Conclusion

In this paper we introduced two decision functions to guide instance-based transfer learning in case the instances are time series and the task at hand is anomaly detection. Both functions are based on two commonly knowns insights about anomalies: they are infrequent and unexpected. We experimentally evaluated

Table 1: A limited excerpt of the experimental evaluation. The number of transferred instances is denoted by n_A. *Density-based* is the density-based decision function with 1NN-DTW anomaly detection. *Cluster-based* is the cluster-based decision function with 1NN-DTW. *CBLOF* is the unsupervised anomaly detection. All reported numbers are classification accuracies on a hold-out test set in the target domain, rounded off. *Combo* is the the combination of 7 separate, randomly chosen source domains.

		Cluster-based		Density-based		CBLOF
Source	Target	n_A	Result	n_A	Result	Result
01:00	00:00	14	**89%**	13	**89%**	58%
03:00	00:00	11	**79%**	11	74%	52%
21:00	00:00	10	**79%**	9	58%	52%
combo	00:00	60	**90%**	46	85%	63%
03:00	06:00	6	**52%**	5	**52%**	39%
11:00	06:00	15	**56%**	8	**56%**	35%
21:00	06:00	7	**52%**	8	48%	39%
combo	06:00	79	**57%**	54	44%	35%
03:00	15:00	6	**58%**	5	**58%**	23%
11:00	15:00	19	**65%**	9	58%	30%
21:00	15:00	7	**58%**	7	**58%**	19%
combo	15:00	85	**67%**	54	54%	27%
03:00	19:00	6	**52%**	5	**52%**	44%
11:00	19:00	16	**60%**	8	48%	40%
21:00	19:00	7	**52%**	8	44%	40%
combo	19:00	81	**56%**	50	48%	44%

the proposed decision functions in combination with a 1NN-DTW classifier by comparing it to an unsupervised anomaly detection algorithm on a real-world data set. The experiments showed that the transfer-based approach outperforms the unsupervised approach in 21 of the 24 data sets. Additionally, both decision functions lead to similar results.

References

1. Andrews, J.T., Tanay, T., Morton, E., Griffin, L.: Transfer representation-learning for anomaly detection. ICML (2016)
2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) 41(3), 1–72 (2009)
3. Chattopadhyay, R., Sun, Q., Fan, W., Davidson, I., Panchanathan, S., Ye, J.: Multisource domain adaptation and its application to early detection of fatigue. ACM Transactions on Knowledge Discovery from Data (TKDD) 6(4), 18 (2012)
4. Fukunaga, K.: Introduction to statistical pattern recognition. Academic press (2013)
5. Kha, N.H., Anh, D.T.: From cluster-based outlier detection to time series discord discovery. In: Revised Selected Papers of the PAKDD 2015 Workshops on Trends and Applications in Knowledge Discovery and Data Mining-Volume 9441. pp. 16–28. Springer-Verlag New York, Inc. (2015)

6. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22(10), 1345–1359 (2010)
7. Spiegel, S.: Transfer learning for time series classification in dissimilarity spaces. In: Proceedings of AALTD 2016: Second ECML/PKDD International Workshop on Advanced Analytics and Learning on Temporal Data. p. 78 (2016)
8. Torrey, L., Shavlik, J.: Transfer learning. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques 1, 242 (2009)
9. Wei, L., Keogh, E.: Semi-supervised time series classification. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 748–753. ACM (2006)
10. Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big Data 3(1), 9 (2016)

Probabilistic Active Learning with Structure-Sensitive Kernels

Dominik Lang[1], Daniel Kottke[2], Georg Krempl[1], and Bernhard Sick[2]

[1] KMD Lab, Faculty of Computer Science,
Otto-von-Guericke University, Magdeburg, Germany
{dominik.lang / georg.krempl}@ovgu.de
[2] IES Group, Faculty of Computer Science,
University of Kassel, Germany
{daniel.kottke / bsick}@uni-kassel.de

Abstract. This work proposes two approaches to improve the pool-based active learning strategy '*Multi-Class Probabilistic Active Learning*' (McPAL) by using two kernel functions based on Gaussian mixture models (GMMs). One uses the kernels for the instance selection of the McPAL strategy, the second employs them in the classification step. The results of the evaluation show that using a different classification model from the one that is used for selection, especially an SVM with one of the kernels, can improve the performance of the active learner in some cases.

Keywords: active learning, gaussian mixture, kernel function, support vector machine, McPAL

1 Introduction & Motivation

Active learning (AL) is a special case of semi-supervised machine learning, in which a learning algorithm has both labeled and unlabeled data available to it and is able to acquire the true labels of instances from an external source, in most cases one or multiple human agents. Since the number of labels that can be acquired is limited due to the cost that the acquisition entails, AL strategies aim to select instances that maximize the learners classification performance while being efficient with respect to the costs. A pool-based AL strategy named '*Multi-Class Probabilistic Active Learning*' (McPAL) [10] has shown to outperform competing strategies. This paper investigates the possibility of improving the performance of the method by including the information captured by a Gaussian mixture model (GMM) into the active learner. To achieve this, two kernel functions that are based on a GMM are used. These structure-sensitive kernel functions, based on the GMM [1] and RWM [17] distance measures, are leveraged by the active learner in two different ways: (1) by being included in the computation of the McPAL score, (2) by being used in the model that performs classification based on the sampled set of labeled instances. These approaches are compared to the original McPAL method as well as random sampling in a

series of experiments on one artificial data set and nine real-world data sets from the UCI machine learning repository [14].

2 Related Research

In active learning, various criteria have been proposed to determine which instances are most helpful to learn a classification model. One of the most common is the model's uncertainty regarding the classification of a sample. A strategy that solely relies on this criterion is known as uncertainty sampling (US) [13, 18]. In their application of US to SVMs, Tong and Koller [20] motivated that the goal is to approximately halve the version space through selecting instances that lie closest to the current decision boundary of the classifier. To extend this to multi-class problems, many methods have been proposed, for example, 'Best-versus-Second-Best' [8, 9, 11] (also referred to as 'Margin Sampling' [18]) or Entropy-based sampling [8, 18, 9]. Solely relying on this criterion to select instances has been shown to be prone to being 'locked in', ignoring possibly informative instances in favor of refining the current decision boundary [18, 12]. Hence, various approaches have been proposed that, in addition to uncertainty, also include other criteria. These include, for example, the diversity of the sampled instances [2, 5] or the density around a candidate instance [3, 4]. A promising AL strategy is Multi-class Probabilistic Active Learning (McPAL) [10], which has shown promising results compared to other approaches. It combines the density, the class posterior probability and the number of already sampled instances in the neighborhood of a candidate instance to estimate the potential gain of acquiring an instance's label. Instances that entail the highest potential gain are selected for labeling by the strategy. This acquisition is performed in a one-by-one fashion.

However, the selection does not have to be solely based on supervised models but can also use unsupervised approaches. Known clustering algorithms like k-medoid [15] or hierarchical clustering [6] as well as generative models like GMMs [16, 7] can be used to model the structure of the data and include it in the selection process.

The classification models, that are used in the process of instance selection, are often used for training the final classifier. Tomanek & Morik [19] investigated, to which degree the bias towards the learning algorithm used in the selection process affects, what they call *label reusability* - i.e., the training of a classifier, other than the one used for selection, on the acquired labeled data. They introduced the terms of *selector* and *consumer* classifiers to describe the model used for selecting instances, and performing classification based on them, respectively. Contrary to their initial assumptions, they concluded that *self-selection* (the selector and consumer classifier are the same) is in fact not in all cases the best choice.

3 Using GMM-based Kernels with the McPAL Strategy

Since the majority of data available in the scenario of AL is unlabeled and therefore carries no explicit information about the mapping of $f : x \mapsto y$, implicit information contained in the structure of the data becomes even more important. The GMM ([17], Eq. 6) and RWM ([17], Eq. 7) distance measures (denoted as Δ_{GMM} and Δ_{RWM}) are based on Gaussian mixture models (GMMs). A GMM models data with a number of J multivariate Gaussian distributions[3]. To speed up the training process, first k-means clustering is performed to find J clusters in the data. Based on the samples belonging to the clusters, the initial means and variances are computed to initialize the Gaussian distributions. Then the components are refined, either with the *Expectation Maximization* (EM) or *Variational Inference* (VI) training method [1], using only the feature vectors x of the samples. The GMMs used in this paper are trained with the VI method. The result of the training is a GMM with J components, component weights ϕ_j[4] that determine the influence of the component in the mixture, as well as the component covariance matrices Σ_j. Building on such a mixture model, the GMM [1] and RWM distance measures [17] consist of the Mahalanobis distance of two instances a and b with respect to the covariance matrices of the mixture model, weighted in two different ways: (1) the distance is weighted by the mixture coefficients of the model (GMM-distance, Eq. 1); (2) the distance is weighted by half the sum of the components responsibilities for the two instances (RWM-distance, Eq. 2). Both measures include the information captured by the GMM into the distance measure. The resulting distance is small, if both instances lie closest to the same GMM component, and large, if their closest GMM components differ. These distance measures are incorporated into kernel functions by substituting the Euclidean distance in the Gaussian RBF kernel with the GMM or RWM distance respectively [17]. The kernel functions thereby keep the parameter γ of the RBF kernel. These kernel functions can be used in kernel-based learning methods like SVMs or Parzen-Window kernel density estimation.

$$\Delta_{GMM}(a,b) = \sum_{j=1}^{J} \phi_j \Delta_{\Sigma_j}(a,b)) \tag{1}$$

$$\Delta_{RWM}(a,b) = \sum_{j=1}^{J} (\frac{1}{2}(p(j|a) + p(j|b))\Delta_{\Sigma_j}(a,b)) \tag{2}$$

The research question of this work is whether the inclusion of structural information by means of such kernels improves the performance of the McPAL approach. To examine that this work investigates two possible ways the McPAL strategy can employ such kernels and to which extent these benefit the strategy. The first approach of using these structure-sensitive kernel functions in combination with

[3] These distributions are referred to as 'components'

[4] As part of the VI training method, the weights of some components can be set close to zero, effectively 'pruning' them from the model

the McPAL strategy is incorporating them into the process of instance selection. To this end, two changes to the method are made that are described in the following.

First, the GMM/RWM kernels replace the Gaussian RBF kernel in the computation of the kernel frequency estimates (denoted as $\overrightarrow{k}$ in Eq. 4), which are required by the McPAL method, by means of the Parzen-Window method. These frequency estimates are computed in the same way as in the original McPAL approach [10], i.e. by computing the kernel density estimates with the Parzen-Window method, but leaving out the normalization by the number of samples.

$$k_{x,y} = \sum_{\{(x',y');y'=y\}} K_{GMM/RWM}(x, x') \tag{3}$$

$$\overrightarrow{k}_x = \{k_{x,y_1}, k_{x,y_2}, \cdots, k_{x,y_n}\} \tag{4}$$

Second, instead of using Parzen-Window estimation, the density estimates are directly taken from the GMM used by the GMM and RWM kernels. The Parzen-Window method places a kernel K with bandwidth h on each of the N samples in the data set, with each of them equally contributing to the resulting density estimate (s. Eq. 5). The GMM uses a fixed number of J multivariate Gaussians $\mathcal{N}$ to model the data, the contribution of each of these components being weighted by the mixture coefficient or component weight ϕ of the component (s. Eq. 6). These changes enable the McPAL strategy to use the information provided by the GMM and RWM kernels in the instance selection process. For the purpose of disambiguation, this modified version of the McPAL strategy is in the following referred to as *StrucPAL*.

$$p(x) = \frac{1}{N}\sum_{i=1}^{N} K_h(x - x_i) = \frac{1}{Nh}\sum_{n=1}^{N} K(\frac{x - x_n}{h}) \tag{5}$$

$$p(x) = \sum_{j=1}^{J} \phi_j \mathcal{N}(x|\mu_j, \Sigma_j) \tag{6}$$

The second approach to use structure-sensitive kernels to improve the performance of the McPAL strategy is by using them in the consumer classifier. This is possible in two ways, either as *'self-selection'* or *'foreign-selection'*. Tomanek & Morik [19] use these terms to refer to, in the first case, the selector and consumer classifiers being the same, or in the second case, the selector and consumer classifiers being different. Therefore, two scenarios for using the GMM and RWM kernels in the classification process are possible. The first is the StrucPAL method being used with self-selection, so pwc_{rwm} or pwc_{gmm} act as both selector and consumer classifier respectively. The second is that the McPAL or StrucPAL strategy is employed for instance selection but classification is performed by a foreign classifier which uses the GMM or RWM kernels - i.e. Parzen-Window

classifier or a SVM.
This work aims to investigate two questions regarding the use of structural information by the McPAL method, in order to gain additional insight into what approaches are worth exploring in future research.
The first question is whether, and to what extent, the performance of the StrucPAL method differs from the original McPAL method. Due to the already mentioned inclusion of the information of the underlying mixture model into the instance selection process, a positive impact on the performance is expected.
The second question is if and to what extent the McPAL and StrucPAL learners benefit from foreign-selection. The first part of this question is to investigate, how the performance of McPAL and StrucPAL learners using self-selection compares to using a SVM with the same kernel as the selector as consumer classifier. The second part of this question is to investigate, how the original McPAL strategy can benefit from consumer classifiers that use the GMM or RWM kernels.

4 Experiments

In our experiments, eight data sets from the UCI machine learning repository [14] are used, namely australian, glass, haberman, heart, qsar-biodeg[5], steel-plates-fault[6], vehicle and vertebral. Furthermore, the phoneme data set from OpenML [21] is used. In addition to that, an artificial 2d data set referred to as blobs is used, consisting of three Gaussians that make up the classes.
The experiments include three AL strategies: *McPAL* (mp), *StrucPAL* (sp) and random sampling (rl). As classifiers, the Parzen-Window classifier (pwc) and support vector machine classifier (svm) are used. The PWC and SVM classifiers use either the Gaussian RBF, the RWM or the GMM kernels, as introduced earlier. The kernel that the classifier uses is denoted in subscript, for example pwc_{rbf}. An active learner in the experiments consists of three components: the AL strategy (al), the selector classifier (cl_{al}) and the consumer classifier (cl). In the case of self-selection, cl and cl_{al} are identical.
Based on a set of 10 seeds for randomization, for each seed, the data sets are split using five-fold stratified cross-validation. One fold per split is used as holdout set to test the trained consumer classifier, while the four remaining folds are used as training data. The initial labeled set $\mathcal{L}$ is initialized with one instance from two randomly picked classes in the training data. The random choice is based on the seed used in the current cross-validation split. Starting with $\mathcal{L}$ initialized with 2 instances and the rest of the training set comprising the unlabeled set $\mathcal{U}$, pool-based AL is performed. As part of this, the labels of 60 instances in total are acquired in a one-by-one fashion with both the selector and consumer classifier being updated after each acquisition. Then the consumer classifier is evaluated on the holdout set using the accuracy metric. This process is repeated until every fold has been used as test set once. The performance scores at every point in the AL process are averaged over all folds. After each of the 10 seeds

[5] in the following abbreviated as 'qsar'
[6] in the following abbreviated as 'steel'

has been used to seed the cross-validation split, the final results are gained by computing the average accuracy per step in the AL process and the standard deviation of the accuracy over all seeds.
The hyperparameters of the models for each data set are determined by performing an exhaustive search over a parameter grid on a subset of the data. This subset is a stratified, seed-based[7] random subsample consisting of 90 instances. The 90 instances are split using three-fold stratified cross-validation, with one fold being used to train a classifier with a given set of parameters, while the other two folds are used to evaluate the performance of this classifier. The small size of this tuning data set is founded in the fact, that in AL applications there is little labeled data available, therefore performing model selection with a large tuning set would be unrealistic. However, a review of the literature on active and semi-supervised learning did not provide a fitting way to determine the hyperparameters without using more labeled data than would be available in this scenario.

5 Results & Discussion

In the following, the results for the scenarios of self-selection and foreign-selection are presented in two ways.
First, the average accuracy scores and the corresponding standard deviation is tabulated for the different active learners on each data set. The highest accuracy score on a data set is printed in bold font. In the case of learners scoring equally, a lower standard deviation decides the winner. In case these are also identical, the first place is shared by these. For each learner, the difference in accuracy score to the highest score on each data set is computed and averaged for all data sets. This average difference in accuracy to the winners is shown in the column '*diff*' . Based on this difference, the learners are ranked, shown in column '*rank*'.
The second way of illustrating the results is so called *learning curve plots*. These show the performance of the learners on a given data set over the entire AL process, that is for each acquired label.

5.1 Self-Selection

First, the results of the experiments for the scenario of self-selection, shown in Tab. 1, will be considered. In the three moments in the learning process at 10, 20 and 30 acquired labels a good performance of the original McPAL method can be observed. It performs best on 6 of 10 data sets at 10 sampled instances, scoring the first rank in the comparison to the two StrucPAL variants and random selection learners with the Parzen-Window classifier with the RBF, GMM and RWM kernels. At 20 sampled instances, McPAL performs best on 5 data sets, scoring second rank and at 30 sampled instances it is best on 8 data

[7] The seed used for the model selection was not used in the splits for the experiments themselves.

sets, scoring first rank again. Based on these observations a solid performance can be attested to the McPAL strategy, although it does not manage to perform best on the steel and vehicle, where it is beaten by random selection with only one exception (vehicle, 10 sampled instances). The StrucPAL method only manages to perform better than McPAL on the blobs data set at 10 and 20 sampled instances as well as on heart at 20 sampled instances, although Fig. 1 shows an overall better performance of StrucPAL on heart. The gap between the scores of StrucPAL to the best performing learner on each data set varies in size, but when averaged leads to the two StrucPAL learners taking the last two ranks in the ranking.
Concluding the results of the self-selection scenario, the StrucPAL method did not provide better classification performance than the original method. Based on this observation it appears, that including the structural information from the Gaussian mixture model in the selection process did not improve the McPAL method.

Table 1. Mean accuracy scores and std (in brackets) after acquiring 10,20 and 30 labeled instances with Parzen Window Classifier (PWC), using self-selection or random-selection. Abbreviations are explained in Sec. 4.

10 labels sampled	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	.64(±0.08)	.81(±0.09)	**.66**(±0.07)	**.69**(±0.08)	**.77**(±0.06)	**.74**(±0.03)	.62(±0.09)	.55(±0.06)	**.41**(±0.03)	**.61**(±0.07)	.032	1
$pwc_{gmm} + sp$	.64(±0.09)	**.87**(±0.04)	.56(±0.07)	.68(±0.07)	.67(±0.1)	.49(±0.12)	.60(±0.11)	.57(±0.07)	.36(±0.05)	.54(±0.08)	.084	6
$pwc_{rwm} + sp$	.63(±0.11)	**.87**(±0.04)	.54(±0.07)	.68(±0.06)	.76(±0.08)	.62(±0.12)	.57(±0.1)	.55(±0.08)	.38(±0.03)	.53(±0.08)	.069	5
$pwc_{rbf} + rl$	**.69**(±0.08)	.78(±0.11)	.62(±0.08)	.65(±0.08)	.67(±0.1)	.69(±0.05)	**.65**(±0.07)	.68(±0.06)	.39(±0.05)	.60(±0.08)	.040	2
$pwc_{gmm} + rl$	.67(±0.09)	.77(±0.1)	.58(±0.1)	.64(±0.06)	.66(±0.09)	.68(±0.05)	.62(±0.06)	.70(±0.07)	.41(±0.06)	.58(±0.08)	.051	3
$pwc_{rwm} + rl$	.66(±0.1)	.77(±0.1)	.54(±0.09)	.64(±0.06)	.68(±0.11)	.71(±0.04)	.60(±0.07)	**.69**(±0.07)	.39(±0.05)	.60(±0.07)	.054	4
20 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	.68(±0.08)	.88(±0.05)	**.72**(±0.05)	**.73**(±0.05)	.75(±0.06)	**.74**(±0.03)	**.72**(±0.04)	.60(±0.05)	.46(±0.03)	**.67**(±0.06)	.033	2
$pwc_{gmm} + sp$	.66(±0.09)	**.90**(±0.02)	.58(±0.05)	.69(±0.04)	.74(±0.08)	.56(±0.11)	.54(±0.1)	.57(±0.07)	.43(±0.04)	.62(±0.06)	.099	6
$pwc_{rwm} + sp$	.64(±0.09)	**.90**(±0.01)	.54(±0.06)	.71(±0.04)	**.78**(±0.07)	.68(±0.09)	.51(±0.11)	.55(±0.08)	.43(±0.03)	.61(±0.07)	.093	5
$pwc_{rbf} + rl$	**.73**(±0.05)	.86(±0.04)	.70(±0.07)	.71(±0.04)	.73(±0.07)	.72(±0.03)	.69(±0.05)	.74(±0.04)	.48(±0.05)	.65(±0.06)	.027	1
$pwc_{gmm} + rl$	.72(±0.05)	.86(±0.04)	.65(±0.07)	.66(±0.06)	.70(±0.07)	.72(±0.04)	.64(±0.06)	**.78**(±0.04)	**.51**(±0.06)	.63(±0.06)	.041	3
$pwc_{rwm} + rl$	.71(±0.07)	.86(±0.04)	.63(±0.08)	.66(±0.06)	.73(±0.1)	.73(±0.03)	.63(±0.06)	.75(±0.04)	.45(±0.05)	.64(±0.06)	.049	4
30 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.76**(±0.04)	**.90**(±0.01)	**.76**(±0.04)	**.74**(±0.03)	**.79**(±0.04)	**.75**(±0.02)	**.74**(±0.04)	.69(±0.03)	.51(±0.02)	**.69**(±0.05)	.020	1
$pwc_{gmm} + sp$	.69(±0.07)	**.90**(±0.02)	.58(±0.05)	.70(±0.03)	.75(±0.08)	.63(±0.09)	.59(±0.08)	.57(±0.07)	.46(±0.03)	.66(±0.05)	.100	5
$pwc_{rwm} + sp$	.65(±0.08)	.89(±0.02)	.56(±0.07)	.70(±0.04)	.78(±0.07)	.70(±0.06)	.57(±0.09)	.55(±0.08)	.46(±0.03)	.65(±0.07)	.102	6
$pwc_{rbf} + rl$	.75(±0.04)	.88(±0.03)	.74(±0.06)	.72(±0.04)	.75(±0.06)	.74(±0.02)	.71(±0.05)	.77(±0.04)	.52(±0.05)	.68(±0.05)	.027	2
$pwc_{gmm} + rl$	.73(±0.06)	.88(±0.03)	.70(±0.07)	.67(±0.05)	.72(±0.08)	.73(±0.03)	.67(±0.05)	**.82**(±0.04)	**.57**(±0.06)	.67(±0.05)	.037	3
$pwc_{rwm} + rl$	.73(±0.06)	.88(±0.03)	.67(±0.07)	.67(±0.05)	.75(±0.09)	.73(±0.02)	.65(±0.06)	.78(±0.04)	.48(±0.06)	.67(±0.05)	.052	4

5.2 Foreign-Selection

How does foreign-selection affect the result of the active learner? Tab. 2 shows the accuracy scores at the stages of 10, 20 and 30 sampled instances. For every AL strategy, self-selection is compared to the use of an SVM (with the same kernel as the selector) as consumer classifier.
As originally pointed out by Tomanek and Morik [19], it can be observed that foreign-selection can be indeed beneficial with regard to classification performance. However, the extent of this varies in the experiments, ranging from a difference in accuracy of 0.01 to 0.08 and is limited to some of the data sets. Based on the averaged difference in score to the best performing method, self-selection scores better than foreign selection in all three segments. This analysis, however, only included a consumer classifier (SVM), that uses the same kernel function as the selector. In order to investigate, how McPAL learners perform, if

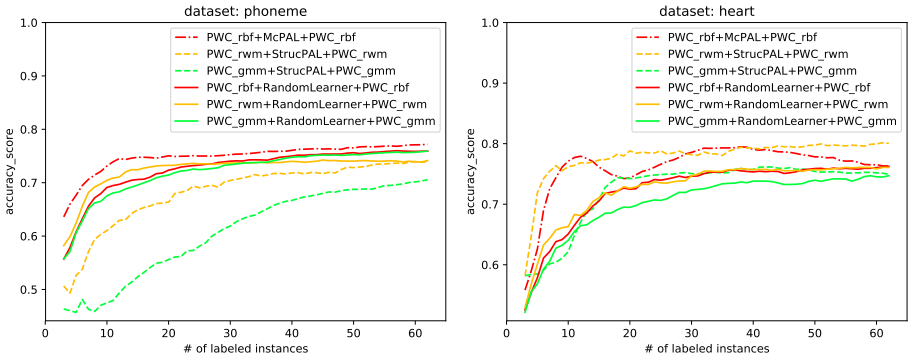


Fig. 1. Learning curves resulting from self-selection on the phoneme and heart data sets

paired with consumer classifiers using the GMM and RWM kernels, a separate tabulation is shown in Tab. 3.

Although the self-selection learner with McPAL+pwc_{rbf} scores the first rank in all three stages, it can be observed that McPAL can benefit from different consumer classifiers. At the stage of 10 sampled instances, an SVM with GMM kernel scores a higher accuracy on the steel (+0.1) and vehicle (+0.07) data sets, with minor gains being provided by an SVM with RBF kernel (+0.01 on qsar, +0.02 on glass) and a SVM with RWM kernel (+0.02 on haberman). However, these gains are accompanied by worse performance than McPAL on other data sets. The advantage provided by the foreign classifiers reduces in the stages of 20 and 30 sampled instances, with svm_{gmm} still showing good gains at 20 sampled instances (+0.05 on steel, +0.1 on vehicle).

Fig. 2 shows the learning curves on the vehicle and steel data sets. While on vehicle a solid advantage of svm_{gmm}, svm_{rbf} and pwc_{gmm} over McPAL in terms of accuracy can be observed, the development on steel is a different one. While the svm_{gmm} and svm_{rwm} learners perform well due to a stagnating but better performance in the early phase, they fail to exploit the additionally acquired labels in the fashion of the other learners, resulting in a slight but increasing advantage for the learners using GMM-based PWCs later, which are in the last phase of the learning process surpassed by svm_{rbf}.

Concluding the results of the foreign-selection scenario it can be summarized, that although self-selection McPAL has performed solidly in the experiments, the results indicate that the use of classifiers with GMM-based kernels in this scenario shows potential and the general use of foreign-selection motivates further research.

Table 2. Mean accuracy scores and std (in brackets) of McPAL and StrucPAL learners using either self-selection or a SVM with the same kernel as the selector for classification.

10 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.64**(±0.08)	**.81**(±0.09)	.66(±0.07)	.69(±0.08)	**.77**(±0.06)	**.74**(±0.03)	.62(±0.09)	.55(±0.06)	.41(±0.03)	**.61**(±0.07)	.006	1
$svm_{rbf} + mp + pwc_{rbf}$	.60(±0.08)	.80(±0.09)	**.68**(±0.07)	**.69**(±0.07)	.62(±0.09)	.71(±0.04)	**.63**(±0.09)	**.56**(±0.06)	**.43**(±0.04)	.57(±0.12)	.027	2
$pwc_{gmm} + sp$	**.64**(±0.09)	**.87**(±0.04)	**.56**(±0.07)	**.68**(±0.07)	**.67**(±0.1)	.49(±0.12)	**.60**(±0.11)	.57(±0.07)	**.36**(±0.05)	**.54**(±0.08)	.009	1
$svm_{gmm} + sp + pwc_{gmm}$	.59(±0.06)	.85(±0.05)	.54(±0.08)	.63(±0.06)	.61(±0.08)	**.51**(±0.12)	.57(±0.13)	**.64**(±0.01)	**.36**(±0.05)	.43(±0.12)	.034	2
$pwc_{rwm} + sp$	.63(±0.11)	**.87**(±0.04)	**.54**(±0.07)	.68(±0.06)	**.76**(±0.08)	**.62**(±0.12)	.57(±0.1)	.55(±0.08)	**.38**(±0.03)	**.53**(±0.08)	.016	1
$svm_{rwm} + sp + pwc_{rwm}$	.61(±0.11)	.84(±0.05)	.53(±0.08)	**.73**(±0.02)	.75(±0.07)	.58(±0.11)	**.60**(±0.11)	**.63**(±0.05)	.37(±0.05)	.49(±0.13)	.016	1
20 labels sampled	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.68**(±0.08)	**.88**(±0.05)	.72(±0.05)	**.73**(±0.05)	**.75**(±0.06)	**.74**(±0.03)	**.72**(±0.04)	**.60**(±0.05)	.46(±0.03)	**.67**(±0.06)	.007	1
$svm_{rbf} + mp + pwc_{rbf}$	.66(±0.09)	.87(±0.04)	**.74**(±0.05)	.72(±0.04)	.66(±0.1)	.72(±0.03)	.72(±0.06)	**.60**(±0.05)	**.51**(±0.04)	.64(±0.07)	.018	2
$pwc_{gmm} + sp$	**.66**(±0.09)	**.90**(±0.02)	**.58**(±0.05)	**.69**(±0.04)	**.74**(±0.08)	**.56**(±0.11)	**.54**(±0.1)	.57(±0.07)	.43(±0.04)	**.62**(±0.06)	.009	1
$svm_{gmm} + sp + pwc_{gmm}$	.63(±0.08)	.88(±0.02)	.57(±0.06)	.67(±0.04)	.70(±0.09)	**.56**(±0.11)	.45(±0.13)	**.64**(±0.01)	**.45**(±0.05)	.59(±0.09)	.024	2
$pwc_{rwm} + sp$	**.64**(±0.09)	**.90**(±0.01)	**.54**(±0.06)	.71(±0.04)	**.78**(±0.07)	**.68**(±0.09)	**.51**(±0.11)	.55(±0.08)	**.43**(±0.03)	.61(±0.07)	.011	1
$svm_{rwm} + sp + pwc_{rwm}$	.62(±0.1)	.88(±0.02)	.54(±0.08)	**.73**(±0.02)	.75(±0.09)	.62(±0.09)	.50(±0.13)	**.63**(±0.05)	.41(±0.05)	**.62**(±0.07)	.016	2
30 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.76**(±0.04)	**.90**(±0.01)	**.76**(±0.04)	**.74**(±0.03)	**.79**(±0.04)	**.75**(±0.02)	.74(±0.04)	**.69**(±0.03)	.51(±0.02)	**.69**(±0.05)	.008	1
$svm_{rbf} + mp + pwc_{rbf}$	.74(±0.07)	**.90**(±0.01)	.76(±0.05)	.73(±0.04)	.79(±0.06)	.72(±0.02)	**.75**(±0.05)	.67(±0.04)	**.58**(±0.04)	.65(±0.04)	.012	2
$pwc_{gmm} + sp$	**.69**(±0.07)	**.90**(±0.02)	**.58**(±0.05)	**.70**(±0.03)	**.75**(±0.08)	**.63**(±0.09)	**.59**(±0.08)	.57(±0.07)	.46(±0.03)	**.66**(±0.05)	.01	1
$svm_{gmm} + sp + pwc_{gmm}$	.66(±0.06)	**.90**(±0.02)	.57(±0.06)	.69(±0.03)	.72(±0.09)	.61(±0.09)	.49(±0.15)	**.65**(±0.02)	**.48**(±0.04)	.63(±0.05)	.023	2
pwc_{rwm}+sp	**.65**(±0.08)	**.89**(±0.02)	**.56**(±0.07)	.70(±0.04)	**.78**(±0.07)	**.70**(±0.06)	**.57**(±0.09)	.55(±0.08)	**.46**(±0.03)	.65(±0.07)	.012	1
$svm_{rwm} + sp + pwc_{rwm}$	.63(±0.08)	**.89**(±0.02)	.54(±0.08)	**.73**(±0.03)	.77(±0.07)	.63(±0.09)	.53(±0.13)	**.63**(±0.05)	.40(±0.05)	**.66**(±0.06)	.022	2

Table 3. Results with the McPAL strategy with PWC and SVM consumer classifiers with different kernels

10 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.64**(±0.08)	**.81**(±0.09)	.66(±0.07)	.69(±0.08)	**.77**(±0.06)	**.74**(±0.03)	.62(±0.09)	.55(±0.06)	.41(±0.03)	**.61**(±0.07)	.021	1
$pwc_{gmm} + mp$	.63(±0.09)	.79(±0.1)	.60(±0.07)	.68(±0.07)	.75(±0.07)	.73(±0.04)	.58(±0.08)	.57(±0.07)	.44(±0.04)	**.61**(±0.07)	.033	2
$pwc_{rwm} + mp$	.63(±0.09)	.79(±0.1)	.59(±0.08)	.67(±0.08)	.74(±0.09)	.73(±0.03)	.55(±0.09)	.56(±0.08)	.40(±0.03)	**.61**(±0.07)	.044	4
$svm_{rbf} + mp$	.60(±0.08)	.80(±0.09)	**.68**(±0.07)	.69(±0.07)	.62(±0.09)	.71(±0.04)	**.63**(±0.09)	.56(±0.06)	.43(±0.04)	.57(±0.12)	.042	3
$svm_{gmm} + mp$	.60(±0.08)	.79(±0.1)	.61(±0.09)	.66(±0.07)	.69(±0.08)	.69(±0.04)	.53(±0.13)	**.65**(±0.0)	**.47**(±0.05)	.54(±0.12)	.048	5
$svm_{rwm} + mp$	.62(±0.11)	.78(±0.1)	.61(±0.09)	**.71**(±0.07)	.69(±0.09)	.71(±0.04)	.52(±0.13)	.63(±0.05)	.38(±0.05)	.57(±0.09)	.049	6
20 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.68**(±0.08)	**.88**(±0.05)	.72(±0.05)	**.73**(±0.05)	**.75**(±0.06)	**.74**(±0.03)	**.72**(±0.04)	.60(±0.05)	.46(±0.03)	**.67**(±0.06)	.017	1
$pwc_{gmm} + mp$	.68(±0.09)	.87(±0.05)	.68(±0.07)	.70(±0.05)	.73(±0.07)	**.74**(±0.03)	.66(±0.06)	.61(±0.08)	.50(±0.04)	.66(±0.06)	.029	3
$pwc_{rwm} + mp$	.68(±0.09)	.87(±0.05)	.65(±0.07)	.70(±0.06)	.72(±0.09)	.73(±0.02)	.64(±0.07)	.63(±0.05)	.45(±0.03)	.66(±0.06)	.039	5
$svm_{rbf} + mp$	.66(±0.09)	.87(±0.04)	**.74**(±0.05)	.72(±0.04)	.66(±0.1)	.72(±0.03)	.72(±0.06)	.60(±0.05)	.51(±0.04)	.64(±0.07)	.028	2
$svm_{gmm} + mp$	.65(±0.1)	.87(±0.04)	.69(±0.08)	.69(±0.06)	.71(±0.08)	.71(±0.04)	.63(±0.09)	**.65**(±0.0)	**.56**(±0.04)	.61(±0.07)	.035	4
$svm_{rwm} + mp$	.68(±0.09)	.87(±0.04)	.65(±0.06)	.71(±0.04)	.71(±0.09)	.72(±0.03)	.62(±0.09)	**.65**(±0.0)	.38(±0.05)	.65(±0.06)	.048	6
30 sampled labels	australian	blobs	glass	haberman	heart	phoneme	qsar	steel	vehicle	vertebral	diff	rank
$pwc_{rbf} + mp$	**.76**(±0.04)	**.90**(±0.01)	**.76**(±0.04)	**.74**(±0.03)	**.79**(±0.04)	**.75**(±0.02)	.74(±0.04)	**.69**(±0.03)	.51(±0.02)	**.69**(±0.05)	.010	1
$pwc_{gmm} + mp$	.75(±0.04)	**.90**(±0.01)	.74(±0.06)	.71(±0.05)	.75(±0.06)	**.75**(±0.02)	.67(±0.06)	.69(±0.04)	.55(±0.03)	.68(±0.05)	.024	3
$pwc_{rwm} + mp$	.75(±0.04)	.89(±0.01)	.69(±0.07)	.70(±0.05)	.75(±0.08)	.74(±0.02)	.66(±0.07)	.69(±0.04)	.49(±0.03)	.68(±0.04)	.039	4
$svm_{rbf} + mp$	.74(±0.07)	**.90**(±0.01)	.76(±0.05)	.73(±0.04)	**.79**(±0.06)	.72(±0.02)	**.75**(±0.05)	.67(±0.04)	.58(±0.04)	.65(±0.04)	.014	2
$svm_{gmm} + mp$	.72(±0.07)	**.90**(±0.01)	.74(±0.07)	.70(±0.05)	.75(±0.07)	.71(±0.05)	.63(±0.08)	.65(±0.0)	**.6**(±0.04)	.63(±0.05)	.040	5
$svm_{rwm} + mp$	.74(±0.06)	.89(±0.02)	.64(±0.07)	.71(±0.04)	.75(±0.08)	.72(±0.03)	.61(±0.09)	.65(±0.0)	.39(±0.05)	.67(±0.05)	.066	6

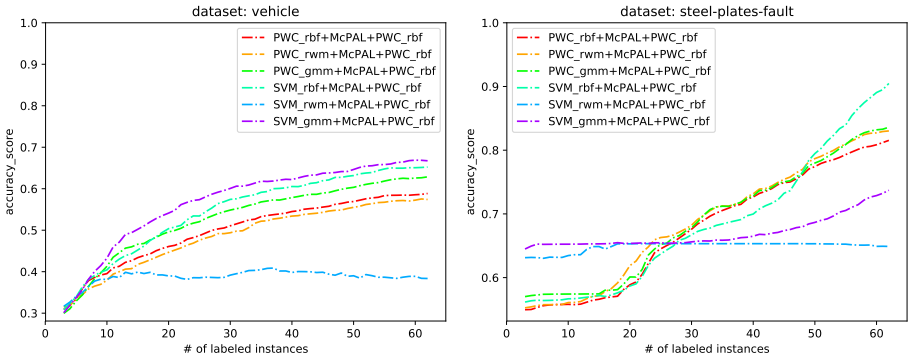


Fig. 2. Learning curves of McPAL learners using different consumer classifiers on the vehicle and steel data sets

6 Conclusion

The experiments explored two possible approaches to incorporate the information of a GMM into the McPAL method. The first approach, using two GMM-based kernel functions in the instance selection process, has shown to not provide an advantage regarding the performance compared to the original method. In total, the original McPAL selection strategy with pwc_{rbf} both as selector and consumer classifier, has shown to perform better than the StrucPAL learners, with random sampling performing better than both methods in few cases. Especially data sets like australian, glass, vehicle and vertebral proofed harder for the StrucPAL learners. One possible explanation for this is that the assumption of the GMM, i.e. that the subpopulations in the data representing the different classes fit a multivariate Gaussian distribution, does not hold in these cases.
The second approach, using the GMM-based kernel functions in the consumer classifiers of a foreign-selection scenario, showed potential gains regarding classification accuracy. Using a svm_{gmm} as consumer classifier for the original McPAL learner has shown to improve the classification performance on the steel and vehicle data sets while performing slightly worse on others. The fact, that learners using the StrucPAL method and PWC with the GMM or RWM kernel generally did not benefit from using an SVM with these kernels proves interesting. It appears that either the use of the same kernel function did not mitigate the adverse effect foreign-selection seems to entail in this case, or that the labeled set sampled by StrucPAL is simply less fit for classification with svm_{gmm} or svm_{rwm}.
Regarding the performance of the SVM classifiers used in the experiments, it has to be considered that the model selection procedure employed in the experiments is admittedly weak. Therefore, it is possible that the hyperparameters used in the experiments, not only for the SVMs but also the other classifiers, are suboptimal. Considering the restrictive nature of the AL setting regarding the availability of labeled data, this circumstance is acceptable, since using more data for model selection would be even more unrealistic in this setting.
It appears that the McPAL strategy already performs very well at selecting the most useful instances and including the information of the GMM does not add to this, in some cases even hindering a good selection. Based on these results it appears that work on the McPAL strategy in the future should focus on improving the method regarding other aspects, for example imbalanced data.
However, using other classifiers to exploit the labeled set sampled with the McPAL strategy has shown to be of possible gain, in order to improve the overall classification performance of the active learner. The use of SVMs as consumer classifiers showed to have potential, although determining fitting hyperparameters in the setting of active learning still poses a problem, that should be investigated further.

Bibliography

[1] C Bishop. Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn. *Springer, New York*, 2007.

[2] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *ICML*, volume 3, pages 59–66, 2003.

[3] Nicolas Cebron and Michael R Berthold. Active learning for object classification: from exploration to exploitation. *Data Mining and Knowledge Discovery*, 18(2):283–299, 2009.

[4] Gang Chen, Tian-jiang Wang, Li-yu Gong, and Perfecto Herrera. Multi-class support vector machine active learning for music annotation. *International Journal of Innovative Computing, Information and Control*, 6(3):921–930, 2010.

[5] Charlie K Dagli, Shyamsundar Rajaram, and Thomas S Huang. Utilizing information theoretic diversity for svm active learn. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 506–511. IEEE, 2006.

[6] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. ACM, 2008.

[7] Dezhi Hong, Hongning Wang, and Kamin Whitehouse. Clustering-based active learning on sensor type classification in buildings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 363–372. ACM, 2015.

[8] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2372–2379. IEEE, 2009.

[9] Ajay J Joshi, Fatih Porikli, and Nikolaos P Papanikolopoulos. Scalable active learning for multiclass image classification. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2259–2273, 2012.

[10] Daniel Kottke, Georg Krempl, Dominik Lang, Johannes Teschner, and Myra Spiliopoulou. *Multi-Class Probabilistic Active Learning*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 586 – 594. IOS Press, 2016.

[11] Jan Kremer, Kim Steenstrup Pedersen, and Christian Igel. Active learning with support vector machines. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(4):313–326, 2014.

[12] Dominik Lang, Daniel Kottke, Georg Krempl, and Myra Spiliopoulou. Investigating exploratory capabilities of uncertainty sampling using svms in active learning. In *Active Learning: Applications, Foundations and Emerging Trends*, 2016.

[13] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[14] M. Lichman. UCI machine learning repository, 2013.

[15] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 79. ACM, 2004.

[16] Tobias Reitmaier and Bernhard Sick. Active classifier training with the 3ds strategy. In *Computational Intelligence and Data Mining (CIDM), 2011 IEEE Symposium on*, pages 88–95. IEEE, 2011.

[17] Tobias Reitmaier and Bernhard Sick. The responsibility weighted mahalanobis kernel for semi-supervised training of support vector machines for classification. *Information Sciences*, 323:179–198, 2015.

[18] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[19] Katrin Tomanek and Katharina Morik. Inspecting sample reusability for active learning. In Isabelle Guyon, Gavin C. Cawley, Gideon Dror, Vincent Lemaire, and Alexander R. Statnikov, editors, *AISTATS workshop on Active Learning and Experimental Design*, volume 16 of *JMLR Proceedings*, pages 169–181. JMLR.org, 2011.

[20] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[21] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.

Simulation of Annotators for Active Learning: Uncertain Oracles

Adrian Calma and Bernhard Sick

Intelligent Embedded Systems
University of Kassel, Germany
{adrian.calma,bsick}@uni-kassel.de

Abstract. In real-world applications the information for previously unknown categories (labels) may come from various sources, often but not always humans. Therefore, a new problem arises: The labels are subject to uncertainty. For example, the performance of human annotators depends on many factors: e.g., expertise/experience, concentration/distraction, fatigue level, etc. Furthermore, some samples are difficult for both experts and machines to label (e.g., samples near the decision boundary). Thus, one question arises: How can one make use of annotators that can be erroneous (uncertain oracles)? A first step towards answering this question is to create experiments with humans, which involves a high time and money effort. This article addresses the following challenge: How can the expertise of erroneous human annotators be simulated? First, we discuss situations in which humans are prone to error. Second, we present methods for conducting active learning experiments with simulated uncertain oracles that possess various degrees of expertise (e.g., local/global or class/region dependent).

Keywords: Active Learning, Uncertain Oracles

1 Introduction

Consider the following problem: we have access to a large set of unlabeled images and we have the possibility to buy labels for any data point, Our first goal is to train a classifier with the highest possible accuracy. A possible approach is to label all the images and then train the classifier on the labeled data set. Now, suppose we have a limited budget, which doesn't allow us to label the all images. Our second goal is to **keep the costs to a minimum**. Thus, we need a strategy to determine which images should be labeled. A naive strategy would be to select the images at random. But, we can do better than that, if we make use of a selection strategy that selects the most *informative* images. Precisely at this point, active learning (AL) comes in, more specifically pool-based active learning (PAL). The learning cycle is presented in Figure 1: there is a **large set of unlabeled data** and our **goal** is **to train a model** (e.g., a classifier). Thus, we need to select **the most informative** data points based on a **selection strategy** and present them to an annotator (e.g., a domain expert), generally

called **oracle**, for labeling. The labeled samples are added to the training set (the set with labeled data), the **classifier is updated** and, depending on the chosen **stopping criteria** (e.g., is there still money in our budget?), we continue to ask for more labels or not.

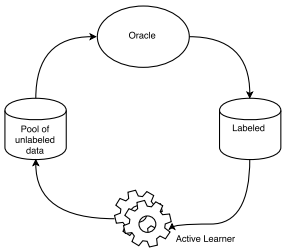


Fig. 1. Pool-based active learning cycle.

At this point we can ask ourselves: Are the labels provided by the human annotators correct? Probably not, as we can assume that humans are prone to error (Section 2). Thus, a new question arise: **How can we deal with uncertainty regarding the labels?** A first step towards answering the previous question is to develop techniques for simulating human experts prone to error. As we assume that they are unsure regarding the classification decision, we call these annotators *uncertain oracles*. Thus, this article focuses on presenting:

- cases in which the uncertain oracle misclassifies data (see Section 3) and
- techniques for simulating uncertain oracles in AL (see Section 4).

In the remainder of this article, we first present the possible causes for erroneous labels and explain what we mean by the term "uncertainty" (Section 2). Then, we present and categorize various types of expertise (Section 3). In Section 4, we introduce possible approaches for simulating error prone oracles. Then, related work will be summarized in Section 5. Finally, Section 6 concludes the article.

2 Motivation – The Problem

By now, we assumed that the answers provided by the oracles are always right. But, it is obvious that they are not always right. On the one hand, the **performance** of human annotators (human oracles) depends on multiple factors, such as: **expertise**, **experience**, level of **concentration**, level of **interest**, or level of **fatigue** [1]. On the other hand, the labels may come from simulations or test stands. Once again, it is justifiable to assume that due to imperfect simulations, sensor noise, or transmission errors, the labels are subject to uncertainty.

Depending on the difficulty of the labeling task, the oracles might be right in case of "easy" classification problems. The more difficult a classification task

is, the likelier it is that the oracle has a higher degree of doubt (i.e., is more uncertain) about its answer. Thus, the label uncertainty depends on the difficulty of the classification task. That is, the number of steps an annotator has to perform for determining the right class, the designated time, and the risk involved by misclassification. This factors come in addition to the previously presented sources of uncertainty, such as required knowledge for problem understanding, experience regarding similar classification problems or labeling tasks, concentration, or tiredness.

What do we mean by "uncertainty"? When humans are asked to provide information about an actual situation, the confidence regarding the given answer depends on diverse factors, such as the difficulty/complexity to assess that information, previous experience, or knowledge. Certainly, there are circumstances when we cannot state our answer with absolute confidence. Thus, we tend to add additional information about the quality of our answer, i.e., to quantify and qualify our confidence [2].

On these grounds, we cannot assume that the oracles are omniscient, but we have to soften the assumption of omniscience: An oracle may be wrong. In this context, the "uncertainty" is the degree of confidence for given label. Consequently we ask ourself, how can we make use of the uncertain oracles, especially, how can we exploit the oracle's firm knowledge?

3 Human Expertise

When an expert has worked for a long period of time on a classification task, he posses more "experience". That is, he has seen and labeled more data than an oracle that just started to work on the labeling task. Therefore, such an oracle possesses *global expertise* about the classification problem. On the other hand, depending on how difficult the classification problem is or on the degree of *expertise* and experience, the oracle may bear only limited knowledge about the learning task, i.e., *local expertise*.

At this point, we assume that the expertise of an oracle (its degree of uncertainty) is time invariant.

3.1 Global Expertise

The annotators have a global expertise in the sense that their knowledge is not limited to a certain region of the input space or to a specific class. They "know" the problem in all its aspects. Still, they may possess different levels of expertise. Moreover, samples exist that are hard to label for both the learning system as well as for the oracles. For example, samples that lie near the decision boundary of a classifier are good examples for data points that might be difficult to label by the oracle and the active learner.

From a practical point of view, we may ask the oracles to provide additional information when they provide labels for samples. This is required for assessing their certainty, or rather their *uncertainty* regarding the provided answer. Such additional information may include asking for [1]:

1. a degree of confidence for one class,
2. membership probabilities for each class,
3. a difficulty estimate, or
4. a relative difficulty estimate for two data points.

In the first case, a sample is presented to the oracle, for example an image. The oracle is asked to provide a class label for the sample and to estimate his degree of confidence. Further help regarding the degree of confidence may be provided: e.g. a a graphical control element with which the oracle sets a certainty value by moving an indicator on a predefined scale (i.e., a slider). Thus, a possible answer may look like *"I select class «cat» and I rate my certainty 3 on a scale from 1 to 4, where 4 is the highest score"*.

Another possibility is to ask the oracle to provide class estimates for each of the possible categories. Given an 3-class classification problem, an answer may be *"The self estimated probability for the first class is 0%, for the second class 30%, and for the third class 70%"*.

The last two cases address cases where the oracle has to estimate how difficult it was for him to label a specific data point. Possible answer may look like *"I choose class «cat» and it was hard for me to determine it"*, if it was asked to label only one sample, or *"It was* easier *for me to label the image depicting a «cat» than the one showing a «liger»"*, if asked to label to images simultaneously.

3.2 Local Expertise

The oracles possess a *local expertise* in the sense that they do not have enough "experience", they can only recognize specific classes, they are more reliable for specific regions of the input space, or for certain features. That is, the human annotators are experts for:

1. different classes,
2. different regions of the input space, or
3. different dimensions of the input space (i.e., features, attributes).

We assume that, in some applications, the oracles have not only diverse degrees of experience and expertise, but they have various levels of proficiency for different parts of the classification problem. For example, the oracle may be more confident and adept in detecting some certain classes. The quality of the given answers and his confidence may vary over the regions of the input space or it may depend on the considered features (dimension of the input space).

It is not required to change the way the active learner queries new labels. The query approaches described in Section 3.1 can be adopted for this case too.

3.3 Disparate Features

Up to this point we assumed that the oracle and the active learner are considering the same features for solving the classification problem. But, this is not always the case. For example, complex processes happen in our brains when we examine

an image. It is hard to say which "features" we consider when trying to recognize or evaluate the content of that specific image. Still the active learner "views" the same image, but it may consider additional features such as histograms or apply filters (e.g. anisotropic diffusion [3] or median [4] filters) or transformations (e.g. Fourier [5] or Hough [6] transform) on the image. Obviously, we can provide these additional information to the oracles, but the active learner might not have access to all features that were "extracted" by the oracles.

Once again, the answers expected from oracles can be implied from Section 3.1. But, you may ask yourself why we do not ask the oracle for additional information regarding the features that it considers for its decision. As we focus on classification tasks, we do not consider it in this work, but it is definitely an interesting research topic, commonly referred to as *active feature selection* [7].

4 Simulate Error Prone Annotators

A first step towards exploiting the knowledge of an uncertain oracle would be to analyze how the current AL paradigms perform in combination with multiple oracles. But, such experiments are costly both in terms of money and time. If we are able to successfully simulate uncertain oracles, then we can better investigate the performance of the selection strategies and of the classifiers without generating additional costs in this research phase. Moreover, based on the gathered knowledge from the investigation of current active learning techniques in a dedicated collaborative interactive learning (D-CIL) context, we can develop new ones, that take the uncertainty into consideration. That brings us to the following questions: how can we simulate error prone annotators (uncertain oracles)?

In the following, we will describe different approaches for simulating the uncertain oracles.

4.1 Omniscient Oracle

For the sake of completeness, we shortly describe how an omniscient oracle can be simulated and what we understand under *experience* in this context. Simulating this type of oracle is straight forward: It returns the true labels of the samples. That is, the labels are not manipulated in any way.

How can we simulate the *experience*? We define the experience as the number of samples the uncertain oracle has already seen and labeled. Thus, when we consider the complete data set for training a classifier (i.e., supervised learning) we can simulate an uncertain oracle with maximal global experience. Global, in the sense that the expertise is not limited to a region of the input space or to a specific class.

4.2 Uncertain Oracle with Global Expertise

At first, we concentrate on how to simulate uncertain oracles with global expertise and the same degree of experience. We assume that the labels near the decision

boundary of the classifier are hard to classify for both the human expert (human oracle) as well as for the classifier. Thus, we can simulate an uncertain oracle by randomly altering (changing) the classes of the samples lying near the decision boundary. A legit question may arise: What is the "right" decision boundary? We do not know, but we can estimate it. As one of the goals of active learningis to be as good as a learner trained in a supervised way, we can train a classifier in a supervised way (i.e., overall data set). The decision boundary resulted from this classifier trained can be used to determine the samples for which the labels are altered.

The next challenge is to simulate oracles that have different levels of experience. For example, the oracle may have just started labeling samples for this type of problem. Thus, they have only a labeled few samples and, of course, their experience is based on a small number of data. One possible way to simulate its "experience" is to reduce the number of samples on which the classifier is trained. As the classifier is used as a model of the experience, by reducing the number of samples we increase the level of uncertainty. By doing so, we simulate an oracle that has little experience. Depending on the reduction factor, uncertain oracles with different levels of experience can be simulated. Moreover, if we can split the data in such a way, that the training set of the classifier used to simulate the uncertain oracle is larger than the pool of unlabeled data. Thus, the data from which the uncertain oracles gathered their experience is larger than the data from which the active learner can select samples for labeling, resulting in a simulated oracle with a higher degree of expertise.

Another possibility to simulate uncertain oracles with different levels of experience is to alternate the parameter values of the classifiers. For example, we can simulate the expertise of an uncertain oracle with a classifier trained with default parameters. For a better expertise, we can imply heuristics (e.g., grid search) to find suitable parameters for the classifiers.

Furthermore, the expertise can be simulated by different types of classifiers. We can use generative or discriminative classifier for simulating the expertise of an expert.

Last but not least, we can add noise to the feature values. Of course, this is not always possible, as it depends on the type of feature (i.e., nominal, continuous, ordinal, etc.) and on the values range. By doing so, we can simulate uncertain oracles that have an experience built on similar samples.

In a nutshell, we can simulate oracles with global and various degrees of expertise by

- modifying (altering) the classes of the samples lying near the decision boundary,
- training different classifier types for various uncertain oracles, and
- training a classifier
  - on training sets of different size (more or less samples than in the pool of unlabeled data),
  - using different parametrization strategies and parameter sets, or
  - adding noise to the feature values (if possible and if it makes sense).

Additionally, any combination of the previous simulation can be implied. For example, if we want to simulate an oracle with little global expertise based on similar samples, we can reduce the training set of the classifier and add noise to the feature values.

4.3 Uncertain Oracle with Local Expertise

The expertise of an oracle can be restricted to a certain class or to a specific region of the input space. Thus, to simulate a better expertise with respect to one or more classes of our choice, we can change the labels of the samples belonging to the classes for which we would like to simulate a little (or no) expertise. It is also possible to exclude the samples belonging to one class, which translates to "the uncertain oracle has no expertise regarding this specific class". One possible approach is to train a generative classifier on these data. The resulting classifier estimates the processes that are supposed to generate the data, i. e. one process generates samples belonging to one class. That is, a process generates samples belonging to only one class. Therefore, we can artificially change the labels of the estimated processes, which results in an erroneousness classification of samples that were assumed to be generated by that process.

The expertise of the uncertain oracles may be restricted to a specific region of the input space. Depending on the feature values, the labeling quality can suffer. For example, an uncertain oracle is more accurate regarding samples that lie in regions of the input space, which have been previously seen or learned by the oracle. We propose two ways to simulate the local expertise: (1) by using various classifier types and (2) by deliberately altering the class affiliations of the samples lying in those regions.

By using different classifier types, the regions of the input space are modeled in different ways and, thus, the result of the classification may vary.

By modifying the classes of the samples lying in specific regions of the input space, the result of the classifier is modified. That is, for samples lying in these regions, the expertise of the uncertain oracle is diminished.

The difference between class based experience and region based experience is showed in Figure 2. Here, we have a region of the input space where two classes strongly overlap, *green* $\circ$'s and *blue* $+$. If we assume that a human expert has firm knowledge about class *green* $\circ$, then he will probably label the samples that belong to the green class correctly and the others not (higher error rate for *blue* $+$ and *red* $\triangle$). On the other had, assuming that the oracle correctly labels samples in a given region of the input space leads us to the conclusion that it labels correctly all the samples in the specified region. For example, the uncertain oracle has a region based expertise for samples having feature values $\in [-1.5, 1.5]$, will lead to correct class affiliation for samples lying in this region. In this concrete case, samples lying in the square defined by $(-1.5, -1.5)$ and $(1.5, 1.5)$ and belonging to either class are labeled correctly.

An overview of the introduced simulation methods is presented in Figure 3. The core of the simulation techniques is the assumption regarding which features are considered. The described simulation methods can be applied for both cases:

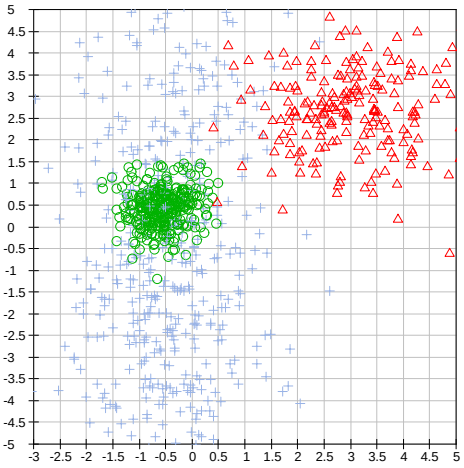


Fig. 2. Samples belonging to three classes (*green* ∘'s, *blue* +'s, and *red* △)'s depicted in the input space, whereby the processes generating samples belonging to *green* ∘'s and *blue* +'s strongly overlap.

when the uncertain oracle considers the same features as the active learner and when not.

4.4 Motivating Example: Generative Classifier based Simulation

One possible way to simulate the expertise of an uncertain oracle is by means of a generative classifier, e.g. a classifier based on mixture models, which is based on a probabilistic mixture modeling approach. That is, for a given D-dimensional input sample $\mathbf{x}'$ we can compute the posterior distribution $p(c|\mathbf{x}')$, i.e., the probabilities for class c membership given the input $\mathbf{x}'$. To minimize the risk of classification errors we then select the class with the highest posterior probability (cf. the principle of *winner-takes-all*). Thus, the "uncertainty" can be computed as $1 - p(c'|j)$, where $c' = \text{argmax}_c\, p(c|j)$. In case of other classifier types (e.g., Support Vector Machines), Platt scaling [8] can be used to transform the outputs into probability distributions.

5 Related Work

In [9], the authors simulate oracles with different types of accuracies: 10% of samples are incorrect, 20% unknown, and 70% uncertain knowledge. k-means clustering is implied in [10] to generate the concepts and to assign the oracles to different clusters, in order to simulate the experience (in this article called "knowledge sets"). Clustering is also used in [11], where some clusters represent regions for which the oracles give unsure as feedback. Virtual oracles for binary classification, with different labeling qualities, controlled by two parameters that

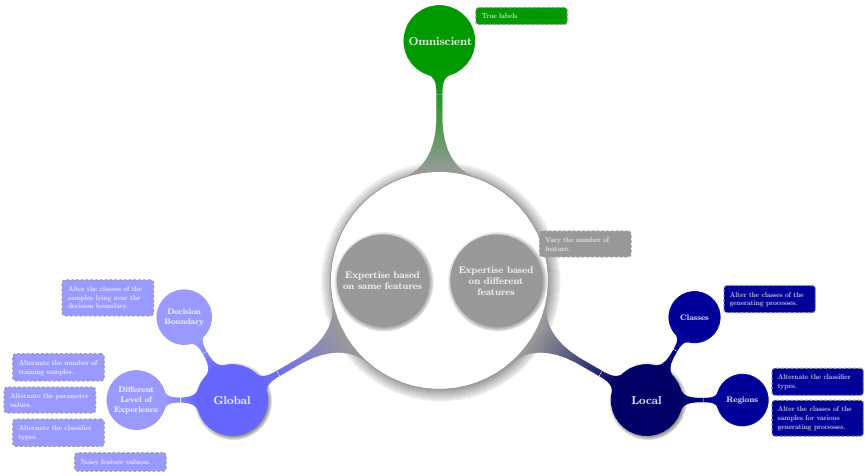


Fig. 3. Types of expertise and possible simulation practices.

represent the label accuracy regarding the two classes are presented in [12]. In [13], a uniform distribution is implied to simulate various behavior of the oracles. Randomly flipping labels with a specific probability [14] and ranges for the noise rate [15] are also applied to simulate uncertain oracles. A Gauss distribution [16] has also been use to simulate the expertise of oracles. But also multiple oracles have been simulated, where their label quality does not vary [17].

6 Conclusion

In this article, we addressed a challenge in the field of AL and, especially, in the field of D-CIL [1], where oracles might be wrong for various reasons. Thus, the queried labels are subject to uncertainty. The research regarding uncertain oracles is still in its infancy, so we proposed **simulation methods for uncertain oracles** in order to help the research go further. The simulation methods will help investigate the performance of the current AL techniques and understand their advantages and disadvantages. Moreover, new questions for future research arise: How can we exploit the uncertain oracles? Is it necessary to re-query labels for already labeled samples? How can we learn (model) the expertise of an uncertain oracle? How do we decide whether the uncertain oracle is erroneous or the process to be learned are nondeterministic? How do we decide whom to ask next?

References

1. Calma, A., Leimeister, J.M., Lukowicz, P., Oeste-Reiß, S., Reitmaier, T., Schmidt, A., Sick, B., Stumme, G., Zweig, K.A.: From active learning to dedicated collaborative interactive learning. In: International Conference on Architecture of Computing Systems, Nuremberg, Germany (2016) 1–8

2. Motro, A., Smets, P., eds.: Uncertainty Management in Information Systems – From Needs to Solutions. Springer US (1997)
3. Weickert, J.: Anisotropic Diffusion in Image Processing. B.G. Teubner Stuttgart (1998)
4. Zhu, Y., Huang, C.: An improved median filtering algorithm for image noise reduction. Physics Procedia **25** (2012) 609–616
5. Cochran, W., Cooley, J., Favin, D., Helms, H., Kaenel, R., Lang, W., Maling, G., Nelson, D., Rader, C., Welch, P.: What is the fast Fourier transform? Proceedings of the IEEE **55** (1967) 1664 – 1674
6. Nixon, M.S., Aguado, A.S.: Feature Extraction and Image Processing. Academic Press (2008)
7. Liua, H., Motoda, H., Yua, L.: A selective sampling approach to active feature selection. Artificial Intelligence **159** (2004) 49–74
8. Platt, J.C.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in Large Margin Classifiers **10** (1999) 61–74
9. Fang, M., Zhu, X.: Active learning with uncertain labeling knowledge. Pattern Recognition Letters **43** (2013) 98–108
10. Fang, M., Zhu, X., Li, B., Ding, W., Wu, X.: Self-Taught Active Learning from Crowds. In: 2012 IEEE 12th International Conference on Data Mining (ICDM), Brussels, Belgium (2012) 1–6
11. Zhong, J., Tang, K., Zhou, Z.H.: Active Learning from Crowds with Unsure Option. In: 24th International Conference on Artificial Intelligence, AAAI Press (2015) 1061–1067
12. Jing, Z., Xindong, W., S., S.V.: Active Learning With Imbalanced Multiple Noisy Labeling. IEEE Transactions on Cybernetics **45** (2015) 1081–1093
13. Kumar, A., Lease, M.: Modeling Annotator Accuracies for Supervised Learning. In: WSDM 2011 Workshop on Crowdsourcing for Search and Data Mining (CSDM 11), Hong Kong, China (2011) 19–22
14. Yan, Y., Rosales, R.: Active learning from multiple knowledge sources. In: 15th International Conference on Artificial Intelligence and Statistics (AISTATS). Volume XX., La Palma, Canary Islands (2012)
15. Du, J., Ling, C.X.: Active learning with human-like noisy oracle. In: IEEE 10th International Conference on Data Mining, Sydney, Australia (2010) 797–802
16. Zhao, L.: An Active Learning Approach for Jointly Estimating Worker Performance and Annotation Reliability with Crowdsourced Data. ArXiv (2014) 1–18
17. Shu, Z., Sheng, V.S., Li, J.: Learning from crowds with active learning and self-healing. Neural Computing and Applications (2017) 1–12

Users behavioural inference with Markovian decision process and active learning

Firas Jarboui[1,2], Vincent Rocchisani[2], and Wilfried Kirchenmann[2]

[1] ENSTA, France and ENIT, Tunisia
[2] ANEO, Boulogne Billancourt, France
{fjarboui,vrocchisani,wkirschenmann}@aneo.fr

1 Introduction

Studies on Massive Open Online Courses (MOOCs) users discuss the existence of typical profiles and their impact on the learning process of students. One of the concerns when creating a new MOOC is knowing how the users behave when going through the contents. We can identify either quantitative methods that allow you to infer hardly interpretable groups of similar behaviour[1] or hardly context-transposable qualitative methods[2]. Our ambition is to find an efficient way to identify the behavioural pattern of interest to a given human expert. Within the *#MOOCLive* project[3], we developed a mix-method to match the quantitative interpretation to the context needs.

2 Methodology

We tackled the following three problems in order to achieve our goal.

- The definition of a quantitative metric to compare behaviours
- The inference of qualitative sets behaviours from existing ones and test their reliability for describing the reality.
- The convergence between the quantitative-based clustering and the qualitative sets of behaviour to classify the users accordingly

In order to achieve our goal we define three main tasks. We start by quantifying the interest of users for the platform's activity. This will allow us to define a distance between their behaviours. Then, we iteratively make hypothetical class definitions and test how well they fit the existing population. This is repeated until both the classifier and the classes suggested by the process are deemed satisfactory. This process' breakdown is represented in *Fig.* 1.

1. **Quantitative modelling of the user:** We define the structure of the MOOC as a Markovian decision process framework. Let H be the history of actions the user performed on the platform. We define the gain function $\widehat{G}_H$ of a user as the expected value of a categorical soft-max probability distribution over SGF, a Sample from the space of all possible Gain Functions.

[3] *#MOOCLive* Virchow-Villermé ANR-15-IDFN-0003-04

The value associated to each element of this sample is the sum of rewards that the user's action would yield under the given gain function. This is thoroughly discussed in[3]. Each user is then characterised by the expected utility of each state with a discount factor γ.

$$\left\{ \begin{array}{l} U(G|H) = \sum_{a \in H} G(a) \\ \mathcal{P}(G|H) = \frac{e^{U(G|H)}}{\sum_{G' \in SGF} e^{U(G'|H)}} \end{array} \right\} \Rightarrow \widehat{G}_H = \sum_{G \in SGF} G \times \mathcal{P}(G|H)$$

2. **Qualitative class definition:** This step is purely human. The experts are asked to interfere and define the classes that will be used to build the quantitative classification. In this stage, the expert intervention is purely based on his *a priori*. If the expert's *a priori* is invalidated during the process, he will have to restart from here with an updated point of view.
3. **Fitting the classification:** To have well classified users a Gaussian kernel label propagation is used. This provides a probability distribution of membership to each pattern for each behaviour. An active learning process is used to iterate the propagation of the labels under the supervision of the human expert. After each fold, we sample the users randomly and test if the output probability distribution makes sense. The human expert either **agrees** with the results, **changes** them or tags them as **unsure**.
   If the rate of changed results is high, we continue the active learning loop. As a result, the rate of bad labels will decay.
   Once the classifier stabilizes, we consider the rate of behaviours that the expert tagged as unsure. *If this exceeds a threshold*, we roll back to the second step to challenge the a priori class definitions.
   If the unsure tags rate is low enough, we can safely assume that the two models converged with respect to the expert.

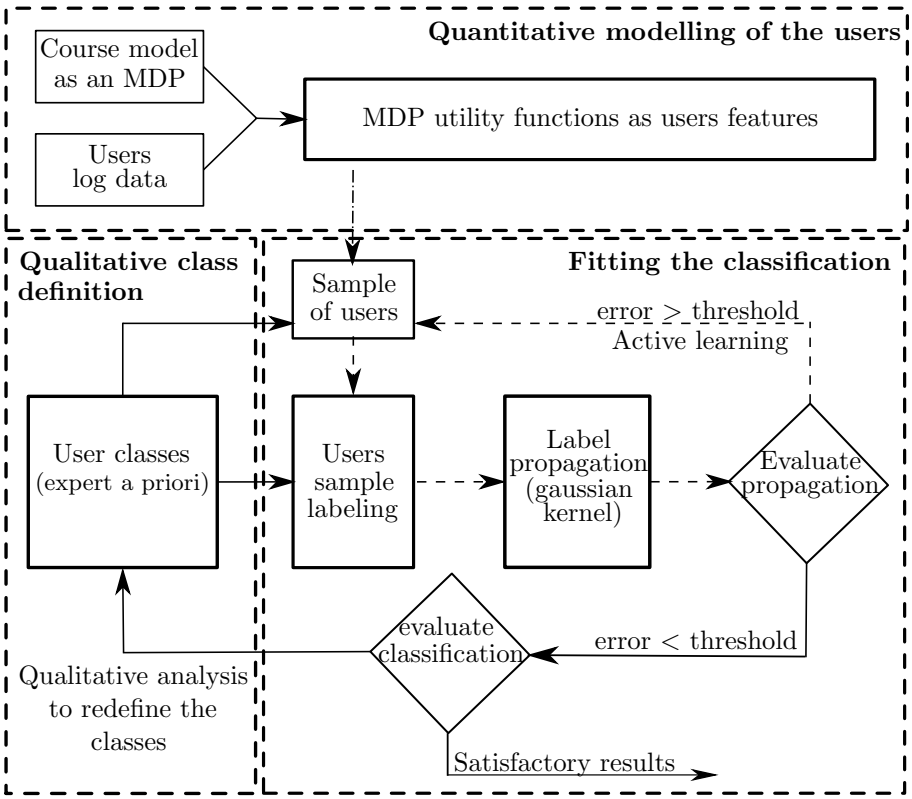
We applied this methodology on a MOOC[4] with a sociologist. We started with an *a priori* of three user profiles. Up to this date, after three iteration of the methodology, we were able to identify seven profiles that fulfil the context needs and to classify the users accordingly.

3 Conclusion

Our method assists a human expert to find the optimal information about the studied population. Although this work is still in progress and only tested on MOOC log data, it should be applicable on other log data streams of information. Future tests will involve marketing related data. We are currently investigating the efficiency of this method as well as the best techniques to use for each step. This is part of a preliminary work for a thesis.

[4] https://www.fun-mooc.fr/courses/VirchowVillerme/06005/session01/about

Fig. 1. Users behavioural inference process



References

1. Chase Geigle and Cheng Xiang Zhai: Modelling MOOC Student Behaviour With Two-Layer Hidden Markov Models. Learning at Scale (2017)
2. Paula de Barba Carleton Corin, Linda Corrin and Gregor Kennedy: Visualizing patterns of student engagement and performance in moocs. (2014)
3. Constantin A. Rothkopf and Christos Dimitrakakis: Preference Elicitation and Inverse Reinforcement Learning. cornell university library (2011)

Multi-Arm Active Transfer Learning for Telugu Sentiment Analysis

Subba Reddy Oota[1], Vijaysaradhi Indurthi[1], Mounika Marreddy[2]
Sandeep Sricharan Mukku[1], and Radhika Mamidi[1]

[1] International Institute of Information Technology, Hyderabad,
[2] Quadratyx, Hyderabad.
ootа.subba@students.iiit.ac.in,vijaya.saradhi@research.iiit.ac.in
mounika0559@gmail.com,sandeep.mukku@research.iiit.ac.in
radhika.mamidi@iiit.ac.in

Abstract. Transfer learning algorithms can be used when sufficient amount of training data is available in the source domain and limited training data is available in the target domain. The transfer of knowledge from one domain to another requires similarity between two domains. In many resource-poor languages, it is rare to find labeled training data in both the source and target domains. Active learning algorithms, which query more labels from an oracle, can be used effectively in training the source domain when an oracle is available in the source domain but not available in the target domain. Active learning strategies are subjective as they are designed by humans. It can be time consuming to design a strategy and it can vary from one human to other. To tackle all these problems, we design a learning algorithm that connects transfer learning and active learning with the well-known multi-armed bandit problem by querying the most valuable information from the source domain.
The advantage of our method is that we get the best active query selection using active learning with multi arm and distribution matching between two domains in conjunction with transfer learning. The effectiveness of the proposed method is validated by running experiments on three Telugu language domain-specific datasets for sentiment analysis.

Keywords: Active Learning, Transfer Learning, Multi-Arm Bandit

1 Introduction

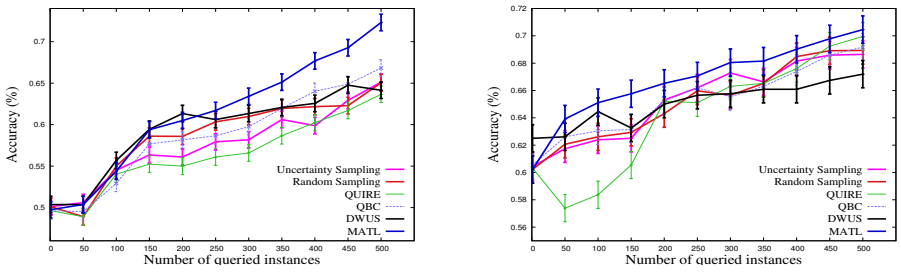
People comment on online reviews and blog posts in social media about trending activities in their regional languages. There are many tools, resources and corpora available to analyze these activities for English language. However, not many tools and resources are available to analyze these activities in resource poor languages like Telugu. With the dearth of sufficient annotated sentiment data in the Telugu language, we need to increase the existing available labeled datasets in different domains. However, annotating abundant unlabeled data manually is very time-consuming, cost-ineffective, and resource-intensive.

To address the above problems, we propose a Multi-Arm Active Transfer Learning (MATL) algorithm, which involves transfer learning [1] and a combination of query selection strategies in active learning [3]. One of the prerequisites

for transfer learning is that the source and target domains should be closely related. We use Maximum Mean Discrepancy (MMD) [2] as a measure to find the closeness between two distributions of the source and target domains. In this paper, we experiment with sentiment analysis of Telugu language domain specific datasets: Movies, Political and Sports[1]. By considering each domain as the source or target domain, we have a total of 6 domain pairs: M-P, M-S, P-M, P-S, S-M, S-P. Figure 1 shows two domain pair results. We evaluate the accuracy with three different classification techniques viz., support vector machines (SVM), extreme gradient boosting (XGBoost), gradient boosted trees (GBT), and meta learning of all these approaches and record the accuracy.

2 Approach & Results

In Multi-Arm active transfer learning approach, it takes both source domain: S = {unlabeled data instances (S_U), labeled data instances (S_L)}, and target domain: T = {unlabeled data instances (T_U), labeled data instances (T_L), test data instances (T_T) (used for measuring classification accuracy at each iteration)}, iterations (n) as an input. A decision making model is built along with this approach to predict the posterior probability for each instance of S_U. After calculating the sampling query distribution $\phi(S(n))$, based on multi-arm bandit approach a best sample instance $x_{i_n} \in S$ is selected for querying. If $x_{i_n} \in S_U$, then this selected sample instance (x_{i_n}) is labeled with an oracle/labeler as y_{i_n} and added to S_L. Now the classifier (C_n) is trained on the total set {updated S_L,T_L}. Using MMD [2], the distance between two distributions is calculated. This process is repeated until reached query budget. The classification model C_n is tested on target test data T_T to measure the accuracy. The reward ($r_n(a_k(n))$) and observation($o_n(a_k(n))$) is updated by comparing the label y_{i_n} given by the oracle/labeler with the classifier ($C_n(x_{i_n})$).



(a) P-S **(b) S-P**

Fig. 1. Performance comparison on Sentiment Analysis

References

1. Gong, B.: Discriminatively learning domain-invariant features for unsupervised domain adaptation. (2013)
2. Gretton, A., Smola, A.J.: A kernel method for the two-sample-problem (2007)
3. Settles, B.: Active learning literature survey. Tech. rep. (2010)

[1] `https://github.com/subbareddy248/Datasets/tree/master`

Probabilistic Expert Knowledge Elicitation of Feature Relevances in Sparse Linear Regression

Pedram Daee*, Tomi Peltola* Marta Soare*, and Samuel Kaski

Helsinki Institute for Information Technology HIIT and
Department of Computer Science, Aalto University, Finland,
firstname.lastname@aalto.fi
*Authors contributed equally.

1 Introduction

In this extended abstract[1], we consider the "small n, large p" prediction problem, where the number of available samples n is much smaller compared to the number of covariates p. This challenging setting is common for multiple applications, such as precision medicine, where obtaining additional samples can be extremely costly or even impossible. Extensive research effort has recently been dedicated to finding principled solutions for accurate prediction. However, a valuable source of additional information, domain experts, has not yet been efficiently exploited.

We propose to integrate expert knowledge as an additional source of information in high-dimensional sparse linear regression. We assume that the expert has knowledge on the relevance of the features in the regression and formulate the knowledge elicitation as a sequential probabilistic inference process with the aim of improving predictions. We introduce a strategy that uses Bayesian experimental design [2] to sequentially identify the most informative features on which to query the expert knowledge. By interactively eliciting and incorporating expert knowledge, our approach fits into the interactive learning literature [1, 8]. The ultimate goal is to make the interaction as effortless as possible for the expert. This is achieved by identifying the most informative features on which to query expert feedback and asking about them first.

2 Method

We introduce a probabilistic model that subsumes both a sparse regression model which predicts external targets, and a model for encoding expert knowledge. We then present a method to query expert knowledge sequentially (one feature at a time), with the aim of getting fast improvement in the predictive accuracy of the regression with a small number of queries.

For the regression, a Gaussian observation model with a spike-and-slab sparsity-inducing prior [5] on the regression coefficients is used: $\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{X}\boldsymbol{w}, \sigma^2 \mathbf{I})$, $w_j \sim \gamma_j \mathrm{N}(0, \psi^2) + (1 - \gamma_j)\delta_0; \gamma_j \sim \mathrm{Bernoulli}(\rho), j = 1, \ldots, p$, where $\boldsymbol{y} \in \mathbb{R}^n$ are

[1] This extended abstract is adapted from [3].

the output values and $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ the matrix of covariate values. The regression coefficients are denoted by $w_1, \ldots, w_p$, and σ^2 is the residual variance. The γ_j indicate inclusion ($\gamma_j = 1$) or exclusion ($\gamma_j = 0$) of the covariates in the regression (δ_0 is a point mass at zero). The prior expected sparsity is controlled by ρ. The expert knowledge on the relevance of the features for the regression is encoded by a feedback model: $f_j \sim \gamma_j \,\mathrm{Bernoulli}(\pi) + (1 - \gamma_j)\,\mathrm{Bernoulli}(1 - \pi)$, where $f_j = 1$ indicates that feature j is *relevant* and $f_j = 0$ *not-relevant*, and π is the probability that the expert feedback is correct relative to the state of the covariate inclusion indicator γ_j.

As the number of covariates p can be large, we assume that it is infeasible, or at least unnecessarily burdensome, to ask the expert about each feature. Instead, we aim to ask first about the features that are estimated to be the most informative given the (small) training data, and frame this problem as a Bayesian experimental design task [2, 9]. We prioritize features based on their expected information gain for the predictive distribution of the regression. As the expert is queried for the feedbacks sequentially, the posterior distribution of the model and the prioritization are recomputed after each feedback in order to use the latest knowledge. At iteration t for feature j, the expected information gain is

$$\mathrm{E}_{p(\tilde{f}_j|\mathcal{D}_t)}\left[\sum_i \mathrm{KL}[p(\tilde{y}|\mathcal{D}_t, \boldsymbol{x}_i, \tilde{f}_j) \parallel p(\tilde{y}|\mathcal{D}_t, \boldsymbol{x}_i)]\right],$$

where $\mathcal{D}_t = \{(y_i, x_i) : i = 1, \ldots, n\} \cup \{f_{j_1}, \ldots, f_{j_{t-1}}\}$ denotes the training data together with the feedback that has been given at previous iterations and $p(\tilde{f}_j|\mathcal{D}_t)$ is the posterior predictive distribution of the feedback for the jth feature. The summation over i goes over the training dataset. This query scheme goes beyond pure prior elicitation [4, 6, 7] as the training data is used to facilitate an efficient expert knowledge elicitation. This is a crucial aspect that enables the elicitation in high-dimensional regression.

3 Discussion

The proposed method was tested in several "small n,large p" scenarios on synthetic and real data with simulated and real users [3]. The results confirm that improved prediction accuracy is already possible with a small number of user interactions, for the task of predicting product ratings based on the relevance of some of the words used in textual reviews. Our method can naturally be used on many other applications where expert feedback is needed, its main advantage being that it efficiently reduces the burden on the expert by asking first the most informative queries. However, the amount of improvement in different applications depends on the type of feedback requested, and on willingness and confidence of experts to provide the feedback. In addition, appropriate interface and visualization techniques are also required for a complete and effective interactive elicitation. These considerations are left for future work.

Acknowledgements This work was financially supported by the Academy of Finland (Finnish Center of Excellence in Computational Inference Research COIN; grants 295503, 294238, 292334, and 284642), Re:Know funded by TEKES, and MindSee (FP7–ICT; Grant Agreement no 611570).

References

1. Amershi, S.: Designing for Effective End-User Interaction with Machine Learning. Ph.D. thesis, University of Washington (2012)
2. Chaloner, K., Verdinelli, I.: Bayesian experimental design: A review. Statistical Science 10(3), 273–304 (08 1995)
3. Daee, P., Peltola, T., Soare, M., Kaski, S.: Knowledge elicitation via sequential probabilistic inference for high-dimensional prediction. Machine Learning (Jul 2017), https://doi.org/10.1007/s10994-017-5651-7
4. Garthwaite, P.H., Dickey, J.M.: Quantifying expert opinion in linear regression problems. Journal of the Royal Statistical Society. Series B (Methodological) pp. 462–474 (1988)
5. George, E.I., McCulloch, R.E.: Variable selection via Gibbs sampling. Journal of the American Statistical Association 88(423), 881–889 (1993)
6. Kadane, J.B., Dickey, J.M., Winkler, R.L., Smith, W.S., Peters, S.C.: Interactive elicitation of opinion for a normal linear model. Journal of the American Statistical Association 75(372), 845–854 (1980)
7. O'Hagan, A., Buck, C.E., Daneshkhah, A., Eiser, J.R., Garthwaite, P.H., Jenkinson, D.J., Oakley, J.E., Rakow, T.: Uncertain Judgements. Eliciting Experts' Probabilisties. Wiley, Chichester, England (2006)
8. Porter, R., Theiler, J., Hush, D.: Interactive machine learning in data exploitation. Computing in Science & Engineering 15(5), 12–20 (2013)
9. Seeger, M.W.: Bayesian inference and optimal design for the sparse linear model. Journal of Machine Learning Research 9, 759–813 (2008)