A New Vision of Collaborative Active Learning

Adrian Calma[a], Tobias Reitmaier[a], Bernhard Sick[a], Paul Lukowicz[b], Mark Embrechts[c]

[a]*Intelligent Embedded Systems Lab, University of Kassel, Kassel, Germany*
(e-mail: {adrian.calma,tobias.reitmaier,bsick}@uni-kassel.de)
[b]*Embedded Intelligence, German Research Center for Artificial Intelligence, University of Kaiserslautern, Kaiserslautern, Germany*
(e-mail: paul.lukowicz@dfki.de)
[c]*Rensselaer Polytechnic Institute, Troy, New York, USA*
(e-mail: embrem@rpi.edu)

Abstract

Active learning (AL) is a paradigm where an active learner has to train a model (e.g., a classifier) which is in principle trained in a supervised way. In contrast to supervised learning, AL has to be done by means of a data set where a low fraction of samples is labeled or even with an initially unlabeled set of samples. To obtain labels for the unlabeled samples, the active learner has to ask an oracle (e.g., a human expert) for labels. In most cases, the goal is to maximize some metric or task performance and to minimize the number of queries at the same time. In this article, we first briefly discuss the state-of-the-art and own, preliminary work in the field of AL. Then, we propose the concept of collaborative active learning (CAL). With CAL, we will overcome some of the harsh limitations of current AL. In particular, we envision scenarios where the expert or the gold standard may be wrong for various reasons. There also might be several or even many experts with different expertise, the experts may label not only samples but also supply knowledge at a higher level such as rules, and we consider that the labeling costs depend on many conditions. Moreover, in a CAL process human experts may even profit by improving their own knowledge, too.

1. Introduction

Machine learning is based on sample data. Sometimes, these data are labeled and, thus, models to solve a certain problem (e.g., a classification or regression problem) can be built using targets assigned to input data of the model. In other cases, data are unlabeled (e.g., for clustering problems) or only partially labeled. Correspondingly, we distinguish the areas of supervised, unsupervised, and semi-supervised learning. In many application areas (e.g., industrial quality monitoring processes [1], intrusion detection in computer networks [2], speech recognition [3], or drug discovery [4]) it is rather easy to collect unlabeled data, but quite difficult, time-consuming, or expensive to gather the corresponding targets. That is, labeling is in principal possible, but the costs may be enormous.

This article focuses on *active learning (AL)*, a machine learning area that is sometimes considered to be a special case of semi-supervised learning (SSL) [5], but it can also be regarded as being closely related to SSL. AL starts with an initially unlabeled or very sparsely labeled set of samples and iteratively increases the labeled fraction of the training data set by "asking the right questions". These questions are then answered by humans (e.g., experts in an application domain), by simulation systems, by means of real experiments, etc., often modeled by an abstract "oracle". Basically, the "idealized" goal of AL is to obtain a model with (almost) the performance of a model trained with a fully labeled data set at (almost) the cost of an unlabeled data set. Thus, we also can say that SSL has to *exploit* the knowledge contained in the unlabeled data, while AL has to *explore* the knowledge contained in the unlabeled data. Typically, the following assumptions are made (amongst others) in AL:

1. The oracle labels single samples or sets of samples (called queries depending on the AL type, see below) presented by an active learner.
2. The oracle is omniscient and omnipresent, i.e., it always delivers the correct answers and it is always available).
3. The labeling costs for all samples are identical.

This article develops a vision to overcome these limitations that are definitely not realistic for many applications. In particular, we assume that

1. an expert may be wrong for various reasons, e.g., depending on her/his experience in the application domain (we still assume we have no "malicious or deceptive experts" that cheat or attack the active learner),
2. there might be several or even many experts with different expertise (e.g., degree or kind of experience),
3. the experts may label not only samples but also infer knowledge at a higher level such as rules (e.g., by assigning a conclusion to a presented premise), and
4. the labeling costs depend on many conditions, e.g., whether samples or rules are labeled, on the location of samples in the input space of a model (i.e., making labeling more or less difficult), the degree of expertise of a human, etc.

Moreover, there may be several tasks that have to be fulfilled at the same time (e.g., movies that are assessed regarding several criteria) and different kinds of information sources (e.g., human experts and simulation systems).

We envision *collaborative active learning (CAL)* approaches where the above limitations no longer hold. Moreover, the humans involved in such a CAL process shall be offered the opportunity to profit by improving their own knowledge.

The field of AL recently has awoken the interest of many companies, such as Microsoft, IBM, Siemens, AT&T, Mitsubishi, or Yahoo. Existing publications of those companies show that AL can be successfully utilized to solve a wide range problems, such as:

- *Text Classification:* Microsoft Research describes in [6] an AL approach for providing personalized news to Project Emporia [7] users in real-time. The news on user selected topics are coming from RSS feeds and tweets. The main challenge consists in training the underlying classifier on-line, as the news' trends have to be promptly classified. This problem is solved in two steps: First, appropriate unlabeled tweets are selected by means of AL and forwarded to Amazon Mechanical Turk (AMT) [8] for labeling. Second, a Bayesian corroboration model, that takes the reliabilities of AMT workers into account, is used to label the corresponding tweets.

  Microsoft and Yahoo Lab present a further application of AL for detecting and filtering abusive user-generated content (UGC) on the Web in [9]. The huge number of posts and the user's ability to learn how to avoid static, hand-coded detection rules, motivates the need of fast, automatic detection systems, that periodically update based on newly labeled data, provided by paid annotators. Therefore, AL is used to reduce the labeling costs.

  IBM uses AL in [10] to carry out a sentiment analysis of texts, i.e., an automatic analysis of emotions expressed in posts. The goal is to identify whether a post reveals positive or negative opinions, e.g., criticism or approval.

- *Speech Recognition:* In [11, 12] AT&T shows how AL techniques can be employed to reduce the effort of transcribing natural spoken language required to generate labeled instances for data-driven speech and language processing systems. The goal of such a system may be to automate the classification of customer calls according to their content (e.g, account balance or rates inquiries) and consequently work through the customer requests with the help of a dialog. The transcription of an utterance is usually performed by a human expert and is very expensive, so the costs for extension and improvement can be reduced by an AL approach. At first, the system is trained based on an initial set of transcriptions and, in the following steps, it iteratively selects from a large pool of utterances those with a low transcription confidence. These utterances are then transcribed by an expert and the system is updated.

  Another approach to minimizing the transcription effort is presented by Microsoft Research in [13], which combines AL with semi-supervised learning to select most likely misclassified as well as frequently occurring, untranscribed utterances.

- *Image Classification:* Mitsubishi Electric Research Laboratories present in [14] an AL approach to minimize the image classification cost. The novelty of the presented approach consists in its possibility to start the training process without being aware of the total number of classes (categories). Initially, a classifier is trained based on a small amount of labeled images. Then, a pair of an unlabeled and a labeled image is selected and presented to the human expert for a "match" or "no-match" response. In case of a "match", the queried image is labeled accordingly (the same category as the labeled image), added to the labeled set and the classifier is updated. Otherwise, based on the selection algorithm another labeled sample is selected for comparison. The query step is repeated until the "match" response is obtained or the unlabeled sample does not match any of the labeled

samples. In the latter case, the number of classes is increased and a new category (label) is assigned to the queried sample.

- *Drug Design:* Quantitative Medicine[1] claims to offer the first AL software as a service (SaaS) solution for drug discovery and development. By using AL techniques the efficiency of identifying optimal drug candidates is substantially improved.

- *Malware Detection:* In order to detect and remove malware (malicious software) most anti-virus vendors rely on signature-based detection of previously seen malwares. Therefore, they have to keep their signature repository up to date by collecting suspicious files, which are then manually analyzed by security experts. As the labeling process is a time-consuming task, AL techniques may help the security experts to reduce the number of manually inspected files. Telekom Innovation Laboratories present in [15] an AL framework to select the most likely to be malware files for manual inspection. Support vector machines (SVM) with radial basis function (RBF) kernel are used as classification algorithm. The most informative samples are acquired in two phases: First, the files for which the classifier is most unconfident (low distance to the separating hyperplane) about the category (malware or benign) are selected for manual analysis. In the second phase, the files that reside deep inside the malicious side (high distance to the separating hyperplane) are picked out for further investigation. The proposed AL techniques was tested in a 10 day experiment, showing an improvement in the daily number of new detected malwares.

- *Recommender Systems:* The Android application Shopr [16] is a mobile recommender system that suggests clothing items from stores situated in user's vicinity. The system requires feedback from users in order to understand and fulfill his/her desires. At this point comes AL into play narrowing down the item space to those that match the user's interest, taking both similarity and dissimilarity between the items into account. For a detailed description of the AL-based recommender system see [17].

In the past years there is a strong increase of devices connected to internet which generate a storm of raw data, making the term *Big Data* become popular. From a data analytics point of view, big data may be defined by the four *V*'s: *V*olume, *V*ariety, *V*elocity and *V*eracity. Volume describes the big size of data, Velocity indicates fast generating rates, Variety implies the heterogeneity of data, and Veracity refers to the uncertainty of data quality [18, 19]. Further, the *Internet of Things* aims at connecting everything with everything, leading to a huge network of things that should possess a certain intelligence that may support and improve the daily life of people around the world. To extract high-quality information from big data it may necessary for the system to receive feedback, e.g, from domain experts. Moreover, the humans should benefit too, by exchanging and improving their knowledge. With regard to the previously presented trends, CAL may contribute to addressing these new, future challenges.

Altogether, we can be sure that there will also be an increasing interest in AL and, as many limitations of AL are abolished, in CAL, too.

In the remainder of this article, we first present some foundations of AL in Section 2 and summarize results of own, preliminary work in Section 3. Section 4 presents some experimental results for our AL techniques. In Section 5 we investigate the above challenges of CAL in more detail and briefly discuss possible solutions. Finally, Section 6 concludes the article by taking a look at possible application fields.

2. Overview of Active Learning Foundations

The motivation of AL is that obtaining plenty of unlabeled data is often quite cheap, while acquiring labels is a task with high costs (monetary or temporal). AL is based on the hypothesis that a process of (iteratively) asking an *oracle* for labels and refining the current model can be realized in a way such that

- the performance of the resulting model is comparable to the performance of a model trained on a fully labeled data set and

[1] `http://qtmed.com/home-page/active-learning/` – last access 02/24/2015

- the overall labeling costs to obtain the final model are much lower (typically simply measured by the number of labels).

Actually, to address the previous requirements it is possible to build an *active learner* that is based on a complementary pair of *model* (e.g., a classifier) and *selection strategy*. With a selection strategy, the active learner decides whether a sample is *informative* and asks the oracle for labels. Here, informative means that the active learner expects a (high) performance gain if this sample is labeled (similarly, a set of samples can also be called informative).

Basically, various kinds of models can be used for AL, but the selection strategy should always be defined depending on the model type (e.g., whether SVMs, neural networks, probabilistic classifiers, or decision trees are chosen to solve a classification problem). In the remainder of this article we focus on AL for classification problems (see [20–23], for example), but AL can be applied to modify the results of clustering (see [24], for example) or to regression problems, too (see [25–28], for example).

In the field of AL, membership query learning (MQL) [29], stream-based active learning (SAL) [30], and pool-based active learning (PAL) [31] are the most important paradigms (see Figure 1, left hand side).

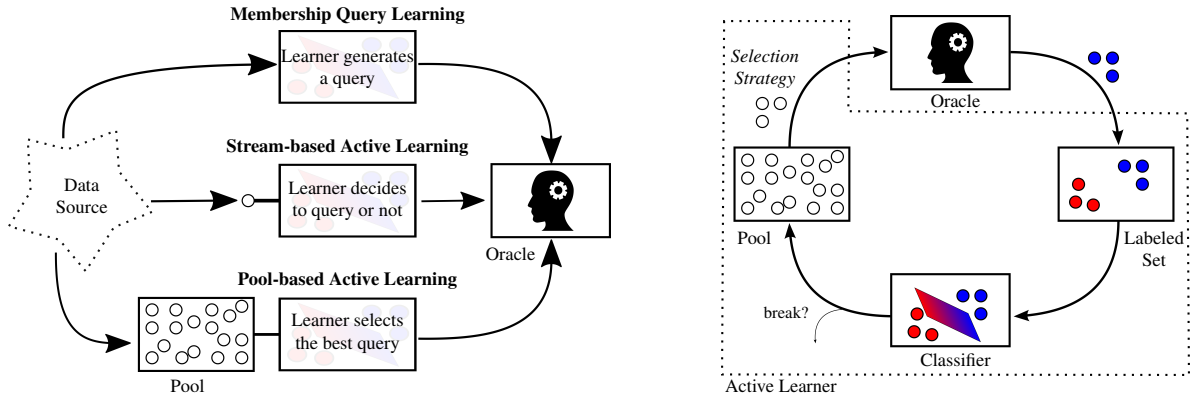
In an (MQL) scenario, the active learner may query labels for any sample in the input space, including samples generated by the active learner itself. Lang and Baum [32], for example, describe an (MQL) scenario with human oracles for classifying written digits. The queries generated by the active learner turned out to be some mixtures of digits, therefore being too difficult for a person to provide reliable answers.

An alternative to (MQL) is (SAL), which assumes that obtaining unlabeled samples generates low or no costs. Therefore, a sample is drawn from the data source and the active learner decides whether or not to request label information. Practically, in an (SAL) setting the source data is scanned sequentially and a decision is made for each sample individually. The SAL scenario has been applied, for example, in [33] to learn ranking functions for retrieving for-sale houses listed on realtor.com that meet the user's preferences.

Typically, SAL selects only one sample in each learning cycle.

For many practical problems a large set of unlabeled samples may be gathered inexpensively and this set is available at the very beginning of the AL process. This motivates the PAL scenario. The learning cycle of PAL is depicted in Figure 1 on the right hand side. Typically, PAL starts with a large pool of unlabeled and a small set of labeled samples. On the basis of the labeled samples the active learner is trained. Then, based on a selection strategy, which considers the "knowledge" of the learner, a query set of unlabeled samples is determined and presented to the oracle (e.g., a human domain expert), who provides the label information. The set of labeled samples is updated with the newly labeled samples and the learner updates its knowledge. The learning cycle is repeated until a given stopping condition is met.

In the remainder of this article we focus on PAL for the sake of simplicity. Many ideas, however, may be transferred to MQL or SAL.



(a) Main AL pradigms.

(b) Learning cycle of standard PAL.

Figure 1: Overview of main AL scenarios with focus on PAL.

A selection strategy for PAL has to fulfill several tasks, two of which shall be given as an example: At an early stage of the AL process, samples have to be chosen in all regions of the input space covered by data (*exploration phase*). At a late stage of the AL process, a fine-tuning of the decision boundary of the classifier has to be realized by choosing samples close to the (current) decision boundary (*exploitation phase*). Thus, "asking he right question" (i.e., choosing samples for a query) is a multi-faceted problem and various selection strategies have been proposed and investigated. We want to emphasize that a successful selection strategy has to consider structure in the (un-)labeled data. We come back to this point later.

Typically, the very limiting assumptions listed in Section 1 are made concerning the oracle and the labeling costs (omniscient, omnipresent oracle that labels samples on a fixed cost basis). Moreover, some other aspects of real-world problems are often more or less neglected by current research, for example:

- In real-world applications, AL has often to start "from scratch", i.e., with no labels at all. This requires sophisticated selection strategies with different behaviors at different phases of the AL process.
- Parameters of the active learner (including parameter of training algorithms of the classifier and the selection strategy) cannot be found by trial-and-error. AL only allows for "one shot".

There are several articles that assess the state of the art in AL. We do not want to replicate this work here but refer to the (in our opinion) most important articles:

- A general introduction to AL, including a discussion of AL scenarios and an overview of query strategies is provided in [34].
- A detailed overview of relevant PAL techniques is part of [20]. In addition to single-view/single-learner methods, alternative approaches are outlined: multi-view/single-learner, single-view/multi-learner, and multi-view/multi-learner.
- For certain problem areas it makes sense to use AL in combination with semi-supervised learning (SSL). AL techniques that integrate SSL techniques are succinctly presented in [21].
- AL in combinations with SVM for solving classification problems are summarized in [35, 36].

3. Preliminary Work – *What are we able to do right now*?

In our preliminary work we tried to overcome some of the mentioned limitations. In particular, we focused on

1. capturing structure in (un-)labeled data to support the exploration/exploitation phases of new selection strategies for PAL,
2. self-adapting weighting schemes for different criteria combined in a selection strategy, and
3. parameter-free AL (apart from parameters that can be found offline using the unlabeled data before the AL process starts, for instance).

We still make the (also very limiting) assumptions concerning oracles and labeling costs that will be addressed later (cf. Section 5). Solutions to the above challenges were not developed separately but with a holistic view on the problem field. Thus, the following subsections sketch closely related solutions and refer to the publications where these approaches are described in much more detail.

3.1. Capturing Structure in Data

Machine learning becomes possible if certain "regularities" or "patterns" in data can be identified and exploited. For classification problems we may assume that the data form clusters of arbitrary shape. For AL, such structure in data has to be captured to improve selection strategies and/or to improve the training of the classifier. In our work, we made two important assumptions that will be outlined in the following. We want to emphasize first that we do not claim that these assumptions always hold. They hold often and to a certain degree and, thus, building solutions based on these assumptions leads to an average, but yet significant improvement of a classifier's performance (shown on average over a number of benchmark data sets, see Section 4).

First, we assume that there is a correspondence of processes in the real world that "generate" the samples that we observe and want to classify on the one hand, and clusters in the training data on the other hand. We may assume further that processes are uniquely assigned to classes and, thus, clusters can also be uniquely assigned to classes. Of course, this does not contradict the fact that clusters belonging to different classes might widely overlap, which means that they cannot be clearly identified if we have the unlabeled data only.

Second, in the real world, these data-generating processes (and the mechanisms necessary to observe these processes) are affected by a (superposition of) random influences (or influences that we have to regard as being random). Two examples are (1) stochasticity which is inherent to certain processes or (2) measurement effects such as sensor noise. Following the generalized central limit theorem [37] we may assume that structure in continuous (real-valued) input dimensions of a classifier may well be captured by means of (mixtures of) Gaussian (i.e., normal) distributions, called components of this mixture model. Apart from that reason, under some mild assumptions it is always possible to model continuous densities using mixtures of Gaussians with arbitrary precision. In practical applications, particular discrete dimensions (integer dimensions) can often be handled like real ones.

Based on these assumptions we developed solutions in an evolutionary approach.

1. *Capture Data Structure in AL:* In a first step, we decided to use probabilistic generative models to capture structure in data and to build classifiers based upon these models. The probabilistic models are parametric density models, mixtures of Gaussians for continuous (or integer where appropriate) input dimensions and special cases of multinomial distributions for the categorical ones. These models can be parametrized (trained) from a set of unlabeled data either in an expectation maximization approach or in a Bayesian approach, called variational Bayesian inference (VI) [38]. The VI has the advantages that, in contrast to EM, effects caused by singularities can be avoided and the number of components in the mixture models can be determined automatically with a built-in pruning technique [39]. Having found the density model, a classifier (CMM: classifier based on mixture models) is constructed using any available labels. This classifier gradually (i.e., with a certain probability) assigns the model components to classes. The components are intended to model the data originating from the data-generating processes in the real world. Our approach for AL based on CMM now exploits the density information in various ways (see below). It was first proposed in [40] and extended in [20].
2. *Revise Captured Structure during AL:* The approach above has the property that the density models are found in an unsupervised way using the unlabeled data available at the very beginning of the AL process. Any label information becoming available throughout the AL process is not used so far to refine the model. With label information, for example, overlapping clusters assigned to different classes could be identified more easily. The approach above is called CMM_{sha} (shared-components classifier) because in an unsupervised training approach all classes "share" the same density model. In a supervised approach, separate density models are trained for the different classes and then combined. This leads to the CMM_{sep} (separate-components classifier) which may basically perform better in many applications regarding data modeling and classification accuracies. In an AL approach, we must start with a CMM_{sha}, but, when more and more labeled data become available, we may iteratively transform this CMM_{sha} towards a CMM_{sep}. In a second step, we realized this idea by adopting techniques from non-parametric density estimation, nearest neighbor classification [41], and transductive learning. The latter is also a variant of semi-supervised learning where labels are found for unlabeled data using the labeled fraction of data. More details about this AL technique can be found in [21]. Preliminary work has shown that the samples queried by means of this techniques are not completely biased to the actively trained CMM_{sep} and can be (re-)used to train a different classifier paradigm as well, e.g., classifiers such as support vector machines (SVM).
3. *Exploit Structure for AL with SVM:* In principal, generative classifiers such as CMM often perform worse than discriminative classifiers such as SVM in many applications. But, on the other hand no density information (or, more general, information concerning structure of data in the input space of the classifier) can be extracted from standard SVM to use it in selection strategies (see below). Thus, a first idea would be to build a generative data model and a discriminative classifier in parallel in an AL process. Having investigated this idea first, we then decided to follow another idea in a third development step of our AL technique: We developed the responsibility weighted Mahalanobis (RWM) kernel [42], a new kernel for SVM that assesses the similarity of samples by means of a Mahalanobis distance in the case of Gaussian mixtures. Thereby, model components that are assumed to be "responsible" for the generation of a sample get a high weight. The RWM kernel can

be used for AL with SVM in combination with various selection strategies. This third evolution step in our approach to capture structure in data is part of our ongoing research.

3.2. Self-Adapting Selection Strategies

A key component of an AL process is the selection strategy. Uncertainty sampling (US) strategies, for example, are frequently used in AL processes. The idea is to select the sample for which the classifier (or a committee of classifiers) is most uncertain concerning its class assignment. This approach has some drawbacks: As the queried samples are always close to the (current) decision boundary, the exploration of the input space may be suppressed, and if more than one sample is queried in each learning cycle, then the selected samples are similar to each other. A selection strategy should be able to detect all decision regions (exploration phase) and fine-tune the decision boundary (exploitation phase), which means that an efficient and effective selection strategy has to find a good trade-off between exploration and exploitation. Thus, our approaches for selection strategies are based on the following two hypotheses:

1. A selection strategy has to consider various aspects and, thus, must combine several criteria.
2. In different phases of the AL process, these criteria must be weighed differently.

In a first step, we developed the selection strategy 3DS which combines three criteria [40]:

1. the *density* of regions where samples are selected,
2. the *distance* of samples to the decision boundary, and
3. the *diversity* of samples in the query set.

The density is an exploration criterion, the distance is an exploitation criterion, and the diversity has to be considered for query sizes larger than one to avoid asking for redundant information (i.e., for efficiency reasons). These criteria can be weighed individually in a linear combination. Moreover, we developed a self-adaptation scheme for 3DS that weights the density criterion more strongly at the beginning of the AL process, in order to explore different regions. In later cycles it is necessary to exploit the gathered information, therefore the distance criterion is emphasized.

In a second step, we extended the 3DS strategy by another criterion:

- the class *distribution* of samples is considered indirectly by evaluating responsibilities.

That is, this 4DS strategy (see [20] for details) aims at labeling samples in a way such that the distribution of the samples approximates the unknown true class distribution. This is especially beneficial for data sets with an unbalanced class distribution, as the generalization performance is improved. How can this be done as we do not know the labels in advance? This is possible (1) by assuming that processes in the "true" world can uniquely be assigned to classes (see above) and (2) by considering the responsibilities of model components (that model these processes) for the samples. Responsibilities are estimates of conditional probabilities, that indicate how likely specific processes modeled by corresponding components are "responsible" for "the generation of a given sample" (i.e., that the sample originates from the considered processes).

The self-adaptation of weights in 3DS was extended in 4DS with the idea of focussing on the class distribution criterion in initial cycles of an AL process.

3.3. Parameter-free AL

In an AL process, many parameters have to be set: parameters of learning algorithms for classifiers, parameters of selection strategies, etc. Typically, a real application of AL only allows for "one shot" for some of these parameters. The selection strategy, for example , should be parameter-free. Other parameters can be tuned, e.g., those of techniques that capture structure in unlabeled data before the AL process starts.

In our work we addressed the following parameter types:

- *Parameters of techniques needed to capture structure in unlabeled data:* Appropriate parameters of the VI algorithm (see above) can in principal be found by repeated training and analysis of reached likelihood values, for instance. Another possibility is to analyze the representativity measure, as described in [43].

- *Parameters of the selection strategies:* Here, we realized the idea that the active learner should be free of such parameters. In 4DS (and 3DS) we start with appropriate initial weights of criteria in the linear combination and let the system self-adapt these weights (see above). A parameter that still remains as it has definitely to be set by a user is the query size. For a query size of one sample, 4DS already is parameter-free. For larger queries we still have to set the weight for the diversity criterion. Finding appropriate heuristics to set this parameter is part of our current research.

- *Parameters of algorithms for classifier training:* Having found the density model, e.g., by means of VI, no further parametrization is required for constructing the CMM classifier, which can be trained using any available label information. The parametrization of SVM is part of our ongoing research. Here, we also aim at adapting the penalty factor C (of C-SVM) and the kernel width γ (of Gaussian kernels) considering the observations made while applying the parametrization heuristic for C and γ presented in [44].

3.4. Summary of Preliminary Research

We could show that our AL approach is able to boost the classification accuracy significantly. Here "significantly" actually means that we applied statistical tests to show the superiority of our techniques on certain significance levels. Apart from accuracy measures (to assess the effectiveness of our AL approach) such as the *ranked performance* on a number of benchmark data sets, we also applied other measures to assess the efficiency of our AL approach (i.e., the learning speed) such as a *data utilization rate* or the *area under the learning curve* [45]. We also defined a new measure, the *class distribution match* [20].

Some (in our opinion) less important assumptions or achievements were not mentioned so far. Examples are the assumption that the computational costs of AL are negligible compared to labeling costs or the fact that our AL approach is in principal able to start the AL process with a completely unlabeled data set (in contrast to many other approaches).

4. Preliminary Experimental Results

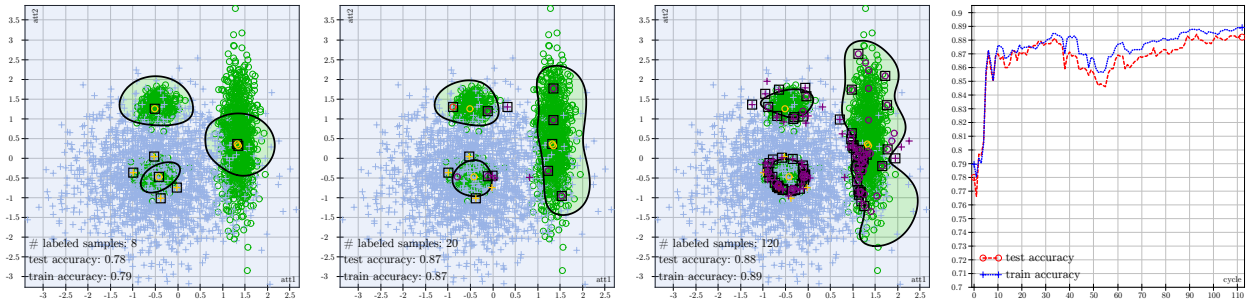
In this section we will compare our AL technique with RWM kernel and 4DS selection strategy to an AL technique with generative classifiers, CMM_{sha} with 4DS strategy, and to an AL technique with discriminative classifiers, SVM with RBF (Gaussian) kernel and uncertainty sampling (US) strategy. First, we visualize the behavior of the two kernels on a two dimensional data set. Second, simulation experiments are conducted and evaluated for 20 benchmark data sets.

4.1. Behavior of SVM with RBF and RWM Kernels

We decided to visualize the behavior of SVM with RBF and RWM kernels on the Clouds data set from the UCI Machine Learning Repository [46]. We performed a z-score normalization, conducted a stratified 5-fold cross-validation and choose the first fold for presentation here. Figures 2 and 3 show the state of SVM with RBF and RWM kernels at different cycles of the AL processes. The orange colored samples correspond to the 8 initially selected samples, whereas the actively selected samples are colored red if they are selected in the current query round otherwise purple. The support vectors are indicated by framing the specific samples with a black square. The decision boundary is depicted as a solid black line. In case of the RWM kernel, Figure 3, the gray colored ellipses correspond to level curves of the Gaussians the RWM kernel is based on (located at centers indicated by large $\times$s). Points on these level curves have a Mahalanobis distance of one to the center of the respective Gaussian.

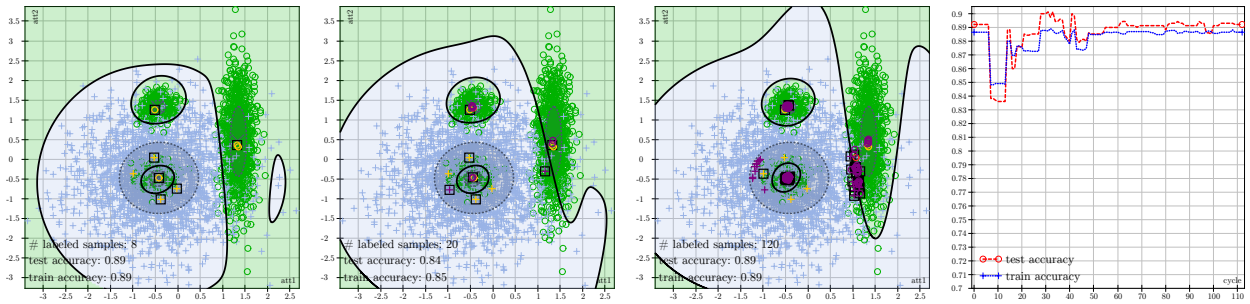
We can see that the SVM with RBF kernel performs worse at the beginning of the AL process, as it does not exploit the unlabeled data. By using structure information derived from the unlabeled data, the RWM kernel achieves a noticeably higher classification accuracy, which is maintained until the end of the AL process.

Furthermore, we can state that US selects samples near the decision boundary (e.g., Figure 2(c)), whereas the 4DS selection strategy by its explorative manner selects samples in other regions, too (e.g., Figure 3(c)).



(a) Cycle 0 (b) Cycle 12 (c) Cycle 112 (d) Learning Curve

Figure 2: Different stages of an AL process with SVM using RBF kernel and US selection strategy. The samples with known label are colored orange (initially selected) and red or purple (actively selected). The decision boundary is depicted as a solid black line.



(a) Cycle 0 (b) Cycle 12 (c) Cycle 112 (d) Learning Curve

Figure 3: Different stages of an AL process with SVM using RWM kernel and 4DS selection strategy. The samples with known label are colored orange (initially selected) and red or purple (actively selected). The decision boundary is depicted as a solid black line.

4.2. Results for 20 Benchmark Data Sets

To evaluate the performance of the active learner with RWM kernel and 4DS selection strategy numerically we conduct experiments on 20 publicly available data sets. For more information regarding general data set characteristics and experimental setup see [21]. The AL techniques are ranked based on a Friedman test [47] with a significance value α of 0.01 followed by a Nemenyi test [48] as post hoc test. A detailed description of the evaluation method can be found in [42].

The classification accuracies achieved by each of the AL paradigms are shown in Table 1. The average ranks and the number of wins summarize the classification performance over all data sets. A good technique yields a low average rank and a large number of wins. The AL technique with RWM kernel and 4DS selection strategy performs best on 16 of the 20 data sets (highest number of wins) and achieves the smallest average rank. The critical difference (CD) plot of the Nemenyi test is shown in Figure 4. As the differences of the average ranks between the active learner for SVM with RWM kernel and 4DS selection strategy and the other two paradigms are greater than the CD we can state that the former performs significantly better than the latter mentioned ones.

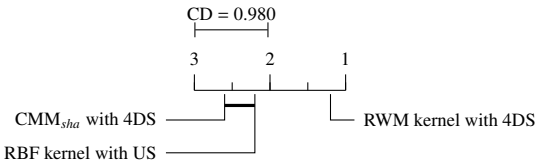


Figure 4: Friedman test with significance value α of 0.01 followed by a Nemenyi post hoc test. Active learners that are connected do not achieve classification accuracies that are significantly different.

Table 1: Classification accuracy (in %) on test data (5-fold cross validation results) for AL of SVM with RBF kernel and US selection strategy, CMM_{sha} and 4DS selection strategy, and SVM with RWM kernel and 4DS selection strategy. The best results are printed in bold face. A good paradigm yields a small rank and a large number of wins.

Data Set	RBF kernel with US	CMM_{sha} with 4DS	RWM kernel with 4DS
Australian	84.06	81.01	**85.65**
Clouds	77.20	**89.30**	88.92
Concentric	99.52	97.32	**99.64**
CreditA	84.35	77.97	**85.65**
CreditG	71.20	69.60	**72.40**
Ecoli	**85.73**	80.30	85.15
Glass	65.89	56.08	**71.01**
Heart	82.96	81.11	**84.81**
Iris	96.67	84.00	**98.00**
Page Blocks	93.06	94.37	**94.52**
Phoneme	80.50	79.22	**80.66**
Pima	**76.04**	70.18	75.00
Ripley	88.96	90.24	**90.40**
Satimage	75.39	83.92	**86.33**
Seeds	91.43	92.86	**97.62**
Two Moons	95.50	99.99	**100.00**
Vehicle	**80.02**	61.11	76.84
Vowel	77.98	88.89	**93.23**
Wine	97.21	96.06	**98.32**
Yeast	56.94	44.47	**58.08**
Mean	83.03	80.90	86.11
Rank	2.200	2.600	1.200
Wins	3.0	1.0	16.0

Table 2: Summary of performance regarding additional evaluation criteria. Larger, positive mean AULC values, smaller mean DUR values and CDM values close to zero are indicators for a good AL approach.

Active Learner	AULC		DUR		CDM	
	Mean	Wins	Mean	Wins	Mean	Wins
RBF kernel with US	0.000	7.0	1.000	6.0	0.080	4.0
CMM_{sha} with 4DS	-1.407	2.5	1.970	1.5	0.058	3.5
RWM kernel with 4DS	2.978	10.5	0.947	12.5	0.048	12.5

Furthermore, three additional evaluation measures are used to assess our results numerically: (1) the area under the learning curve (AULC) [45], (2) the data utilization rate (DUR) [45], and (3) the class distribution match (CDM) [20]. The evaluation results are summarized in Table 2. A good active learner achieves a large, positive AULC value, a DUR value less than one, a CDM value close to zero, and, of course, a large number of wins for each of the evaluation measures. Table 2 shows that the AL technique with RWM kernel and 4DS selection strategy outperforms the other two paradigms regarding all evaluation criteria.

5. Challenges for Future Research on Collaborative Active Learning – *What are the unanswered questions that we will address?*

Up to now, we discussed the state of the art and, in particular, our own efforts to improve the state of the art. In the preceding section we have shown that significant achievements were made. Now, the curtain falls, and the stage is set for the next scene: *collaborative active learning (CAL)*. In our future work we will answer many questions, most of which caused by the harsh limitations sketched in Section 1. To give some examples for questions: Is it possible to train a classifier actively with labels that are subject to uncertainty? Whom do we ask for labels, if there is a "pool" of experts available? Do we query label information from more than one expert? How do we exploit the various, possibly contradictory label information? Is it cheaper or faster, to query for label information for a process (e.g., modeled by a rule premise for which a conclusion has to be found) instead for a batch of samples? How can we give feedback to the experts and how can the experts in turn learn from the active learner? How do we determine if we reached a saturation point (e.g., more label information will not increase the performance)?

5.1. Challenge 1: Uncertain Oracles

In a first step, we address the very obvious fact that oracles are not always right. In principal, labels are subject to uncertainty. Here, the meaning of the term uncertainty is adopted from [49]. That is, "uncertain" is a generic term to address various aspects such as "unlikely", "doubtful", "implausible", "unreliable", "imprecise", "inconsistent", or "vague".

In real-world applications the labels may come from various sources, often but not always humans. Therefore, a new problem arises: The labels are subject to uncertainty for different reasons. For example, the performance of human annotators depends on many factors: e.g., expertise/experience, concentration/distraction, boredom/disinterest, fatigue level, etc. Furthermore, some samples are difficult for both experts and machines to label (e.g., samples near the decision boundary). Results of real experiments or simulations may be influenced, too: There may be stochasticity which is inherent to a certain process, sensor noise, transmission errors, etc., just to mention a few. Thus, we face many questions: How can we make use of uncertain oracles (annotators that can be erroneous)? How do we decide whether an already queried sample has to be labeled again? How do we deal with noisy experts whose quality varies over time (e.g., they gather experience with the task, they get fatigued)? How does remuneration influence the labeling quality of a noisy expert (e.g., if they are payed better, they are more accurate)? How can we decide whether the expert is erroneous or an observed process itself is nondeterministic?

As a starting point, we may assume that the "expertise of an expert" (i.e., the degree of uncertainty of an oracle) is time-invariant and global in the sense that it does not depend on certain classes, certain regions of the input space of the model to be learned (e.g., a classifier), etc. Then, we may ask experts for either (1) one class label with a degree of confidence, (2) membership probabilities for each class (with or without confidence labels), (3) lower bounds for membership probabilities (cf. [50]), (4) a difficulty estimate for a data object that is labeled, (5) relative difficulty estimates for two data objects ("easier" or "more difficult" to label), etc. Then, we have to define appropriate ways to model that uncertainty (e.g., second-order distributions over parameters of class distributions in a probabilistic framework) and to consider it in selection strategies (e.g., with additional criteria) and for the training of a classifier (e.g., with gradual labels).

Different query strategies based on density power divergence [51] (such as β-divergence and γ-divergence) for PAL are proposed in [52], under the assumption that there is a binary classification problem to solve and wrong labels are induced by careless mistakes or failures of experimental instruments. Thus, the noise is assumed to be uniformly-distributed over the samples space.

5.2. Challenge 2: Multiple Uncertain Oracles

In a second step, we address situations where several, individually uncertain oracles (e.g., several human experts with different degree of expertise) contribute their knowledge to an AL process. Thus, AL will now rely on the collective intelligence of a group of oracles. We see this step as a first important step towards collaborative active learning.

In various applications, different, uncertain oracles may contribute labels to an AL process (cf. Figure 5). These experts may not only have different degrees of expertise. They also may have more or less expertise for different parts of the problem that has to be solved, e.g., for different classes that have to be recognized, for different regions of the input space, for different dimensions of the input space (attributes), etc. Also, experts may learn from others and improve over time, for instance. Now, we face many new questions: How can we recognize and model the expertise of humans? How can we decide whom to ask next and how can we merge the uncertain label information (cf. also challenge 1)? How can such exploration and exploitation phases be interwoven? Can we identify groups of experts that should cooperate in a labeling process? How to proceed if experts are only available on a part time basis?

As a starting point, we may initially assume that the "expertise of an expert" is known. In principal, we are convinced that generative, probabilistic models can be taken to model the individual knowledge of experts and not only the "global" knowledge of the active learner. Uncertainty may again be captured with second-order approaches. New selection strategies must then not only choose samples, but also oracles. If the expertise of an oracle is not known, it must be stated either by asking for difficulty or confidence estimates or by comparing it to the knowledge of others (e.g., by asking an expert who has to be assessed questions with already known answers). In order to explore solutions to challenge 2, we are also confronted with the problem of simulation: We have to simulate several uncertain oracles with the different characteristics mentioned above.

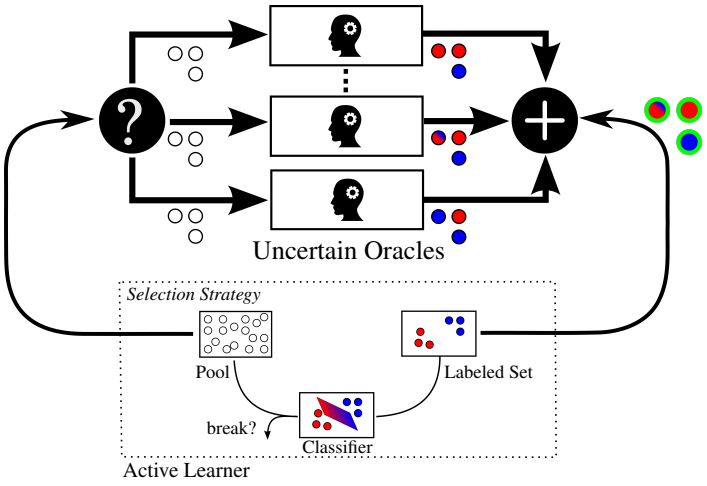


Figure 5: Learning cycle of PAL with multiple uncertain experts.

5.3. Challenge 3: Alternative Query Types

By exploring and modeling the knowledge of oracles as sketched above, the costs of AL would increase substantially. In the other hand, we might ask oracles such as human experts for more abstract knowledge with the goal to reduce the number of queries this way.

In many applications, active learners could ask for more "valuable" knowledge. Examples are conclusions that a human expert gives for a presented rule premise, or correlations between different features or features and classes that an expert provides in order to identify important or redundant features. Questions that arise in this context are: Which questions can be asked? How can we provide (i.e., visualize, for instance) the required information to the expert? How can we combine different kinds of expert statements, e.g., about samples, rules, relations between features, etc? How can we use this information to initialize the models that are trained or to restrict the model capabilities in an appropriate way (e.g., if features are known not to be correlated?

As a starting point, we could investigate the case of annotating rule premises with conclusions. To stay in a probabilistic framework we could obtain user-readable rule premises by marginalization of density functions from a generative process model. Figure 6 gives an example for a density model consisting of three components in a three dimensional input space. The first two dimensions x_1 and x_2 are continuous and, thus, modeled by bivariate Gaussians whose centers are described by larger crosses (+). The ellipses are level curves (surfaces of constant density) with shapes defined by the covariance matrices of the Gaussians. Here, due to the diagonality of the covariance matrices these ellipses are axes-oriented and their projection onto the axes is also shown. The third dimension x_3 is categorical with categories A (red), B (green), and C (blue). The distributions of x_3 are illustrated by the histograms next to every component. Here, only categories with a probability strictly greater than the average are considered in rules in order to simplify the resulting rules. The components modeling sets of circles (green) and crosses (red) are already labeled, resulting in two rules for the components $i = 1$ and $i = 3$:

if x_1 is *low* and x_2 is *high* and x_3 is A or x_3 is B then class = red,
if x_1 is *high* and x_2 is *high* and x_3 is C then class = green.

Now, the active learner presents the following rule premise and asks for a conclusion in form of a class assignment:

x_1 is *high* and x_2 is *low* and x_3 is B.

This information could then be used to (re-)train a classifier, e.g., in a transductive learning step.

5.4. Challenge 4: True Collaboration of Human Experts in AL

In a fourth step we could pave the way for a true collaboration of human experts in AL, which will essentially be based on the capability of humans to learn and the ability of the active learner to provide appropriate feedback to the humans to enable them to learn. Then, the new technique actually deserves to be called CAL.

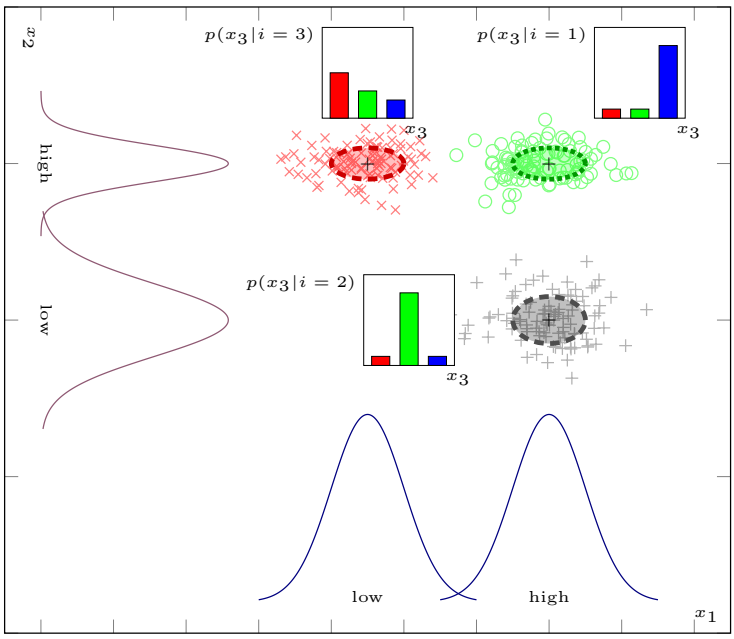


Figure 6: Asking for conclusions of rule premises.

In particular applications, experts would be interested in getting feedback from an active learner, in improving their own knowledge, and sharing their expertise with others. As an important requirement, the active learner must be able to give feedback to the humans and asking for comments on such feedback. Some possible kinds of interactions with humans are (cf. also [53]):

> The following rule appears to be very certain because ... !
> The following rule is in conflict with your knowledge because ... !
> Other experts are much less uncertain concerning the following rule than you are ... !
> Can you confirm the following rule ... ?
> Can you confirm that the following two features are not correlated ... ?
> Can you confirm that the following feature is very important ... ?
> Can you provide additional samples for the following regions of the input space of the classifier ... ?

Solutions to this challenge (which is based on appropriate solutions for the preceding three challenges) will open the door for online AL (cf. SAL mentioned in Section 2). Some of the many new questions that have to be answered are: How can we deal with time-invariant knowledge of oracles? Which information should be provided and how (e.g., with/without certainty estimates, restriction to "crisp" rules or not)? How must we adapt the active learner and the selection strategies? In particular, a compromise has to be found between modeling capabilities on the one hand and the abilities of humans to actually understand readable rules on the other.

As a starting point, we may stay within our probabilistic framework, consider the individual knowledge of humans (challenge 2) and present samples and rules (obtained by marginalization from density models to make them human-readable as sketched above, challenge 3) with fused statements (labels or conclusions) and certainty estimates. Then, the time-variance of human knowledge must be considered by extending the solutions from challenge 2. Again, the evaluation of any proposed techniques will be a challenge by itself.

5.5. Challenge 5: Complex Cost Schemes

In many real-world applications obtaining class information may be possible at different costs, e.g., some class information is more expensive than other or the labeling costs depend on the location of the sample in the input space. This already applies to a "conventional" AL setting without the many ideas discussed in challenges 1 – 4. In a CAL setting, considering complex cost schemes is even more important.

For CAL applications, we must consider costs that depend on

1. *samples with their classes:* As mentioned above, labeling costs may depend on the class (e.g., some kinds of error classes in an industrial production process may be more difficult to detect than others) or on the location of the sample in the input space (e.g., samples close to the decision boundary require higher temporal effort), for instance.
2. *query types:* It is obvious that different labeling costs have to be foreseen for samples (with or without certainty estimates) or for more complex queries such as rule premises. The cost schemes have to be even more detailed in a CAL setting with feedback to the humans (e.g., with queries such as "Can you confirm that ...?").
3. *oracles (experts):* The costs of humans may depend on their expertise, their temporal effort, their availability (e.g., working hour may be modeled with finite costs, otherwise costs are infinite), etc.

In principle, all these costs may change over time, too. The basic questions in this context are: How can a cost scheme be defined? How must the selection strategies be adapted?

As a starting point, we suggest to choose the first point from the list above and investigate solutions in a "classical" AL setting. Then, the most important case for CAL must be addressed, second point. After successfully extending the cost schemes to address the first two tasks we advance to apply CAL with human experts, third point.

5.6. Further Challenges

Some other important challenges must be addressed as well or they will be subject to future research. Apart from these challenges we still face the already discussed requirements such as "parameter-free" AL or self-adaptation of selection strategies.

1. *Stopping Criterion:* Currently, the stopping criterion in real-world applications is based on economic factors, e.g., the learner queries samples as long as the budget allows it. The challenge consists in knowing when to stop querying for labels. One possibility may be to determine the point at which the cost of querying more labels is higher than costs for misclassification. Another possibility is to determine when the learner is at least as good as the group of annotators. For such a "self-stopping criterion", the active learner should be able to assess its own performance.
2. *Performance Assessment:* In AL, the performance of an active learner must be assessed by means of several criteria to capture effectiveness and efficiency of AL. For this purpose, we used a ranked performance measure, a data utilization measure, the area under the learning curve, and a class distribution measure in our preliminary work (see, e.g., [20, 21]). CAL requires additional measures, e.g., to assess the various learning costs or to evaluate the learning progress of human experts.
3. *Dynamic Environment:* Above, we have sketched CAL which takes place in a time-variant environment in the sense that the knowledge of experts improves over time. But, the observed and modeled processes could be time-variant, too. That is, these processes may change slightly (e.g., due to increased wear), become obsolete or new processes corresponding to known or to new, previously unknown, classes may arise during the application of the model. Then, a major challenge consists in developing online AL / CAL techniques that cope with such effects. Mixtures of PAL and SAL techniques would be needed.

6. Summary and Outlook

In this article, we have sketched our vision of collaborative active learning which will certainly be discussed in more detail in the near future. In the novel field of CAL, we would like to concentrate on developing classifiers that take class information uncertainty into consideration, identifying the annotators' level of expertise, making use of different levels of expertise and fusing possibly contradicting knowledge, labeling abstract knowledge, and improving the expertise of the experts. In the envisioned CAL system, human domain experts should benefit from sharing their knowledge in the group. They should receive feedback which will improve their own level of expertise.

In principal, many application areas could benefit from CAL techniques. We can distinguish two possible basic cooperation scenarios: First, scenarios involving specialists (e.g., industrial experts) and, second, scenarios involving non-experts (e.g., crowd-sourcing).

In the former scenario, the number of humans will be lower, the humans are motivated, their expertise will be easier to capture, they collaborate over longer time periods, etc. Typical industrial problems are, for example, product quality control (e.g., deflectometry, classification of errors on silicon wafers or mirrors, analysis of sewing or garments in clothing industry, etc.), fault detection in technical and other systems (e.g., analysis of fault memory entries in control units of cars, analysis of different kinds of errors in cyber-physical systems, etc.), planing of product development processes (e.g., in drug design), or fraud detection and surveillance (e.g., credit card fraud, detection of tax evasion, intrusion detection, or video surveillance). This scenario may be called *Dedicated CAL*.

In the latter scenario, we face larger, open groups of people that will be available for shorter time spans. Typical crowd-sourcing applications will address problems where queries (samples or rules) can easily be understood and assessed by many people including "non-experts" and, thus, be based on video, audio, text or image data. CAL may even be a core component of recommender systems, e.g., to suggest television programs. This scenario may be called *Opportunistic CAL*, as the active learner hay to make the most of the current situation.

References

[1] Sick, B.: Online tool wear monitoring in turning using time-delay neural networks. In: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Volume 1. (1998) 445–448

[2] Hofmann, A., Schmitz, C., Sick, B.: Intrusion detection in computer networks with neural and fuzzy classifiers. In Kaynak, O., Alpaydin, E., Oja, E., Xu, L., eds.: Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP). Volume 2714 of LNCS., Springer-Verlag Berlin Heidelberg (2003) 316–324

[3] Fook, C., Hariharan, M., Yaacob, S., Adom, A.: A review: Malay speech recognition and audio visual speech recognition. In: International Conference on Biomedical Engineering. (2012) 479–484

[4] Malhat, M.G., Mousa, H.M., El-Sisi, A.B.: Clustering of chemical data sets for drug discovery. In: International Conference on Informatics and Systems. (2014) DEKM–11 – DEKM–18

[5] Chapelle, O., Schlkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge, MA, USA (2006)

[6] Paquet, U., Gael, J.V., Stern, D., Kasneci, G., Herbrich, R., Graepel, T.: Vuvuzelas & active learning for online classification (2010)

[7] Microsoft Research: Project Emporia. `http://fuse.microsoft.com/projects/project-emporia` (last accessed 04/22/2015)

[8] Amazon: Mechanical Turk. `https://www.mturk.com/mturk/welcome` (letzter Zugriff 03/31/2015)

[9] Chu, W., Zinkevich, M., Li, L., Thomas, A., Tseng, B.L.: Unbiased online active learning in data streams. In: 17th International Conference on Knowledge Discovery and Data Mining (SIGKDD'11), San Diego, CA (2011) 195–203

[10] Melville, P., Sindhwani, V.: Active dual supervision: Reducing the cost of annotating examples and features. In: Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing. HLT '09, Boulder, CO (2009) 49–57

[11] Hakkani-Tür, D., Riccardi, G., Tur, G.: An active approach to spoken language processing. ACM Transactions on Speech and Language Processing **3** (2006) 1–31

[12] Tur, G., Schapire, R.E., Hakkani-Tür, D.: Active learning for spoken language understanding. In: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03), Hong Kong, China (2003) 276–279

[13] Yu, D., Varadarajan, B., Deng, L., Acero, A.: Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion. Computer Speech & Language **24** (2010) 433–444

[14] Joshi, A.J., Porikli, F., Papanikolopoulos, N.P.: Scalable active learning for multi-class image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012) 2259–2273

[15] Nissim, N., Moskovitch, R., Rokach, L., Elovici, Y.: Novel active learning methods for enhanced PC malware detection in windows OS. Expert Systems with Applications **41** (2014) 5843–5857

[16] Trottmann, U.: Shopr. `https://github.com/UweTrottmann/Shopr` (last accessed 03/31/2015)

[17] 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 (CARR'14), Amsterdam, Niederlande (2014) 10–17

[18] Zhou, Z.H., Chawla, N., Jin, Y., Williams, G.: Big data opportunities and challenges: Discussions from data analytics perspectives. Computational Intelligence Magazine, IEEE **9** (2014) 62–74

[19] IBM: The four v's of big data. `http://www.ibmbigdatahub.com/infographic/four-vs-big-data` (letzter Zugriff 04/27/2015)

[20] Reitmaier, T., Sick, B.: Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4DS. Information Sciences **230** (2013) 106–131

[21] Reitmaier, T., Calma, A., Sick, B.: Transductive active learning – a new semi-supervised learning approach based on iteratively refined generative models to capture structure in data. Information Sciences **239** (2014) 275–298

[22] Constantinopoulos, C., Likas, A.C.: An incremental training method for the probabilistic RBF network. IEEE Transactions on Neural Networks **17** (2006) 966–974

[23] Zhang, Y., Yang, H., Prasad, S., Pasolli, E., Jung, J., Crawford, M.: Ensemble multiple kernel active learning for classification of multisource remote sensing data. Selected Topics in Applied Earth Observations and Remote Sensing **8** (2015) 845–858

[24] Marcacini, R., Correa, G., Rezende, S.: An active learning approach to frequent itemset-based text clustering. In: International Conference on Pattern Recognition. (2012) 3529–3532

[25] Cai, W., Zhang, Y., Zhou, J.: Maximizing expected model change for active learning in regression. In: International Conference on Data Mining. (2013) 51–60

[26] Pasolli, E., Melgani, F.: Gaussian process regression within an active learning scheme. In: International Geoscience and Remote Sensing Symposium. (2011) 3574–3577

[27] Demir, B., L., B.: A multiple criteria active learning method for support vector regression. Pattern Recognition **47** (2014) 2558–2567
[28] Douaka, F., Melgania, F., Benoudjitb, N.: Kernel ridge regression with active learning for wind speed prediction. Applied Energy **103** (2013) 328–340
[29] Angluin, D.: Queries and concept learning. Machine Learning **2** (1988) 319–342
[30] Atlas, L., Cohn, D., Ladner, R., El-Sharkawi, M.A., Marks, II, R.J.: Training connectionist networks with queries and selective sampling. In: Advances in Neural Information Processing Systems 2, Denver, CO, Morgan Kaufmann (1990) 566–573
[31] Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, Springer-Verlag New York, Inc. (1994) 3–12
[32] Lang, K., Baum, E.: Query learning can work poorly when a human oracle is used. In: Proceedings of the IEEE International Joint Conference on Neural Networks. (1992) 335–340
[33] Yu, H.: SVM selective sampling for ranking with application to data retrieval. In: 11th International Conference on Knowledge Discovery and Data Mining (SIGKDD'05), New York, NY, USA (2005) 354–363
[34] Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin, Department of Computer Science (2009)
[35] Jun, J., Horace, I.: Active learning with SVM. In Ramón, J., Dopico, R., Dorado, J., Pazos, A., eds.: Encyclopedia of Artificial Intelligence. Volume 3. IGI Global (2009) 1–7
[36] Kremer, J., Pedersen, K.S., Igel, C.: Active learning with support vector machines. Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery **4** (2014) 313–326
[37] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, Chichester, NY, USA (2001)
[38] Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York, NY, USA (2006)
[39] Fisch, D., Sick, B.: Training of radial basis function classifiers with resilient propagation and variational bayesian inference. In: Proceedings of the International Joint Conference on Neural Networks, Atlanta, GA, USA,, IEEE (2009) 838–847
[40] Reitmaier, T., Sick, B.: Active classifier training with the 3DS strategy. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, Paris, France, IEEE (2011) 88–95
[41] Reitmaier, T., Calma, A.: Resp-*k*nn: A semi-supervised classifier for sparsely labeled data in the field of organic computing. In: Organic Computing Doctoral Dissertation Colloquium. Volume 4., kassel university press GmbH (2014) 85–97
[42] Reitmaier, T., Sick, B.: The responsibility weighted Mahalanobis kernel for semi-supervised training of support vector machines for classification. `http://arxiv.org/abs/1502.04033` (last accessed 04/22/2015)
[43] Fisch, D., Kalkowski, E., Sick, B., Ovaska, S.J.: Towards automation of knowledge understanding: An approach for probabilistic generative classifiers. under review (2015)
[44] Keerthi, S.S., Lin, C.J.: Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation **15** (2003) 1667–1689
[45] Culver, M., Kun, D., Scott, S.: Active learning to maximize area under the ROC curve. In: Proceedings of the Sixth International Conference on Data Mining, Hong Kong, China, IEEE (2006) 149–158
[46] Asuncion, A., Newman, D.: UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml/` (03/02/2014)
[47] Friedman, M.: A comparison of alternative tests of significance for the problem of *m* rankings. The Annals of Mathematical Statistics **11** (1940) 86–92
[48] Nemenyi, P.: Distribution-free Multiple Comparisons. PhD thesis, Princeton University. PhD thesis, Princeton University (1963)
[49] Motro, A., Smets, P., eds.: Uncertainty Management in Information Systems – From Needs to Solutions. Springer Verlag, London, UK (1997)
[50] Andrade, D., Sick, B.: Lower bound bayesian networks – an efficient inference of lower bounds on probability distributions in bayesian networks. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, AUAI Press (2009) 10–18
[51] Basu, A., Shioya, H., Park, C.: Statistical Inference: The Minimum Distance Approach. CRC Press (2011)
[52] Sogawa, Y., Ueno, T., Kawahara, Y., Washio, T.: Active learning for noisy oracle via density power divergence. Neural Networks **46** (2013) 133–143
[53] Horeis, T., Sick, B.: Collaborative knowledge discovery & data mining: From knowledge to experience. In: Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, Honolulu, Hawaii, USA,, IEEE (2007) 421 – 428