



Toward optimal probabilistic active learning using a Bayesian approach

Daniel Kottke¹ · Marek Herde¹ · Christoph Sandrock¹ · Denis Huseljic¹ · Georg Krempf² · Bernhard Sick¹

Received: 20 November 2020 / Revised: 8 March 2021 / Accepted: 12 April 2021 /
Published online: 4 May 2021
© The Author(s) 2021

Abstract

Gathering labeled data to train well-performing machine learning models is one of the critical challenges in many applications. Active learning aims at reducing the labeling costs by an efficient and effective allocation of costly labeling resources. In this article, we propose a decision-theoretic selection strategy that (1) directly optimizes the gain in misclassification error, and (2) uses a Bayesian approach by introducing a conjugate prior distribution to determine the class posterior to deal with uncertainties. By reformulating existing selection strategies within our proposed model, we can explain which aspects are not covered in current state-of-the-art and why this leads to the superior performance of our approach. Extensive experiments on a large variety of datasets and different kernels validate our claims.

Keywords Active learning · Classification · Probabilistic active learning

Editors: Annalisa Appice, Sergio Escalera, Jose A. Gamez, Heike Trautmann.

✉ Daniel Kottke
daniel.kottke@uni-kassel.de

Marek Herde
marek.herde@uni-kassel.de

Christoph Sandrock
christoph.sandrock@uni-kassel.de

Denis Huseljic
dhuseljic@uni-kassel.de

Georg Krempf
g.m.krempf@uu.nl

Bernhard Sick
bsick@uni-kassel.de

¹ Intelligent Embedded Systems, University Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany

² Algorithmic Data Analysis, University Utrecht, Princetonplein 5, 3584, CC, Utrecht, The Netherlands

1 Introduction

To train classifiers with machine learning algorithms in a supervised manner, we need labeled data. Whereas gathering unlabeled instances is easy, the annotation with class labels is often expensive, exhaustive, or time-consuming and needs, consequently, to be optimized. Active learning (AL) algorithms aim to reduce annotation costs efficiently and effectively (Settles 2009). For that purpose, a selection strategy successively chooses the most useful labeling candidate from the pool of unlabeled instances and acquires the corresponding label from an oracle. Throughout this article, we focus on omniscient oracles that always provide labels according to the correct label distribution.

Many AL algorithms completely rely on the information provided by the classifier that is to be optimized (Settles 2009). This might lead to problems: Originally, classifiers are designed to get training data that is representative for the learning task. But this assumption is not valid for AL tasks, as the distribution of labeled instances is biased by the selection strategy (Dasgupta 2009). As a consequence, the estimates (e. g., the class probabilities) from a classifier are also biased, which may lead to a poor assessment of the usefulness of labeling candidates.

In this article, we propose to use a decision-theoretic approach to measure the usefulness of a labeling candidate in terms of (expected) performance gain. To determine the performance of a classifier, we evaluate its predictions on what we know about the data from a data-driven perspective. Although we can estimate the data distribution, we are uncertain about the true class posterior probabilities. Accordingly, we model these class posterior probabilities as a random variable based on our current observations in the dataset. For this model, we use a Bayesian approach by incorporating a conjugate prior to our observations. Thereby, we obtain more robust usefulness estimates for the candidates.

Our approach builds on three pillars that also explain the title of this contribution. Although the general ideas of these pillars have been mentioned in literature before, this article should not be seen as a simple extension of these works as we changed the existing models substantially. (1) We approximate the usefulness of one candidate on a representative subset, as introduced in “*toward optimal AL*” by Roy and McCallum (2001). (2) We estimate the usefulness by determining the decision-theoretic gain in performance, as introduced in “*probabilistic AL*” by Kottke et al. (2016). (3) We use a *Bayesian approach* and introduce a conjugate prior distribution to calculate the predictive posterior distribution. Thereby, we consider the certainty of a classifier on its predictions (Murphy 2006).

The contributions of this article are as follows:

- We propose a new decision-theoretic selection strategy xPAL, which calculates the gain in performance using a Bayesian approach over a set of unlabeled evaluation instances.
- The equations of xPAL can serve as a unifying model for decision-theoretic AL methods. Hence, we can describe other AL methods with simple replacements in xPAL and argue why these modifications impair the candidates’ selection.
- Our experiments on 29 datasets from different domains confirm our approach’s superiority compared to several baselines across different kernels.
- We visualize how the selection strategies assess the usefulness of selection candidates differently on a toy dataset.

The remainder of this article is structured as follows: First, we discuss related work in Sect. 2. In Sect. 3, we define our problem and provide the foundations for our model. In

Sect. 4, we propose our new method xPAL and show how it theoretically and empirically relates to state-of-the-art approaches in Sect. 5. We evaluate our results experimentally and discuss our key findings in Sect. 6. We close this article with a conclusion and an outlook on our future work in that field.

2 Related work

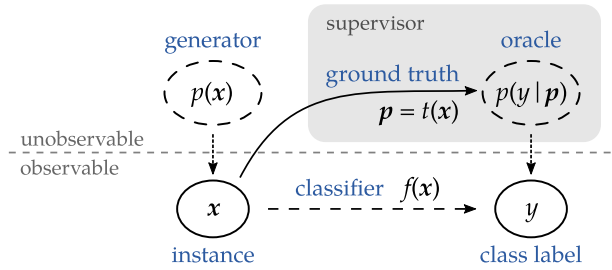
The central component of an AL algorithm is the selection strategy. The most naïve one is to choose the next candidate randomly (Settles 2009). A common heuristic is uncertainty sampling (Lewis and Gale 1994). The idea is to use, e. g., the estimated class posteriors of probabilistic classifiers or the distance to the decision boundary to build a usefulness score (Settles 2012). This *exploits* the current classification hypothesis by labeling instances close to the decision boundary. In contrast to density-based approaches (Nguyen and Smeulders 2004), it ignores the representativeness of selected instances for the entire training set, and fails to perform *exploration* (Bondu et al. 2010; Osugi et al. 2005). That is, it does not search the instance space for large regions with incorrect classifications. This might lead to even worse performance compared to random sampling (Settles 2012). Hence, there exist variants that add random sampling (Žliobaitė et al. 2014; Thrun and Möller 1992), use reinforcement learning (Osugi et al. 2005) or simulated annealing (Zoller and Buhmann 2000) to balance exploitation and exploration, or combine it with a density weight (Donmez et al. 2007) and a variety of further factors, including sample diversity (Weigl et al. 2015; Xu et al. 2007; Brinker 2003) and class priors (Calma et al. 2018).

Uncertainty sampling is a special case of adaptive submodular maximization (Cuong et al. 2014), and several works have established links between submodularity and AL (Cuong et al. 2014; Golovin and Krause 2010; Guillory and Bilmes 2010). An example for a recent approach, built on these works, is filtered active submodular selection (FASS) (Wei et al. 2015). FASS combines uncertainty sampling with a submodular data subset selection framework, capturing both sample informativeness and representativeness. For Gaussian Process classifiers, a Bayesian information theoretic AL approach is Bayesian Active Learning by Disagreement (BALD) (Houlsby et al. 2011). BALD aims to select instances with the highest marginal uncertainty about the class label but simultaneously high confidence for the individual settings of the model's parameters.

The query by committee (QBC) method (Seung et al. 1992) builds classifier ensembles and aims to reduce the disagreement between them. To improve balancing of exploration and exploitation in ensembles of active learners, Baram et al. (2004) proposed a formulation as a multi-armed bandit problem. Here, each active learner corresponds to one slot machine whose relative progress in performance is tracked over time, and on each trial one active learner is chosen for selecting an instance using the EXP4 algorithm. Furthermore, reinforcement learning approaches have been proposed that learn a policy for selecting active learners, for example by modelling active learning as a Markov decision process (Konyushkova et al. 2018).

In 2001, Roy and McCallum (2001) proposed expected error reduction. As shortly addressed in the introduction, they aim to estimate the expected generalization error if a candidate gets an additional label. Thus, they simulate each label for each labeling candidate and evaluate the mean error using the unlabeled instances. To estimate the probabilities, they use the class posteriors provided by probabilistic classifiers. Chapelle (2005)

Fig. 1 A schematic illustration of how to learn from examples



noticed that these estimates are highly unreliable (esp. at the beginning of the training) and therefore suggested the use of a beta prior.

Krempf et al. (2015) and Kottke et al. (2016) address the issue pointed out by Chapelle and named their approach probabilistic AL. They propose to use a distribution of the class posterior probability instead of using the classifier outputs directly. Calculating the expectation over this posterior leads to a decision-theoretic approach and leads to a mathematically sound approach.

3 Problem formulation and foundations

In “The Nature of Statistical Learning Theory,” Vapnik (1995) introduced a holistic concept on how to **learn from examples**. He defined three different components that take part in such a process, namely a generator, a supervisor, and a learning machine.¹ The *generator* creates random vectors $x \in \mathbb{R}^D$ (D-dimensional feature space) independently drawn from a fixed but unknown probability distribution $p(x)$. The *supervisor* provides class labels $y \in \mathcal{Y} = \{1, \dots, C\}$ (C is the number of classes) for every instance x according to a conditional distribution $p(y|x)$ which is also fixed but unknown. In our case, the learning machine is a *classifier* $f_\theta(x)$ with some parameters θ . The goal is to choose that learning machine that approximates the supervisor’s response best.

We adopt the above definition for the active learning scenario by refining the role of the (omniscient) supervisor:

Definition 1 (*Supervisor*) A supervisor consists of:

1. A *ground truth* which is an unknown but fixed, deterministic function $t : \mathbb{R}^D \rightarrow [0, 1]^C$ that maps an instance x to a probability vector $p = t(x)$ with $\sum_{i=1}^C p_i = 1$. Each element describes the true probability for the corresponding class given the instance x .
2. An *oracle* which provides a class label $y \in \mathcal{Y}$ for every instance x according to the ground truth $p = t(x)$. Hence, the label is sampled from a categorical distribution $y \sim \text{Cat}(t(x))$. Note that this implies an omniscient oracle.

¹ We adapt the terms and notation slightly. We use calligraphy for sets, bold font for vectors, and $p(\cdot)$ is either the probability density function or the probability mass of a discrete probability space. Please note the difference between p and $p(\cdot)$ (the latter is always a function).

We visualize the learning process in Fig. 1. The generator provides instances \mathbf{x} for which the oracle provides the class label y based on the ground truth $t(\mathbf{x}) = \mathbf{p} = (p_1, \dots, p_C)$. Unfortunately, we solely have information about the instance-label-pair (\mathbf{x}, y) but not on the generator, the ground truth, or the oracle.

In the technical community, the process of data generation is often described from a *model-driven perspective*: Then it is assumed that each class y has its own data generator $p(\mathbf{x}|y)$. Hence, every instance \mathbf{x} has exactly one label, which is also called ground truth. Due to noise during data generation, different classes might appear in the same region, but still, the true label exists. Our view (as given in Definition 1 and Fig. 1) is purely *data-driven*: Looking at the data, we do not know why there are different labels in the same region. It could be due to noise in the data generation or due to the imperfectness of the oracle. When learning a classifier, the reason does not matter: We only observe that the oracle provides different labels for similar instances according to some proportion \mathbf{p} which we call *ground truth*.

In the field of **active learning**, we assume to have an unlabeled dataset $\mathcal{U} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ (candidate pool) given by the generator. Labels are usually not available at the beginning but can be acquired from the oracle (Settles 2009), which chooses the label according to the ground truth.

A **selection strategy** selects an instance $\mathbf{x} \in \mathcal{U}$, and we acquire the corresponding label $y \in \mathcal{Y}$ from the oracle. We remove the newly labeled instance from the candidate pool $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathbf{x}\}$, add the instance-label-pair to the labeled set $\mathcal{L} \leftarrow \mathcal{L} \cup \{(\mathbf{x}, y)\}$, and retrain the classifier on \mathcal{L} .

We use a kernel-based classifier with **kernel** K which describes the similarity of two instances \mathbf{x} and \mathbf{x}' . In our experiments, we use three different kernels (see Sect. 6) but our method is not restricted to these kernels.

Definition 2 (*Kernel Frequency Estimate*) The kernel frequency estimate $k_{\mathbf{x}}^{\mathcal{L}}$ of an instance \mathbf{x} is determined using the set of labeled instances \mathcal{L} . The y -th element of that C -dimensional vector describes the similarity-weighted number of labels of class y :²

$$k_{\mathbf{x}, y}^{\mathcal{L}} = \sum_{(\mathbf{x}', y') \in \mathcal{L}} \mathbb{1}_{y=y'} K(\mathbf{x}, \mathbf{x}'). \quad (1)$$

We denote $f^{\mathcal{L}}$ as a **classifier** which uses the labeled data \mathcal{L} for training.³ Similar to the Parzen Window Classifier (PWC) used in Chapelle (2005), the classifier $f^{\mathcal{L}}$ predicts the **most frequent class**:

$$f^{\mathcal{L}}(\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} \left(k_{\mathbf{x}, y}^{\mathcal{L}} \right). \quad (2)$$

Our method requires estimating kernel frequencies which is straight-forward for the PWC but also possible for other classifiers. For example, Beyer et al. (2015) estimates kernel frequencies (called label statistics) for Naive Bayes, k -Nearest Neighbour, and Tree-Based classifiers.

² $\mathbb{1}_{cond}$ denotes the indicator function which returns 1 if *cond* is true and 0 otherwise.

³ To simplify the notation, we do not mention the parameters θ .

4 Toward optimal probabilistic active learning using a Bayesian prior

The idea of our approach is to estimate the expected performance gain that a new instance would provide if we would acquire its label from the oracle. Then, we select the most promising instance for actual labeling. Within the next subsections, we explain the necessary steps towards the final method.

4.1 Estimating the risk

In this article, we use the misclassification error as our performance measure (this can easily be changed). To optimize this performance, we minimize the estimated risk using the zero-one loss similarly to Vapnik (1995).

Definition 3 (*Risk, Zero-one Loss*) The risk describes the expected value of the loss L with respect to the joint distribution $p(\mathbf{x}, y)$ given a classifier $f^{\mathcal{L}}$:

$$R(f^{\mathcal{L}}) = \mathbb{E}_{p(\mathbf{x}, y)} [L(y, f^{\mathcal{L}}(\mathbf{x}))] \quad (3)$$

$$= \mathbb{E}_{p(\mathbf{x})} \left[\mathbb{E}_{p(y|\mathbf{x})} [L(y, f^{\mathcal{L}}(\mathbf{x}))] \right]. \quad (4)$$

The zero-one loss returns 0 if the prediction of the classifier $f^{\mathcal{L}}(\mathbf{x})$ is equal to the true class y and 1 otherwise:

$$L(y, f^{\mathcal{L}}(\mathbf{x})) = \mathbb{1}_{f^{\mathcal{L}}(\mathbf{x}) \neq y}. \quad (5)$$

As the generator $p(\mathbf{x})$ is not observable, we use a Monte-Carlo integration using a set of instances \mathcal{E} which is able to represent the generator. For simplicity, we use the complete set of available instances, i. e. the labeled and the unlabeled data ($\mathcal{E} = \{\mathbf{x} : (\mathbf{x}, y) \in \mathcal{L}\} \cup \mathcal{U}$). Following the notation of Japkowicz and Shah (2011), we calculate the empirical risk $R_{\mathcal{E}}$ as follows:

$$R_{\mathcal{E}}(f^{\mathcal{L}}) = \frac{1}{|\mathcal{E}|} \sum_{\mathbf{x} \in \mathcal{E}} \mathbb{E}_{p(y|\mathbf{x})} [L(y, f^{\mathcal{L}}(\mathbf{x}))] \quad (6)$$

$$= \frac{1}{|\mathcal{E}|} \sum_{\mathbf{x} \in \mathcal{E}} \sum_{y \in \mathcal{Y}} p(y|\mathbf{x}) L(y, f^{\mathcal{L}}(\mathbf{x})) \quad (7)$$

4.2 Introducing a conjugate prior

The conditional class probability $p(y|\mathbf{x})$ from Eq. (7) depends on the ground truth t which is unknown (see Fig. 1):

$$p(y|\mathbf{x}) = p(y|t(\mathbf{x})) = p(y|\mathbf{p}) = \text{Cat}(y|\mathbf{p}) = p_y. \quad (8)$$

As a consequence, the probability $p(y|\mathbf{x})$ is exactly the y -th element of the unknown ground truth vector \mathbf{p} . We can use the nearby labels from \mathcal{L} (represented in $\mathbf{k}_x^{\mathcal{L}}$, Definition 2) to

estimate the ground truth \mathbf{p} as the oracle provides the labels according to \mathbf{p} (see Fig. 1). If we assume a smooth distribution \mathbf{p} (in the sense that small changes of \mathbf{x} do not change \mathbf{p} much), the estimate with an appropriate kernel is close to the ground truth with sufficiently many labels. Although the latter cannot be assumed in all cases, domain experts usually have lots of experience when describing a kernel function for their data. Moreover, the results of our experiments in Sec. 6 show that our method also works with a kernel using a simple heuristic. For estimation, we use a Bayesian approach by determining the posterior predictive distribution, i. e. calculating the expected value over all possible ground truth values \mathbf{p} (see Murphy (2006) for details on predictive distributions):

$$p(y|\mathbf{x}) \approx p(y|\mathbf{k}_x^{\mathcal{L}}) = \mathbb{E}_{p(\mathbf{p}|\mathbf{k}_x^{\mathcal{L}})} [p_y] = \int p(\mathbf{p}|\mathbf{k}_x^{\mathcal{L}}) p_y d\mathbf{p}. \quad (9)$$

To determine the posterior probability $p(\mathbf{p}|\mathbf{k}_x^{\mathcal{L}})$ of the ground truth \mathbf{p} at instance \mathbf{x} , we use Bayes' theorem in Eq. (10). The likelihood $p(\mathbf{k}_x^{\mathcal{L}}|\mathbf{p})$ is a multinomial distribution as each label y has been drawn from $Cat(y|\mathbf{p})$ (see Fig. 1).⁴ We **introduce a prior** $p(\mathbf{p})$ which we choose to be a Dirichlet distribution with parameter $\boldsymbol{\alpha} \in \mathbb{R}^C$ as this is the conjugate prior of the multinomial distribution. We choose an indifferent prior and set each element to the same value ($\alpha_1 = \dots = \alpha_C \in \mathbb{R}^{>0}$) such that none of the classes is favoured. Using this prior can be seen as adding α_y pseudo-instances to every class y (Bishop 2006, p. 77). This means that in case of high values of $\boldsymbol{\alpha}$, we need many labeled instances (i. e., high frequency estimates $\mathbf{k}_x^{\mathcal{L}}$) to get distinct posterior probabilities.

As we use the conjugate prior of the multinomial likelihood, there exists an analytic solution for the posterior which is a Dirichlet distribution (Murphy 2006).

$$p(\mathbf{p}|\mathbf{k}_x^{\mathcal{L}}) = \frac{p(\mathbf{k}_x^{\mathcal{L}}|\mathbf{p})p(\mathbf{p})}{p(\mathbf{k}_x^{\mathcal{L}})} \quad (10)$$

$$= \frac{\text{Multinom}(\mathbf{k}_x^{\mathcal{L}}|\mathbf{p}) \cdot \text{Dir}(\mathbf{p}|\boldsymbol{\alpha})}{\int \text{Multinom}(\mathbf{k}_x^{\mathcal{L}}|\mathbf{p}) \cdot \text{Dir}(\mathbf{p}|\boldsymbol{\alpha}) d\mathbf{p}} \quad (11)$$

$$= \text{Dir}(\mathbf{p}|\mathbf{k}_x^{\mathcal{L}} + \boldsymbol{\alpha}) \quad (12)$$

Now, we determine the conditional class probability $p(y|\mathbf{k}_x^{\mathcal{L}})$ from Eq. (9) by calculating the expected value of the Dirichlet distribution (Murphy 2006):

$$p(y|\mathbf{k}_x^{\mathcal{L}}) = \mathbb{E}_{\text{Dir}(\mathbf{p}|\mathbf{k}_x^{\mathcal{L}}+\boldsymbol{\alpha})} [p_y] \quad (13)$$

$$= \int \text{Dir}(\mathbf{p}|\mathbf{k}_x^{\mathcal{L}} + \boldsymbol{\alpha}) p_y d\mathbf{p} = \frac{(\mathbf{k}_x^{\mathcal{L}} + \boldsymbol{\alpha})_y}{\|\mathbf{k}_x^{\mathcal{L}} + \boldsymbol{\alpha}\|_1}. \quad (14)$$

⁴ Normally, the multinomial distribution only allows non-negative integers as observations. Hence, we use it as an analogy. As our probability is normalized, we can also calculate its density for real-valued observations $\mathbf{k}_x^{\mathcal{L}}$.

The last term describes the y -th element of the normalized vector $\mathbf{k}_x^{\mathcal{L}} + \boldsymbol{\alpha}$. For normalization, we use the sum of all elements denoted as the 1-norm $\|\cdot\|_1$.

4.3 Risk difference using the conjugate prior

We insert Eq. (14) into the empirical risk (Eq. (7)). As we approximate $p(y|\mathbf{x})$ with $p(y|\mathbf{k}_x^{\mathcal{L}})$, this is an approximation of the empirical risk based on the labeled data \mathcal{L} . Hence, we add \mathcal{L} as an argument of the estimated empirical risk:

$$\hat{R}_{\mathcal{E}}(f^{\mathcal{L}}, \mathcal{L}) = \frac{1}{|\mathcal{E}|} \sum_{x \in \mathcal{E}} \sum_{y \in \mathcal{Y}} \frac{(\mathbf{k}_x^{\mathcal{L}} + \boldsymbol{\alpha})_y}{\|\mathbf{k}_x^{\mathcal{L}} + \boldsymbol{\alpha}\|_1} L(y, f^{\mathcal{L}}(\mathbf{x})). \tag{15}$$

We now assume that we add a new labeled candidate (\mathbf{x}_c, y_c) to the labeled set \mathcal{L} and denote the new set $\mathcal{L}^+ = \mathcal{L} \cup \{(\mathbf{x}_c, y_c)\}$. To determine how much this new instance-label-pair improved the performance of our classifier f , we estimate the gain in terms of risk difference using the probability $p(y|\mathbf{k}_x^{\mathcal{L}^+})$ to estimate the ground truth. We here use the same observations $\mathbf{k}_x^{\mathcal{L}^+}$ to estimate the risk of both the current and the new classifier (with simulated labels) whereas Roy and McCallum (2001) use $p(y|\mathbf{k}_x^{\mathcal{L}})$ for estimating the performance of $f^{\mathcal{L}}$ and $p(y|\mathbf{k}_x^{\mathcal{L}^+})$ for $f^{\mathcal{L}^+}$. We have two reasons to believe that we are correct: (1) When using different ground truth estimates for calculating the risks, we would not find out if a difference comes from the change in the classifier or from the change in the ground truth function. (2) In this article, we assume that the oracle is omniscient and, therefore, always correct. Consequently, we can assume that more labels should provide more accurate estimates. Accordingly, $p(y|\mathbf{k}_x^{\mathcal{L}^+})$ should also be used for the current classifier $f^{\mathcal{L}}$.

$$\Delta \hat{R}_{\mathcal{E}}(f^{\mathcal{L}^+}, f^{\mathcal{L}}, \mathcal{L}^+) = \hat{R}_{\mathcal{E}}(f^{\mathcal{L}^+}, \mathcal{L}^+) - \hat{R}_{\mathcal{E}}(f^{\mathcal{L}}, \mathcal{L}^+) \tag{16}$$

$$= \frac{1}{|\mathcal{E}|} \sum_{x \in \mathcal{E}} \sum_{y \in \mathcal{Y}} \frac{(\mathbf{k}_x^{\mathcal{L}^+} + \boldsymbol{\alpha})_y}{\|\mathbf{k}_x^{\mathcal{L}^+} + \boldsymbol{\alpha}\|_1} \cdot (L(y, f^{\mathcal{L}^+}(\mathbf{x})) - L(y, f^{\mathcal{L}}(\mathbf{x}))). \tag{17}$$

4.4 The expected probabilistic gain

If we reduce the error under the new model \mathcal{L}^+ , the risk difference in Eq. (17) becomes negative. Therefore, we negate this term as we aim to maximize the gain in Definition 4.

Definition 4 (*Expected Probabilistic Gain*) The probabilistic gain describes the expected change in classification risk R when acquiring the label y_c of candidate $\mathbf{x}_c \in \mathcal{U}$. As the label y_c and the corresponding ground truth $t(\mathbf{x}_c)$ are unknown, we estimate $p(y_c|\mathbf{x}_c)$ with $p(y_c|\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}})$ according to Eq. (14) using $\text{Dir}(\boldsymbol{\beta})$ as prior. We write $\mathcal{L}^+ = \mathcal{L} \cup \{(\mathbf{x}_c, y_c)\}$.

$$\text{xgain}(\mathbf{x}_c, \mathcal{L}, \mathcal{E}) = \mathbb{E}_{p(y_c|\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}})} \left[-\Delta \hat{R}_{\mathcal{E}}(f^{\mathcal{L}^+}, f^{\mathcal{L}}, \mathcal{L}^+) \right] \tag{18}$$

$$\begin{aligned}
&= - \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k}_{x_c}^{\mathcal{L}} + \boldsymbol{\beta})_{y_c}}{\|\mathbf{k}_{x_c}^{\mathcal{L}} + \boldsymbol{\beta}\|_1} \cdot \frac{1}{|\mathcal{E}|} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \\
&\quad \frac{(\mathbf{k}_x^{\mathcal{L}^+} + \boldsymbol{\alpha})_y}{\|\mathbf{k}_x^{\mathcal{L}^+} + \boldsymbol{\alpha}\|_1} \cdot \left(L(y, f^{\mathcal{L}^+}(\mathbf{x})) - L(y, f^{\mathcal{L}}(\mathbf{x})) \right)
\end{aligned} \tag{19}$$

For simplicity, we set $\boldsymbol{\beta} = \boldsymbol{\alpha}$.

We define the selection strategy xPAL to choose the candidate that optimizes the xgain score.

Definition 5 (*Selection Strategy: xPAL*) The selection strategy xPAL (Expected Probabilistic Gain for AL) chooses this candidate $\mathbf{x}_c^* \in \mathcal{U}$ with:

$$\mathbf{x}_c^* = \arg \max_{x_c \in \mathcal{U}} (\text{xgain}(\mathbf{x}_c, \mathcal{L}, \mathcal{E})). \tag{20}$$

5 Theoretical and qualitative comparison

To provide an understanding of how the xPAL selection strategy works, we compare our new method to the most similar selection strategies by reformulating their approaches within our mathematical framework wherever possible. We provide the proofs for all theorems in the supplemental material. In Table 1, we summarize the primary differences and show the computational complexity.

In Fig. 2, we illustrate how the theoretical differences affect the actual choice of eight candidates on a toy dataset with two classes (blue diamonds and red rectangles). For classification, we use the same setup as in Sect. 6. The first eight labeled instances, chosen by the selection strategy, are marked with a gray circle. The background color shows how the respective selection strategy rates the usefulness of an area—darker areas are considered more useful than brighter areas.

5.1 Expected probabilistic gain for AL (xPAL)

As seen in Fig. 2, the currently labeled set \mathcal{L} of xPAL is evenly spaced across the input space. That is, xPAL queried representative samples of the data set in the more explorative phase at the beginning, which leads to a rather good decision boundary with only eight labels. Focusing on the current usefulness scores indicated by green background color, we see that regions close to the decision boundary and regions with very few labels (green area at the bottom) are preferred. Moreover, we notice more usefulness at the right decision boundary compared to the left one as this area is seen as being more relevant (due to the higher density).

5.2 Expected error reduction (EER)

Theorem 1 *The selection criterion of expected error reduction (EER) by Roy and McCallum (2001) can be written as follows. The extension of adding a beta-prior ϵ proposed by Chapelle (2005) is given in blue color.*

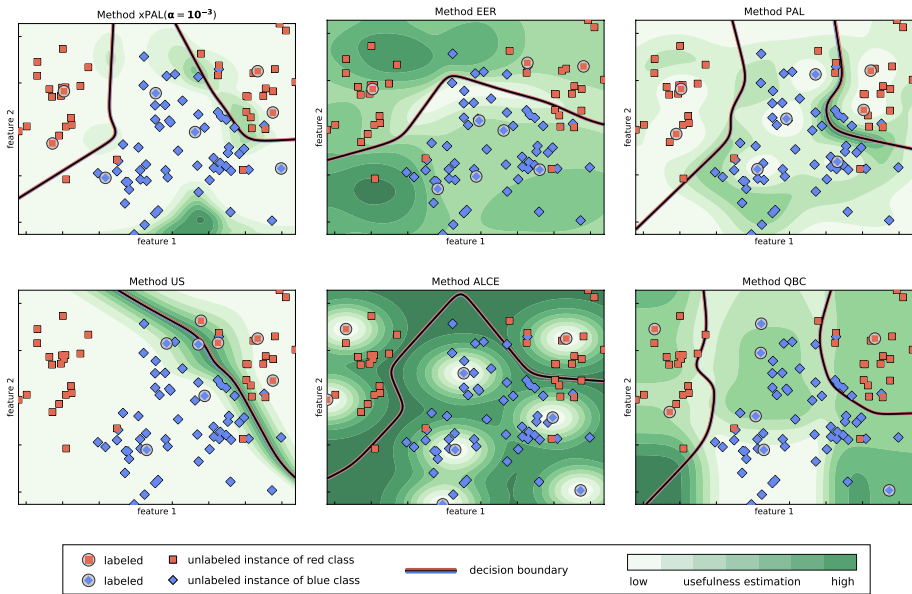


Fig. 2 Visualization of acquisition behavior for different selection strategies. The green color indicates how useful a selection strategy considers a region. The usefulness depends on the selection criterion of the strategy. The eight labeled instances have been selected by the corresponding selection strategy. Thereby, one can see where the selection strategy selected instances in the past and how the usefulness is spatially distributed to select the next instance for labeling

Table 1 Summary of differences between xPAL and the four most similar methods evaluated on four criteria: (1) Is the usefulness estimated on a representative subset? (2) Does the method consider the performance gain? (3) Is some sort of prior included to handle uncertainties? (4) What is the asymptotic time complexity for determining the usefulness of one candidate sample?

Method	\mathcal{E}	Gain	Prior	$\mathcal{O}(\cdot)$
xPAL	✓	✓	✓	$ \mathcal{E} \cdot \mathcal{Y} ^2$
PAL	✗	✓	✓	$ \mathcal{Y} ^2$
EER	(✓)	✗	(✓)	$ \mathcal{U} \cdot \mathcal{Y} ^2$
US	✗	✗	✗	$ \mathcal{Y} $

$$\begin{aligned}
 eer(\mathbf{x}_c, \mathcal{L}, \mathcal{U}) &= \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \epsilon)_{y_c}}{\|\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \epsilon\|_1} \cdot \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x} \in \mathcal{U}} \\
 &\sum_{y \in \mathcal{Y}} \frac{(\mathbf{k}_{\mathbf{x}}^{\mathcal{L}^+} + \epsilon)_y}{\|\mathbf{k}_{\mathbf{x}}^{\mathcal{L}^+} + \epsilon\|_1} \cdot L(y, f^{\mathcal{L}^+}(\mathbf{x}))
 \end{aligned}
 \tag{21}$$

Comparing Eqs. (21) to (19), we see that there are only a few differences highlighted in orange color. The main difference is the optimization objective as expected error reduction tries to query instances that minimize the expected error instead of the

expected gain as in xPAL. Originally, Roy and McCallum (2001) also introduced the reduction in error as the optimization criterion. By using $p(y|\mathbf{k}_x^{\mathcal{L}})$ to estimate the risk for the current classifier, the corresponding term is constant for all candidates and can be omitted (see also the discussion in Sect. 4.3). Second, EER neglects the labeled instances \mathcal{L} as it only uses \mathcal{U} for Monte-Carlo integration. They assume that the unlabeled instances approximate the generator $p(\mathbf{x})$ sufficiently well. In the original version, Roy and McCallum (2001) point out that the posterior estimates need to be reliable. Later, Chapelle (2005) addresses this limitation by introducing a beta-prior ϵ (highlighted in blue), which serves a similar goal as our prior α .

Although the theoretical differences of the two strategies seem to be small, we see a clear difference in the acquired instances and in the usefulness estimation in Fig. 2. Interestingly, the region close to the decision boundary is considered the least useful. Accordingly, EER neglects information there. Hence, this detail in the model has a huge impact on the selection and also the performance.

5.3 Probabilistic active learning (PAL)

Theorem 2 *The selection criterion of (multi-class) probabilistic active learning (PAL) by Kottke et al. (2016) can be written as follows.*

$$\begin{aligned} \text{pal}(\mathbf{x}_c, \mathcal{L}) = & -\hat{p}(\mathbf{x}_c) \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{1})_{y_c}}{\|\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{1}\|_1} \sum_{y \in \mathcal{Y}} \frac{(\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}^+} + \mathbf{1})_y}{\|\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}^+} + \mathbf{1}\|_1} \\ & \cdot \left(L(y, f^{\mathcal{L}^+}(\mathbf{x}_c)) - L(y, f^{\mathcal{L}}(\mathbf{x}_c)) \right) \end{aligned} \quad (22)$$

The probabilistic active learning approach by Kottke et al. (2016) does not consider a set \mathcal{E} for risk estimation but estimates the risk locally only for the candidate \mathbf{x}_c . Hence, we set $\mathcal{E} = \{\mathbf{x}_c\}$. Instead, they include an estimated density weight $\hat{p}(\mathbf{x}_c)$ for their local gain. As a prior distribution, they use the indifferent prior $\mathbf{1}$. The original method is non-myopic. As xPAL is myopic, we ignored this for the theoretical discussion. Note that the original PAL version also uses $p(y|\mathbf{k}_x^{\mathcal{L}})$ to estimate the performance of the current classifier similarly to Roy and McCallum (2001). Due to a difference in calculating the performance estimate, we can show that the original PAL equations can also be written as in Eq. 22. So in fact, they were using $p(y|\mathbf{k}_x^{\mathcal{L}^+})$ without knowledge. The complete 2-pages proof is given in the appendix.

In general, we see a similar acquisition behavior of PAL and xPAL (see Fig. 2). We see areas of high usefulness near the decision boundary and in sparsely labeled regions. It seems that xPAL is more sensitive to the actual position of the instances as it considers the set \mathcal{E} , and PAL only approximates this by using the density $\hat{p}(\mathbf{x}_c)$. Hence, the influence of a new label on the complete classification task is only approximated in PAL.

5.4 Uncertainty sampling (US)

Theorem 3 *The selection criterion of confidence-based uncertainty sampling (US) by Lewis and Gale (1994) can be written as follows.*

$$\text{us}(\mathbf{x}_c, \mathcal{L}) = \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{0})_{y_c}}{\|\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{0}\|_1} \cdot L(y, f^{\mathcal{L}}(\mathbf{x}_c)) \quad (23)$$

Uncertainty sampling does not consider a set for risk estimation, but it solely estimates the error at the candidate \mathbf{x}_c based on the current observations without any prior. Hence, it completely relies on the class posterior estimates from the classifier. Therefore, it might overestimate its certainty.

We observe this problem in Fig. 2 as US only finds one decision boundary and sticks at exploiting this. As it is not aware that the class posteriors on the left are highly unreliable (no labeled data here), it will only consider this region if the labels of all other candidates have been acquired. We notice a lack of exploration.

5.5 Active learning with cost embedding (ALCE)

The approach proposed by Huang and Lin (2016) uses an embedding with some special distance measure in a hidden space with non-metric multidimensional scaling. As this follows an entirely different way of approaching the problem, it is not possible to transfer this algorithm to our framework. As shown in Fig. 2, this approach explores the data space quite uniformly and is rather exploratory than exploitative.

5.6 Query by committee (QBC)

Query by committee (Seung et al. 1992) uses an ensemble of classifiers that are trained on bootstrapped replicates of the labeled set \mathcal{L} . With few labels, the strategy explores the dataset due to high randomness in the subsets (see Fig. 2). Later, it starts exploiting more.

6 Experimental evaluation

To evaluate the quantitative performance of xPAL, we conduct experiments on real-world datasets.⁵ We provide information on the used datasets, algorithms, and the experimental setup. We compare xPAL to state-of-the-art methods using scikit-activeml⁶ (Kottke et al. 2021) and show how the prior parameter affects the results.

6.1 Datasets and competitors

We selected 27 datasets from the openML library (Vanschoren et al. 2013) and two pre-processed text datasets from Hernández-González et al. (2018) with TF-IDF features. For

⁵ <https://github.com/dakotaprobal>.

⁶ <https://github.com/scikit-activeml/scikit-activeml>.

the latter, we assigned the majority vote as the true class. In the supplemental material, we list all used datasets with their openML-identifier and show specific characteristics such as the number of instances, features, and instances per class.

Next to xPAL, we use multi-class probabilistic AL (PAL) by Kottke et al. (2016), confidence-based uncertainty sampling (US) by Lewis and Gale (1994), active learning with cost embedding (ALCE) by Huang and Lin (2016), query by committee (QBC) by Seung et al. (1992), expected error reduction (EER) by Chapelle (2005), and a random selector. We set all parameters according to the default values in the paper. For QBC, the disagreement within the randomly drawn sets, measured by the Kulback-Leibler divergence, describes the usefulness of a candidate. We use 25 classifiers as the committee and each of them is trained on a bootstrapped version of \mathcal{L} with only a selection of features according to (Shi et al. 2008).

Additionally, we implemented a baseline that has additional access to *all* labels of the unlabeled set \mathcal{U} . It successively (*greedily*) selects the candidate, which minimizes the true empirical risk on \mathcal{U} and \mathcal{L} , called GREEDY-ALL. It is equal to xPAL where the estimated class probability from Eq. 8 is set to one for the true class.

6.2 Experimental setup

To evaluate our experiments, we randomly split each dataset into a training set consisting of 60% of the instances and a test set containing the remaining 40% and repeat that 100 times. As we start without any labeled instances, \mathcal{U} contains the whole training set at the beginning, and \mathcal{L} is empty. We acquire 200 labels for every dataset or stop when \mathcal{U} is empty.

For classification, we use the Parzen window classifier for all selection strategies. We applied three different kernels depending on the type of data. For numerical data, we z-standardize all features and use a radial basis function (RBF) kernel with bandwidth γ which is defined as follows:

$$K_{\text{rbf}}(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2). \quad (24)$$

We set the bandwidth of the kernel ($\gamma = 1/(2s^2)$) according to the mean criterion proposed by Chaudhuri et al. (2017) with $\sigma_p = 1$:

$$s = \sqrt{\frac{2N \sum_{j=1}^D \sigma_p^2}{(N-1) \ln \frac{N-1}{\delta^2}}}, \quad \delta = \sqrt{2} \cdot 10^{-6}, \quad N = \min(|\mathcal{U} \cup \mathcal{L}|, 200). \quad (25)$$

For categorical data, we use the hamming-distance kernel proposed by Hutter et al. (2014):

$$K_{\text{ham}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \sum_{d=1}^D \mathbb{1}_{x_d \neq x'_d}\right), \quad (26)$$

where the hyperparameter γ is again determined through the mean bandwidth criterion.

For the text datasets which contain TF-IDF features, we apply the cosine similarity kernel

$$K_{\text{cos}}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^T \cdot \mathbf{x}'}{\|\mathbf{x}\|_2 \cdot \|\mathbf{x}'\|_2}. \quad (27)$$

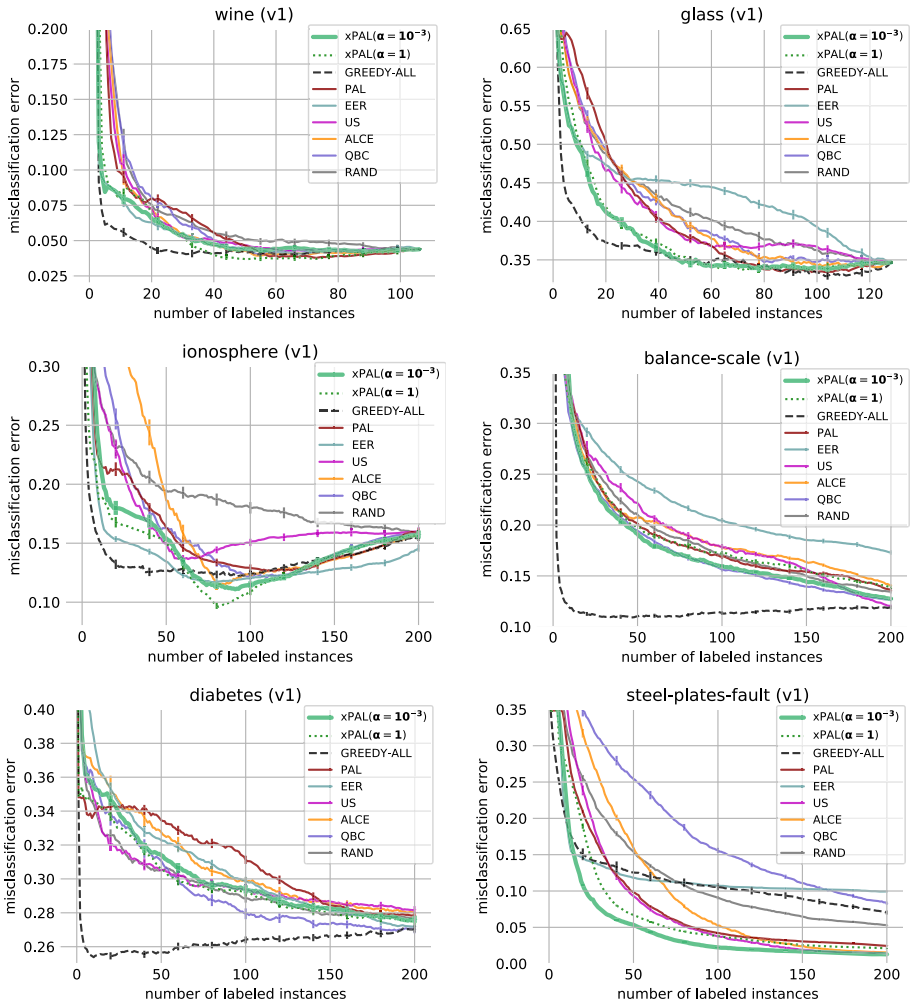


Fig. 3 Learning curves for six selected datasets. Each plot shows the misclassification error of xPAL and the competing algorithms w. r. t. the number of acquired labels. The learning curve that reaches a low error fast is considered best. The bars denote the standard error ($\frac{\sigma}{\sqrt{n}}$). The plots of the remaining 16 datasets are given in the supplemental material

6.3 Comparison between xPAL and competitors

We visualize our results using learning curves in Fig. 3 and rank statistics in Figs. 4, 5, and 6. More results are given in the appendix. The learning curves show the misclassification error (averaged over the 100 repetitions) on the test set after each label acquisition for every combination of an algorithm and a dataset. The learning curve that reaches a low error fast is considered best. For each curve, we show the standard error $\frac{\sigma}{\sqrt{N}}$ over all repetitions. Due to space limitations, we only show xPAL with $\alpha = 10^{-3}$ and $\alpha = 1$. In Sect. 6.4, we show that xPAL is superior for all tested priors and that the results do not change much.

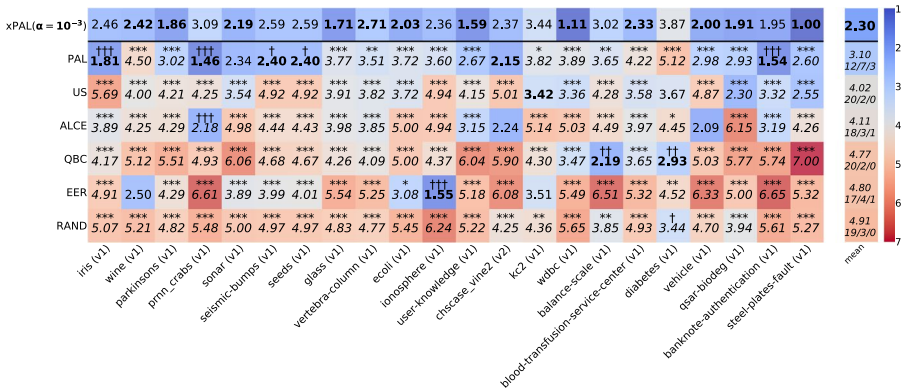


Fig. 4 The mean rank for all combinations of selection strategies and numerical datasets (RBF kernel) across 100 repetitions. The best strategy is printed in bold. Three stars (***) indicate significantly better results of xPAL with p value .001, two stars (**) indicate a p -value of .01 and one star (*) of .05. Analogously, significantly better performance of a competitor is shown with †

Fig. 5 The mean rank of selection strategies on datasets with categorical features using the hamming kernel

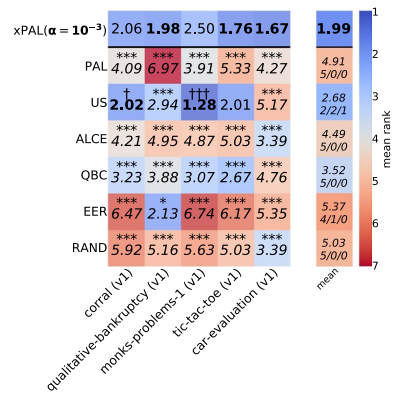
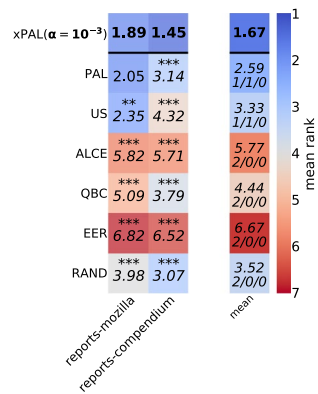


Fig. 6 The mean rank of selection strategies on text datasets using the cosine kernel



Almost all learning curves show that the supervised baseline (GREEDY-ALL) performs perfectly in an early phase. This is not surprising as it knows all labels (even from the

unlabeled set \mathcal{U}) to optimize the error on the training set. As seen in *steel-plates-fault*, this baseline does not achieve the best performance in all cases because of the greedy selection (no look-ahead). In that example, an optimal baseline would need to create a strategy for more than just the upcoming candidate. Also, the xPAL approach (green, bold line) with $\alpha = 10^{-3}$ performs well. For convenience, we plotted the xPAL also with $\alpha = 1$ as another alternative. The differences between both curves are rather small.

As it remains difficult to quantitatively assess the performance due to the large amount of datasets, we provide the **mean rank plot** in Figs. 4, 5, and 6. For this purpose, we calculated the rank of the area under the learning curve for each of the 100 repetitions and average this rank for every combination of a selection strategy and a dataset. As stated in Sect. 6.2, we calculated the performance for up to 200 labels. From this point, most learning curves do not change much anymore, as one can see in Fig. 3. Hence, the area under the learning curve provides a good overview of the whole learning process. To investigate the learning process at one specific point (e.g., at the beginning), the learning curves (Figs. 3, 10, 11 and 12) are more meaningful. We use color to visualize the performance: blue color means good rank, and red color indicates bad performance. The rank of the best algorithm is printed in bold. Moreover, we performed a Wilcoxon-signed-rank test to assess if the pairwise differences between xPAL and its competitors are significant. Three stars (***) indicate significantly better results of xPAL with a p value of .001, two stars (**) indicate a p value of 0.01 and one star (*) of .05. Analogously, significantly better performance of a competitor is shown with †. We yield the mean column (right) by averaging the ranks over all datasets. The pattern (a/b/c) in the second row of each cell summarizes a) the number of highly significant wins, c) the number of highly significant losses, and b) neither of both.

We separated the ranking plots w. r. t. the kernel function. Figure 4 shows results with the RBF kernel, Fig. 5 with the hamming-distance kernel, and Fig. 6 with the cosine similarity kernel. One can observe that xPAL has the lowest mean rank for all kernels and is always printed in blueish color across the datasets. No other algorithm performs as robust. The strongest competitor is PAL. But on the categorical data, we observe a clear performance difference between PAL and xPAL. One reason might be the difficulty of obtaining a reliable density estimation for categorical data.

6.4 Robustness of prior parameter

In Fig. 7, we show the mean ranking over all numerical datasets for different choices of priors α . Compared to the other strategies (left image), there is only a small difference across all choices. Comparing xPAL with $\alpha = 10^{-3}$ to the other priors (right image), we see that there are datasets where the selected xPAL is significantly outperformed, but in general, the effect is negligible. Also, all mean ranks are between 3.27 and 3.63, which validates the robustness of our parameter. We propose to use $\alpha = 10^{-3}$ as default.

6.5 Computation time

In Table 1, we already showed the theoretical time complexity. In this section, we now show the actual computation time which of course also depends on the efficiency of the implementation. Therefore, we artificially generated datasets with 500, 1000, ..., 2500

Fig. 7 The mean rank averaged over all numerical datasets for different parameters α . The 2nd row shows (wins/ties/losses) based on the results of the Wilcoxon-signed-rank test

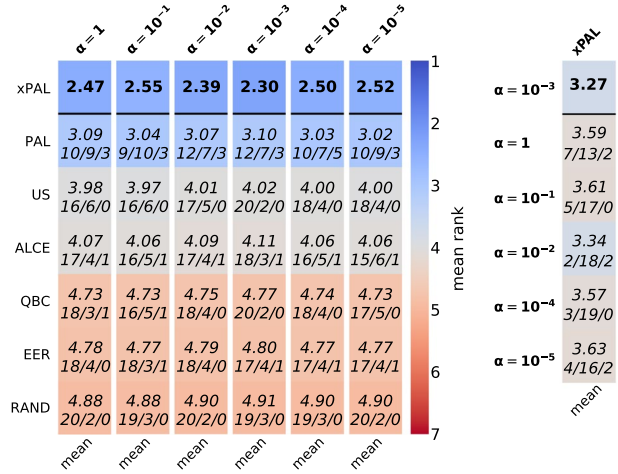
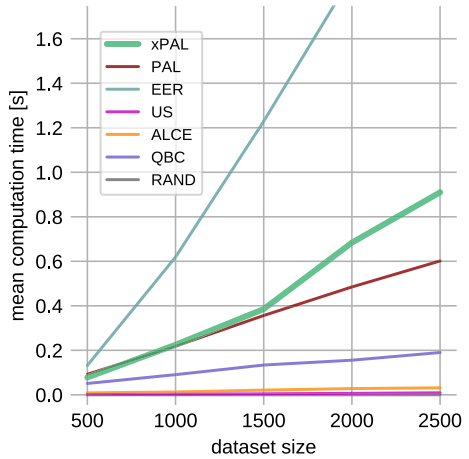


Fig. 8 Mean computation time per label acquisitions on artificial data (2, 4, 6 classes) with varying dataset size



instances and 2, 4, 6 classes. With every selection strategy, we acquired 200 labels and report the mean computation time on a personal computer in Fig. 8. For EER and xPAL, we clearly see the dependence on $|\mathcal{U}|$ and $|\mathcal{E}|$, respectively. As xPAL only needs to calculate the loss difference on instances, where the decision actually changes, we can reduce the computation time by a significant amount. Because of the inefficient optimization in PAL, we are even comparably fast to PAL for dataset with less than 1000 instances.

7 Conclusion

In this article, we moved toward optimal probabilistic AL by proposing xPAL. It is a decision-theoretic approach that determines the expected performance gain for labeling a candidate using a conjugate prior. We used this model to show the similarities and differences to the most related approaches and compared them by showing how each method selects

their instances in a synthetic example. Moreover, we provide an exhaustive experimental evaluation indicating the superiority of xPAL and the robustness of its prior parameter.

In future work, we aim to apply this idea to other cost-sensitive loss functions and for error-prone annotators as this is a current limitation of this article. Moreover, we research possibilities to use the concept of xPAL to define a stopping criterion and to apply it for other classifier types. The combination of xPAL with methods of deep learning is also promising. However, several challenges need to be addressed, such as unreliable estimates of the class probabilities and the estimation of the vector $\mathbf{k}_{x_c}^{\mathcal{L}}$. The former might be solvable by using techniques that improve the returned probabilities (e. g., by using Bayesian neural networks). The latter could be addressed by transforming samples into a latent representation (e. g., by using variational autoencoders). The resulting features would allow for a kernel density estimation. To extend this idea to regression problems, it will be necessary to combine the normally distributed output with a conjugate prior distribution (e. g., Gaussian-Wishart). This would allow for an analytic solution of the posterior which enables reliable estimation of the risk.

Appendix

Proof for Theorem 1

In Sect. 2, Roy and McCallum (2001) describe the algorithm: The estimate the expected loss from Eq. (4) using a Monte-Carlo approach over \mathcal{P} . They describe to use the unlabeled pool for that. In our work, we call this the candidate set \mathcal{U} . Their algorithm consists of 4 steps: In short, they calculate the *average expected loss* for every instance $x_c \in \mathcal{U}$. Therefor, they consider every possible label $y_c \in \mathcal{Y}$ and add the pair (x_c, y_c) to the training set \mathcal{D} (here: \mathcal{L}). They call the resulting set \mathcal{D}^* (here \mathcal{L}^+). The resulting expected losses are averaged, weighted with the respective posterior probability $p(y_c|x_c)$.

$$p(y_c|x_c) = \frac{(\mathbf{k}_{x_c}^{\mathcal{L}})_{y_c}}{\|\mathbf{k}_{x_c}^{\mathcal{L}}\|_1} \tag{28}$$

The posterior probabilities for our kernel-based classifier are determined using Eq. 28. Chapelle (2005) proposed to include a beta-prior and thereby extended the approach by Roy and McCallum (2001).

$$p(y_c|x_c) = \frac{(\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon)_{y_c}}{\|\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon\|_1} \tag{29}$$

The resulting equation can be simplified as follows:

$$\text{eer}(\mathbf{x}_c, \mathcal{L}, \mathcal{U}) = \sum_{y_c \in \mathcal{Y}} p(y_c|x_c, \mathcal{L}) \cdot \frac{1}{|\mathcal{U}|} \sum_{x \in \mathcal{U}} \left(1 - \max_{y \in \mathcal{Y}} p(y|x, \mathcal{L}^+) \right) \tag{30}$$

$$= \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon)_{y_c}}{\|\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon\|_1} \cdot \frac{1}{|\mathcal{U}|} \sum_{x \in \mathcal{U}} \left(1 - \max_{y \in \mathcal{Y}} \frac{(\mathbf{k}_{x_c}^{\mathcal{L}^+} + \epsilon)_y}{\|\mathbf{k}_{x_c}^{\mathcal{L}^+} + \epsilon\|_1} \right) \tag{31}$$

$$= \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon)_{y_c}}{\|\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon\|_1} \cdot \frac{1}{|\mathcal{U}|} \sum_{x \in \mathcal{U}} \left(\sum_{y \in \mathcal{Y}} \mathbf{1}\{y = f^{\mathcal{L}^+}(x)\} \left(1 - \frac{(\mathbf{k}_{x_c}^{\mathcal{L}^+} + \epsilon)_y}{\|\mathbf{k}_{x_c}^{\mathcal{L}^+} + \epsilon\|_1} \right) \right) \tag{32}$$

$$= \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon)_{y_c}}{\|\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon\|_1} \cdot \frac{1}{|\mathcal{U}|} \sum_{x \in \mathcal{U}} \left(\sum_{y \in \mathcal{Y}} \mathbf{1}\{y \neq f^{\mathcal{L}^+}(x)\} \cdot \frac{(\mathbf{k}_{x_c}^{\mathcal{L}^+} + \epsilon)_y}{\|\mathbf{k}_{x_c}^{\mathcal{L}^+} + \epsilon\|_1} \right) \tag{33}$$

$$= \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon)_{y_c}}{\|\mathbf{k}_{x_c}^{\mathcal{L}} + \epsilon\|_1} \cdot \frac{1}{|\mathcal{U}|} \sum_{x \in \mathcal{U}} \left(\sum_{y \in \mathcal{Y}} \frac{(\mathbf{k}_{x_c}^{\mathcal{L}^+} + \epsilon)_y}{\|\mathbf{k}_{x_c}^{\mathcal{L}^+} + \epsilon\|_1} \cdot L(y, f^{\mathcal{L}^+}(x)) \right) \tag{34}$$

□

Proof of Theorem 2

Multi-class probabilistic active learning (PAL) by Kottke et al. (2016) describes the expected gain in accuracy. Instead of evaluating this gain on a representative subset, they solely consider the gain locally. To proof Theorem 2, we need to set the m parameter of PAL to $m = 1$, which means that we only consider one possible label acquisition in each iteration. Kottke et al. (2016) model the hypothetical labels using a labeling vector $\mathbf{l} \in \mathbb{N}^C$ which describes the number of potentially added labels for each class. As we only consider one label at a time ($m = 1$), these vectors are unit vectors with a 1 at element of the considered class y_c and 0 otherwise. Hence, $l \in \{\mathbf{e}_1, \dots, \mathbf{e}_C\}$.

$$l_i = \begin{cases} 1 & i = y_c \\ 0 & \text{else} \end{cases} \tag{35}$$

For simplicity, we use \mathbf{k} instead of writing \mathbf{k}_{x_c} as PAL solely considers the candidate x_c and no other instance. Moreover, we know that $\mathbf{k}^{\mathcal{L}^+} = \mathbf{k} + \mathbf{e}_{y_c}$, as we increment the frequency estimate of the simulated class y_c by 1 (the similarity of x_c to x_c is always 1). Additionally to \mathbf{l} , Kottke et al. (2016) model the classifier’s decision using a vector \mathbf{d} , which is 1 for the class of the future decision and 0 otherwise.

$$d_i = \begin{cases} 1 & \arg \max_y (k_y^{\mathcal{L}^+}) = i \\ 0 & \text{else} \end{cases} \tag{36}$$

For simplicity, we do not write the iterators at sums and products if they iterate from $i = 1$ to C . Based on the old classifier $f^{\mathcal{L}}$ and the new classifier $f^{\mathcal{L}^+}$, we write $\hat{y} = f^{\mathcal{L}}(x_c)$ and $\hat{y}^+ = f^{\mathcal{L}^+}(x_c)$ for the old and the new prediction.

$$\begin{aligned} & \text{pal}(x_c, \mathcal{L}) \\ &= \hat{p}(x_c) \cdot \sum_{\mathbf{l}} \left(\underbrace{\left(\prod_{j=\sum(k_i+1)}^{(\sum(k_i+l_i+d_i+1)-1)} \frac{1}{j} \right)}_{=I} \cdot \underbrace{\prod_{i=1}^C \left(\prod_{j=k_i+1}^{k_i+l_i+d_i} j \right)}_{=II} \cdot \underbrace{\frac{\Gamma((\sum l_i) + 1)}{\prod(\Gamma(l_i + 1))}}_{=III} \right) - \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1} \end{aligned} \tag{37}$$

$$I = \left(\prod_{j=\sum(k_i+1)}^{\sum(k_i+l_i+d_i+1)-1} \frac{1}{j} \right) = \left(\prod_{j=0}^1 \frac{1}{\sum(k_i+1)+j} \right) = \frac{1}{\sum(k_i+1)} \cdot \frac{1}{\sum(k_i+1)+1} \tag{38}$$

$$= \frac{1}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \tag{39}$$

$$\Pi = \prod_{i=1}^C \left(\prod_{j=k_i+1}^{k_i+l_i+d_i} j \right) = \prod_{i=1}^C \left(\prod_{j=1}^{l_i+d_i} k_i + j \right) = \prod_{i=1}^C \begin{cases} 1 & l_i + d_i = 0 \\ (k_i + 1) & l_i + d_i = 1 \\ (k_i + 1)(k_i + 2) & l_i + d_i = 2 \end{cases} \tag{40}$$

$$= \prod_{i=1}^C \begin{cases} (k_i + 1) & l_i + d_i = 1 \\ (k_i + 1)(k_i + 2) & l_i + d_i = 2 \end{cases} = \begin{cases} (\mathbf{k} + \mathbf{1})_{y_c} \cdot (\mathbf{k} + \mathbf{2})_{y_c} & \hat{y}^+ = y_c \\ (\mathbf{k} + \mathbf{1})_{y_c} \cdot (\mathbf{k} + \mathbf{1})_{\hat{y}^+} & \text{else} \end{cases} \tag{41}$$

$$\text{III} = \frac{\Gamma(\sum l_i + 1)}{\prod(\Gamma(l_i + 1))} = \frac{\Gamma(2)}{1} = 1 \tag{42}$$

We now insert I, II, III back into Eq. 37.

$$\begin{aligned} \text{pal}(\mathbf{x}_c, \mathcal{L}) &= \hat{p}(\mathbf{x}_c) \cdot \sum_{\mathcal{Y}} \frac{1}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot \begin{cases} (\mathbf{k} + \mathbf{1})_{y_c} \cdot (\mathbf{k} + \mathbf{2})_{y_c} & \hat{y}^+ = y_c \\ (\mathbf{k} + \mathbf{1})_{y_c} \cdot (\mathbf{k} + \mathbf{1})_{\hat{y}^+} & \text{else} \end{cases} - \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1} \end{aligned} \tag{43}$$

$$= \hat{p}(\mathbf{x}_c) \cdot \sum_{y_c \in \mathcal{Y}} \frac{1}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot \begin{cases} (\mathbf{k} + \mathbf{1})_{y_c} \cdot (\mathbf{k} + \mathbf{2})_{y_c} & \hat{y}^+ = y_c \\ (\mathbf{k} + \mathbf{1})_{y_c} \cdot (\mathbf{k} + \mathbf{1})_{\hat{y}^+} & \text{else} \end{cases} - \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1} \tag{44}$$

$$= \hat{p}(\mathbf{x}_c) \cdot \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot \begin{cases} (\mathbf{k} + \mathbf{2})_{y_c} & \hat{y}^+ = y_c \\ (\mathbf{k} + \mathbf{1})_{\hat{y}^+} & \text{else} \end{cases} - \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1} \tag{45}$$

We divide the sum into two parts: (A) The subset of all labels (\mathcal{Y}_{\neq}) that change the decision, (B) and the labels ($\mathcal{Y}_{=}$) that do not change the decision. Please remember that a new label y_c could change the decision of \hat{y}^+ as it includes the new label. They are defined as follows:

$$\mathcal{Y} = \mathcal{Y}_{\neq} \dot{\cup} \mathcal{Y}_{=} = \{y_c \in \mathcal{Y} : \hat{y} \neq \hat{y}^+\} \dot{\cup} \{y_c \in \mathcal{Y} : \hat{y} = \hat{y}^+\} \tag{46}$$

Now, we consider both cases independently.

A) Labels that change the decision For all $y_c \in \mathcal{Y}$ with $\hat{y} \neq \hat{y}^+$, we know that $\hat{y}^+ = y_c$.

It follows that $L(y_c, \hat{y}^+) - L(y, \hat{y}) = -1$.

$$\sum_{y_c \in \mathcal{Y}_{\neq}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot \begin{cases} (\mathbf{k} + \mathbf{2})_{y_c} & \hat{y}^+ = y_c \\ (\mathbf{k} + \mathbf{1})_{\hat{y}} & \text{else} \end{cases} \tag{47}$$

$$= \sum_{y_c \in \mathcal{Y}_{\neq}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1} \cdot \frac{(\mathbf{k} + \mathbf{2})_{y_c}}{\|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1}$$

$$= \sum_{y_c \in \mathcal{Y}_{\neq}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1} \cdot \frac{(\mathbf{k}^{\mathcal{L}^+} + \mathbf{1})_{y_c}}{\|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \tag{48}$$

$$= - \sum_{y_c \in \mathcal{Y}_{\neq}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1} \sum_{y \in \{y_c\}} \frac{(\mathbf{k}^{\mathcal{L}^+} + \mathbf{1})_y}{\|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot (L(y, \hat{y}^+) - L(y, \hat{y}))$$

B) Labels that do not change the decision Here, we can use the following implications to rewrite the cases from Eq. 45 into the sum:

- $y_c = \hat{y} \Rightarrow \hat{y}^+ = y_c$
- $y_c \neq \hat{y} \Rightarrow \hat{y}^+ \neq y_c$

$$\sum_{y_c \in \mathcal{Y}_{=}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot \begin{cases} (\mathbf{k} + \mathbf{2})_{y_c} & \hat{y}^+ = y_c \\ (\mathbf{k} + \mathbf{1})_{\hat{y}} & \text{else} \end{cases} - \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1} \tag{49}$$

$$= \left(\sum_{y_c \in \mathcal{Y}_{= \setminus \{\hat{y}\}}} \frac{(\mathbf{k} + \mathbf{1})_{y_c} \cdot (\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \right) + \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}} \cdot (\mathbf{k} + \mathbf{2})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} - \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1} \tag{50}$$

$$= \left(\sum_{y_c \in \mathcal{Y}_{= \setminus \{\hat{y}\}}} \frac{(\mathbf{k} + \mathbf{1})_{y_c} \cdot (\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \right) + \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}} \cdot (\mathbf{k} + \mathbf{2})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \tag{51}$$

$$- \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}} \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1}$$

$$= \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \left(\sum_{y_c \in \mathcal{Y}_{= \setminus \{\hat{y}\}}} (\mathbf{k} + \mathbf{1})_{y_c} \right) + (\mathbf{k} + \mathbf{2})_{\hat{y}} - (\|\mathbf{k} + \mathbf{1}\|_1 + 1) \tag{52}$$

$$= \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \left(\sum_{y_c \in \mathcal{Y}_{= \setminus \{\hat{y}\}}} (\mathbf{k} + \mathbf{1})_{y_c} \right) + (\mathbf{k} + \mathbf{1})_{\hat{y}} - \|\mathbf{k} + \mathbf{1}\|_1 \tag{53}$$

$$= \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k} + \mathbf{1}\|_1 \cdot \|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \left(\sum_{y_c \in \mathcal{Y}_{=}} (\mathbf{k} + \mathbf{1})_{y_c} \right) - \left(\sum_{y_c \in \mathcal{Y}} (\mathbf{k} + \mathbf{1})_{y_c} \right) \tag{54}$$

$$= - \sum_{y_c \in \mathcal{Y}_{\neq}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1} \cdot \frac{(\mathbf{k} + \mathbf{1})_{\hat{y}}}{\|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \tag{55}$$

$$= - \sum_{y_c \in \mathcal{Y}_{\neq}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1} \sum_{y \in \{\hat{y}\}} \frac{(\mathbf{k}^{\mathcal{L}^+} + \mathbf{1})_y}{\|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot (L(y, \hat{y}^+) - L(y, \hat{y})) \tag{56}$$

In the last step, we use that $\hat{y} \neq \hat{y}^+ \implies L(\hat{y}, \hat{y}^+) - L(\hat{y}, \hat{y}) = 1$. Additionally, we use that $y_c \neq \hat{y}$ applies and thus $k_{\hat{y}} = k_{\hat{y}}^{\mathcal{L}^+}$. Next, we combine both cases:

$$\text{pal}(\mathbf{x}_c, \mathcal{L}) = -\hat{p}(\mathbf{x}_c) \cdot \left(\sum_{y_c \in \mathcal{Y}_{\neq}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1} \sum_{y \in \{y_c\}} \frac{(\mathbf{k}^{\mathcal{L}^+} + \mathbf{1})_y}{\|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot (L(y, \hat{y}^+) - L(y, \hat{y})) \right) \tag{57}$$

$$+ \sum_{y_c \in \mathcal{Y}_{\neq}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1} \sum_{y \in \{\hat{y}\}} \frac{(\mathbf{k}^{\mathcal{L}^+} + \mathbf{1})_y}{\|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot (L(y, \hat{y}^+) - L(y, \hat{y})) \tag{58}$$

$$= -\hat{p}(\mathbf{x}_c) \cdot \sum_{y_c \in \mathcal{Y}_{\neq}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1} \sum_{y \in \{y_c, \hat{y}\}} \frac{(\mathbf{k}^{\mathcal{L}^+} + \mathbf{1})_y}{\|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot (L(y, \hat{y}^+) - L(y, \hat{y})) \tag{59}$$

Because of $L(y, \hat{y}^+) - L(y, \hat{y}) = 0$ for $y \notin \{y_c, \hat{y}\}$ and for $y_c \in \mathcal{Y}_{=}$, we can change this equation to

$$\text{pal}(\mathbf{x}_c, \mathcal{L}) = -\hat{p}(\mathbf{x}_c) \cdot \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k} + \mathbf{1})_{y_c}}{\|\mathbf{k} + \mathbf{1}\|_1} \sum_{y \in \mathcal{Y}} \frac{(\mathbf{k}^{\mathcal{L}^+} + \mathbf{1})_y}{\|\mathbf{k}^{\mathcal{L}^+} + \mathbf{1}\|_1} \cdot (L(y, \hat{y}^+) - L(y, \hat{y})). \tag{60}$$

□

Proof of Theorem 3

According to Settles (2009), the usefulness score for “least confidence uncertainty sampling” is determined by the following equation and can easily be rewritten. We denote: $\hat{y} = f^{\mathcal{L}}(\mathbf{x}_c)$.

$$\text{us}(\mathbf{x}_c, \mathcal{L}) = 1 - p(\hat{y}|\mathbf{x}_c) = \sum_{y_c \in \mathcal{Y}} \mathbb{1}_{y_c = f^{\mathcal{L}}(\mathbf{x}_c)} (1 - p(y_c|\mathbf{x}_c)) \tag{61}$$

$$= \sum_{y_c \in \mathcal{Y}} \mathbb{1}_{y_c = f^{\mathcal{L}}(\mathbf{x}_c)} \left(1 - \frac{(\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{0})_{y_c}}{\|\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{0}\|_1} \right) \tag{62}$$

$$= \sum_{y_c \in \mathcal{Y}} \mathbb{1}_{y_c \neq f^{\mathcal{L}}(\mathbf{x}_c)} \left(\frac{(\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{0})_{y_c}}{\|\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{0}\|_1} \right) = \sum_{y_c \in \mathcal{Y}} \frac{(\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{0})_{y_c}}{\|\mathbf{k}_{\mathbf{x}_c}^{\mathcal{L}} + \mathbf{0}\|_1} \cdot L(y_c, f^{\mathcal{L}}(\mathbf{x}_c)) \tag{63}$$

□

Table 2 Description of datasets

openML	name	inst.	feat.	class distribution
61	iris	150	4	[0.33, 0.33, 0.33]
187	wine	178	13	[0.33, 0.4, 0.27]
1488	parkinsons	195	22	[0.25, 0.75]
446	prnn_crabs	200	7	[0.5, 0.5]
40	sonar	208	60	[0.53, 0.47]
1500	seismic-bumps	210	7	[0.33, 0.33, 0.33]
1499	seeds	210	7	[0.33, 0.33, 0.33]
41	glass	214	9	[0.33, 0.36, 0.06, 0.14, 0.04, 0.08]
1523	vertebra-column	310	6	[0.19, 0.32, 0.48]
39	ecoli	336	7	[0.43, 0.23, 0.01, 0.01, 0.1, 0.06, 0.01, 0.15]
59	ionosphere	351	34	[0.36, 0.64]
1508	user-knowledge	403	5	[0.25, 0.32, 0.3, 0.06, 0.06]
814	chscase_vine2 (v2)	468	2	[0.45, 0.55]
1063	kc2	522	21	[0.8, 0.2]
1510	wdbc	569	30	[0.63, 0.37]
11	balance-scale	625	4	[0.08, 0.46, 0.46]
1464	blood-transfusion-s.	748	4	[0.76, 0.24]
37	diabetes	768	8	[0.65, 0.35]
54	vehicle	846	18	[0.26, 0.25, 0.26, 0.24]
1494	qsar-biodeg	1055	41	[0.66, 0.34]
1462	banknote-auth.	1372	4	[0.56, 0.44]
1504	steel-plates-fault	1941	33	[0.65, 0.35]
40669	corral	160	6	[0.56, 0.44]
1495	bankruptcy	250	6	[0.43, 0.57]
333	monks	556	6	[0.5, 0.5]
50	tic	958	9	[0.35, 0.65]
40664	car	1728	21	[0.7, 0.22, 0.04, 0.04]
-	reports-mozilla	675	100	[0.23, 0.09, 0.43, 0.25]
-	reports-compendium	962	56	[0.09, 0.33, 0.23, 0.35]

Description of datasets

A detailed description of the datasets is available in Tab. 2. We provide the openML identifier⁷, the dataset's name, the number of instances and features, and the distribution of classes (the list describes the fraction of class 1 in the first element, the fraction of class 2 in the second element, etc).

⁷ <https://www.openml.org/>.

More experimental results

In this section, we provide more plots from our experimental evaluation. Please refer to the original paper for the detailed explanation of the experimental setup and the discussion of the results.

Usefulness plots with randomly selected labels

See Fig. 9.

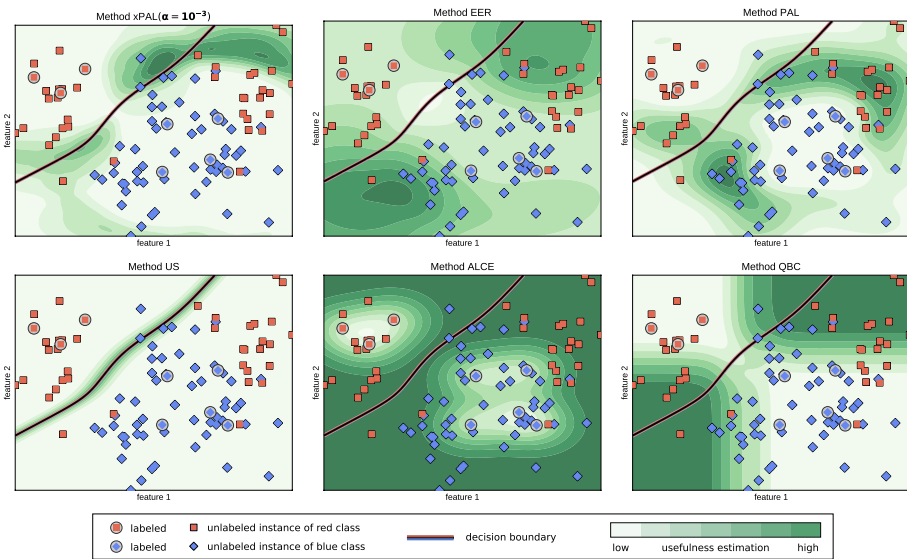


Fig. 9 Visualization of acquisition behavior for different selection strategies. The green color indicates how useful a selection strategy considers a region. The usefulness depends on the selection criterion of the strategy. The eight labels have been randomly selected and are similar for all strategies to emphasize how different selection strategies assess the usefulness of different regions

Learning curves

See Figs. 10, 11 and 12.

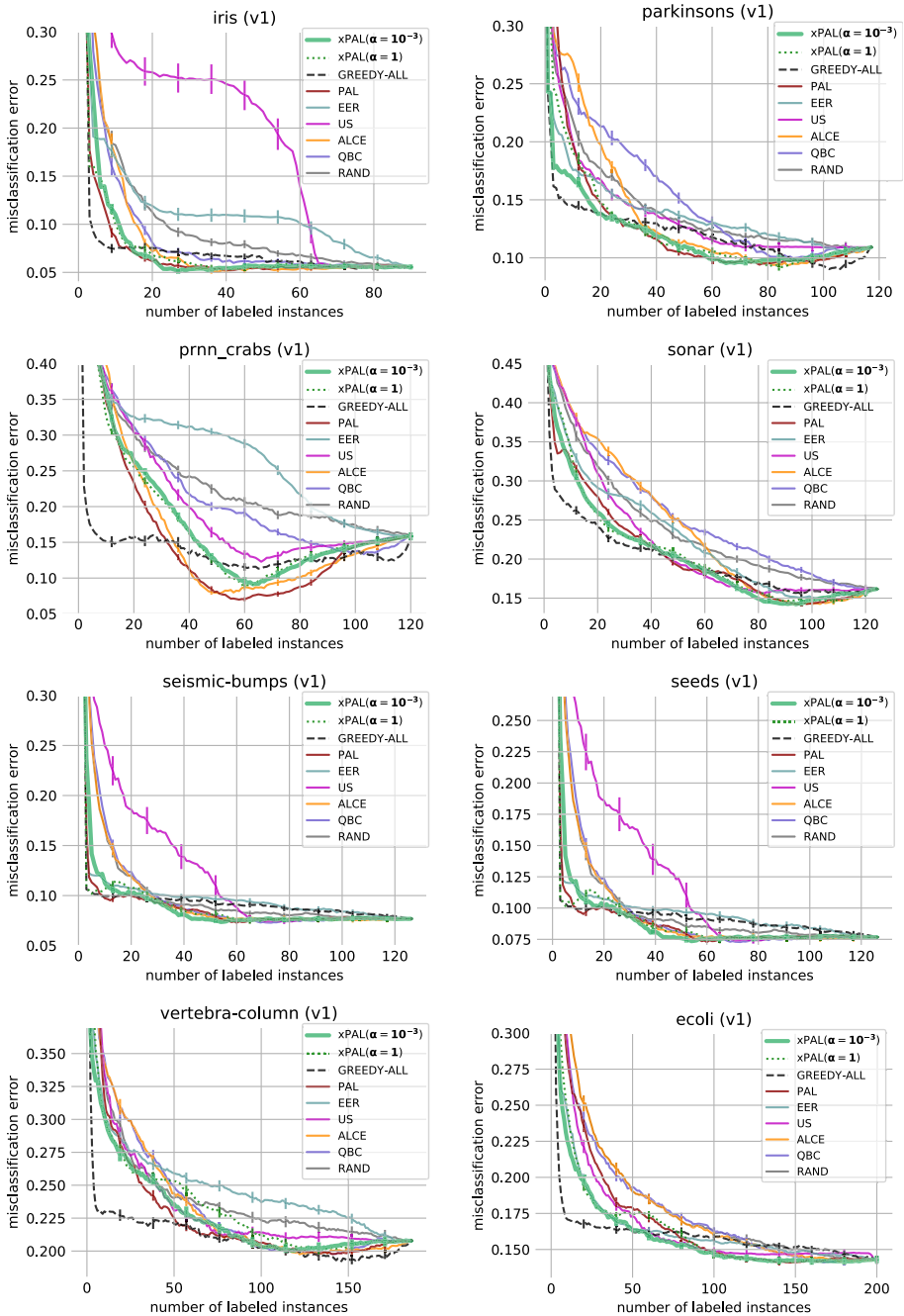


Fig. 10 Mean accuracy learning curves comparing xPAL to its competitors. High values and fast convergence is considered best. The bars denote the standard error ($\frac{\sigma}{\sqrt{n}}$)

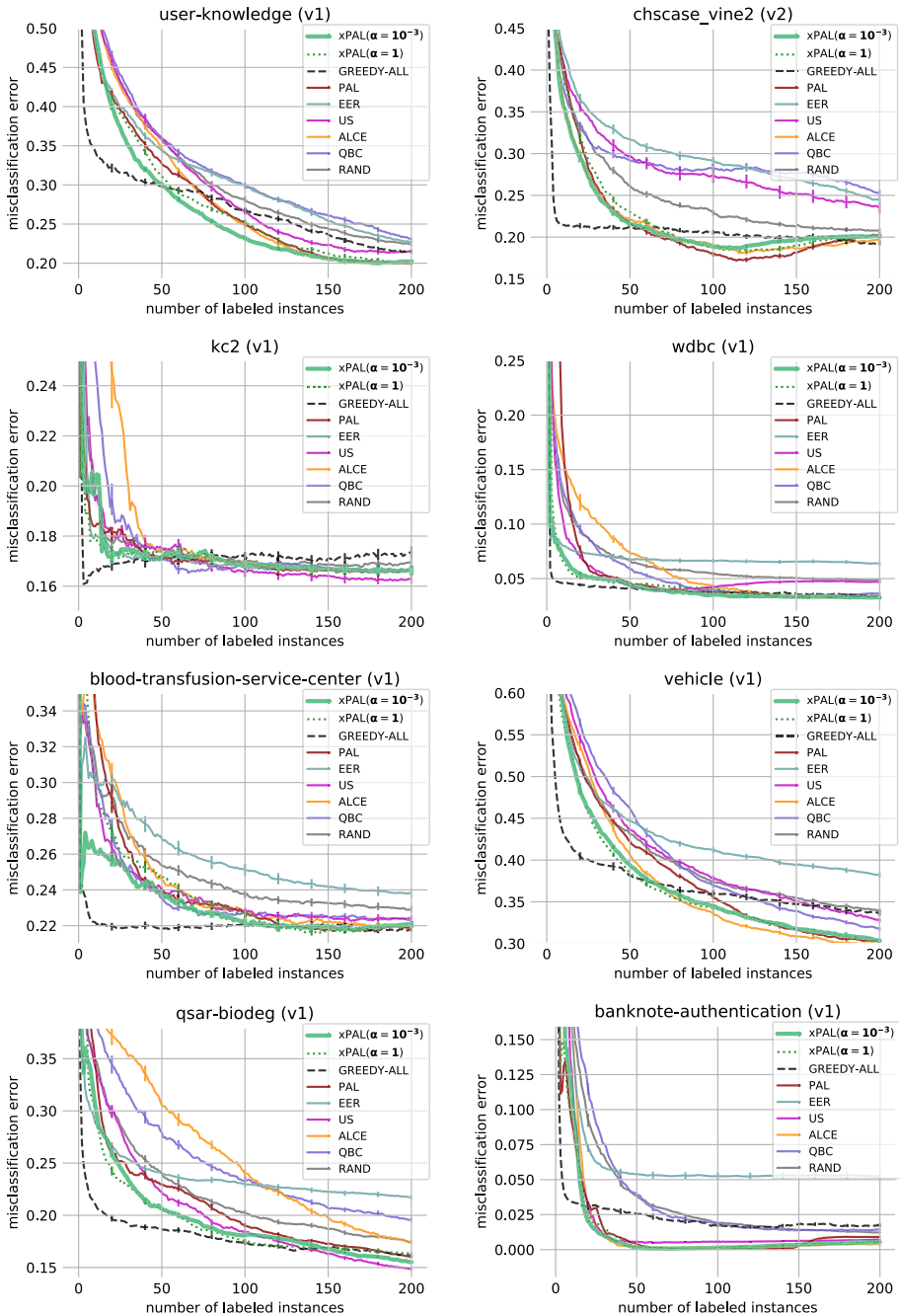


Fig. 11 Mean accuracy learning curves comparing xPAL to its competitors. High values and fast convergence is considered best. The bars denote the standard error ($\frac{\sigma}{\sqrt{n}}$)

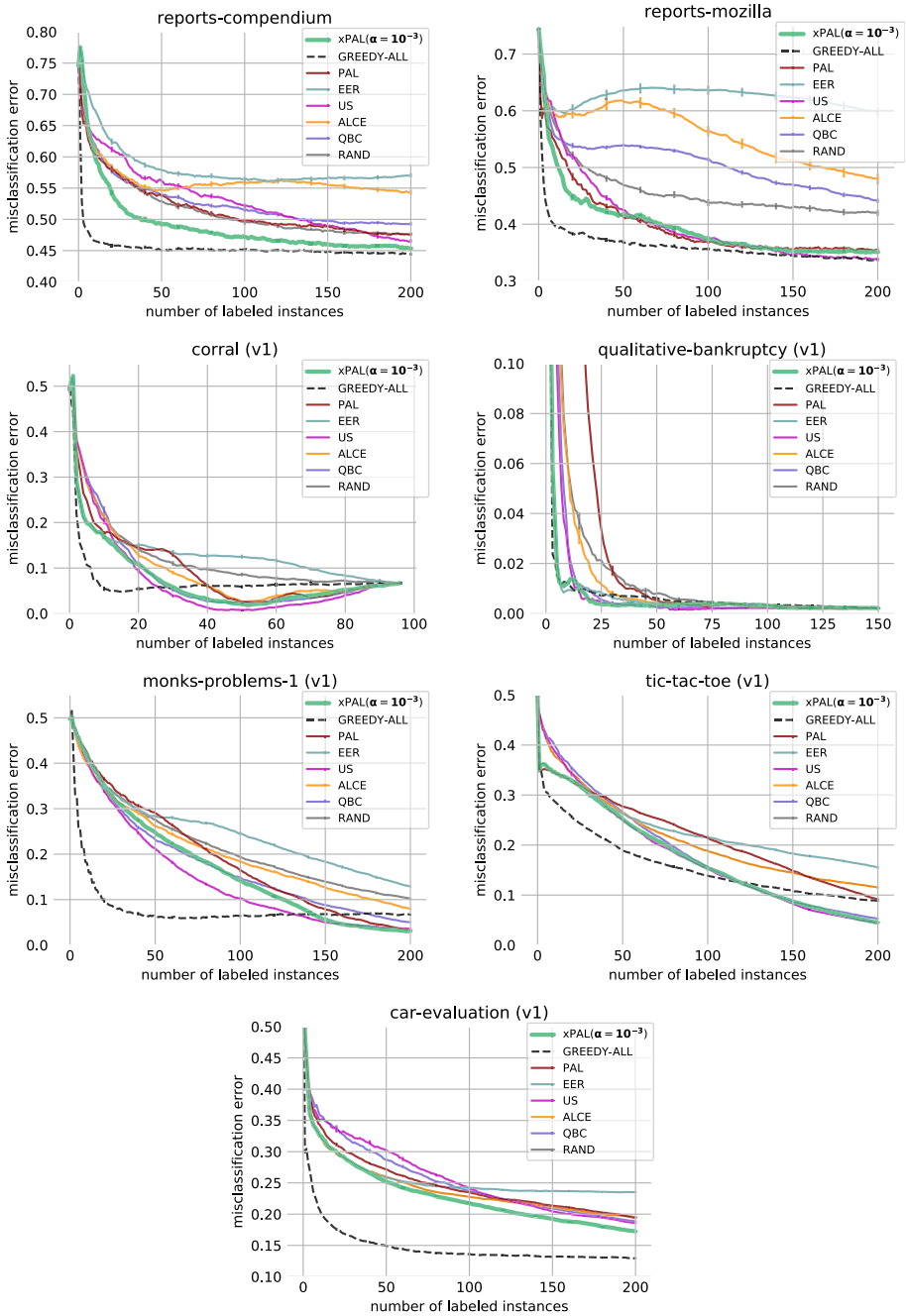


Fig. 12 Mean accuracy learning curves comparing xPAL to its competitors. High values and fast convergence is considered best. The bars denote the standard error ($\frac{\sigma}{\sqrt{n}}$)

Table 3 Averaged area under the learning curve performances including standard deviations

	xPAL(10 ⁻³)	PAL	US	ALCE	QBC	EER	RAND
iris (v1)	.084 (±0.022)	.080 (±0.022)††	.199 (±0.096)***	.096 (±0.024)***	.099 (±0.024)***	.123 (±0.050)***	.113 (±0.029)***
wine (v1)	.067 (±0.017)	.079 (±0.019)***	.077 (±0.020)***	.077 (±0.017)***	.082 (±0.017)***	.066 (±0.019)	.084 (±0.022)***
parkinsons (v1)	.122 (±0.028)	.131 (±0.025)***	.141 (±0.032)***	.141 (±0.025)***	.156 (±0.030)***	.140 (±0.030)***	.147 (±0.027)***
prnm_crabs (v1)	.184 (±0.038)	.162 (±0.030)†††	.207 (±0.042)***	.170 (±0.030)†††	.220 (±0.034)***	.270 (±0.047)***	.237 (±0.040)***
sonar (v1)	.206 (±0.031)	.207 (±0.029)	.222 (±0.028)***	.239 (±0.028)***	.255 (±0.029)***	.227 (±0.040)***	.240 (±0.033)***
seeds (v1)	.097 (±0.021)	.096 (±0.021)†	.133 (±0.053)***	.107 (±0.023)***	.108 (±0.023)***	.106 (±0.026)***	.111 (±0.022)***
seismic-bumps (v1)	.097 (±0.021)	.096 (±0.021)†	.133 (±0.053)***	.107 (±0.023)***	.108 (±0.023)***	.106 (±0.026)***	.111 (±0.022)***
glass (v1)	.378 (±0.037)	.406 (±0.036)***	.411 (±0.041)***	.411 (±0.037)***	.414 (±0.038)***	.437 (±0.042)***	.423 (±0.041)***
vertebra-column (v1)	.231 (±0.031)	.235 (±0.032)**	.239 (±0.033)***	.239 (±0.031)***	.240 (±0.029)***	.252 (±0.033)***	.246 (±0.033)***
ecoli (v1)	.166 (±0.024)	.177 (±0.022)***	.178 (±0.023)***	.189 (±0.026)***	.187 (±0.021)***	.172 (±0.025)*	.191 (±0.026)***
ionosphere (v1)	.152 (±0.028)	.160 (±0.029)***	.172 (±0.028)***	.180 (±0.035)***	.168 (±0.027)***	.140 (±0.023)†††	.194 (±0.036)***
user-knowledge (v1)	.273 (±0.021)	.286 (±0.024)***	.307 (±0.032)***	.294 (±0.021)***	.330 (±0.024)***	.317 (±0.026)***	.317 (±0.027)***
chcscase_vine2 (v2)	.223 (±0.021)	.221 (±0.021)	.287 (±0.063)***	.221 (±0.023)	.294 (±0.030)***	.303 (±0.039)***	.253 (±0.023)***
kc2 (v1)	.174 (±0.018)	.173 (±0.017)*	.174 (±0.019)	.185 (±0.024)***	.179 (±0.019)***	.173 (±0.019)	.177 (±0.019)***
wdbc (v1)	.045 (±0.009)	.058 (±0.009)***	.058 (±0.011)***	.066 (±0.014)***	.059 (±0.012)***	.072 (±0.019)***	.069 (±0.014)***
balance-scale (v1)	.188 (±0.017)	.194 (±0.016)**	.199 (±0.017)***	.200 (±0.015)***	.182 (±0.016)††	.229 (±0.022)***	.195 (±0.018)**
blood-transfusion-s. (v1)	.231 (±0.017)	.245 (±0.021)***	.239 (±0.017)***	.244 (±0.022)***	.239 (±0.018)***	.259 (±0.029)***	.250 (±0.019)***
diabetes (v1)	.303 (±0.020)	.311 (±0.018)***	.301 (±0.021)	.310 (±0.026)*	.296 (±0.030)††	.309 (±0.021)**	.298 (±0.021)†
vehicle (v1)	.375 (±0.018)	.387 (±0.016)***	.412 (±0.026)***	.378 (±0.018)	.412 (±0.019)***	.436 (±0.026)***	.409 (±0.023)***
qsar-biodeg (v1)	.198 (±0.016)	.214 (±0.014)***	.206 (±0.019)***	.261 (±0.033)***	.254 (±0.030)***	.239 (±0.025)***	.224 (±0.021)***
banknote-auth. (v1)	.019 (±0.003)	.018 (±0.002)†††	.025 (±0.005)***	.024 (±0.005)***	.048 (±0.010)***	.070 (±0.022)***	.046 (±0.010)***
steel-plates-fault (v1)	.056 (±0.005)	.084 (±0.008)***	.085 (±0.017)***	.113 (±0.013)***	.192 (±0.030)***	.127 (±0.013)***	.128 (±0.016)***
corral (v1)	.079 (±0.023)	.098 (±0.028)***	.076 (±0.020)†	.098 (±0.027)***	.087 (±0.022)***	.131 (±0.032)***	.121 (±0.032)***
qualitative-bankruptcy (v1)	.012 (±0.004)	.048 (±0.012)***	.016 (±0.006)***	.021 (±0.007)***	.017 (±0.006)***	.013 (±0.005)*	.023 (±0.009)***
monks-problems-1 (v1)	.169 (±0.014)	.190 (±0.020)***	.148 (±0.019)†††	.206 (±0.022)***	.178 (±0.019)***	.248 (±0.026)***	.218 (±0.022)***
tic-tac-toe (v1)	.175 (±0.012)	.217 (±0.019)***	.177 (±0.014)	.213 (±0.016)***	.186 (±0.015)***	.231 (±0.023)***	.213 (±0.016)***
car-evaluation (v1)	.232 (±0.010)	.251 (±0.013)***	.258 (±0.016)***	.243 (±0.014)***	.256 (±0.017)***	.258 (±0.014)***	.243 (±0.014)***

Table 3 (continued)

	xPAL(10^{-3})	PAL	US	ALCE	QBC	EER	RAND
reports-mozilla	.397 (± 0.027)	.399 (± 0.026)	.404 (± 0.020)**	.559 (± 0.068)***	.508 (± 0.030)***	.625 (± 0.070)***	.461 (± 0.043)***
reports-compendium	.490 (± 0.021)	.517 (± 0.019)***	.532 (± 0.022)***	.563 (± 0.032)***	.528 (± 0.022)***	.583 (± 0.030)***	.516 (± 0.023)***

Low values are considered best

Three stars (***) indicate significantly better results of xPAL with p value .001, two stars (**) indicate a p value of 0.01 and one star (*) of .05. Analogously, significantly better performance of a competitor is shown with †

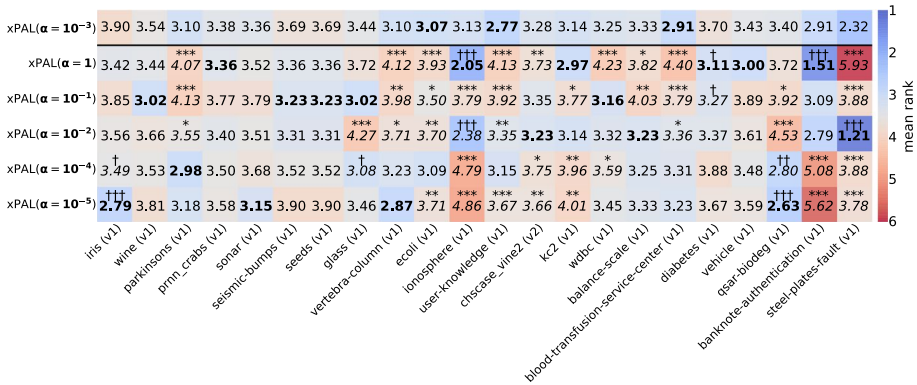


Fig. 13 The mean rank for xPAL with different parameters and datasets across 100 repetitions. The best parameter is printed in bold. The Wilcoxon signed rank test shows pairwise significance between xPAL with $\alpha = 10^{-3}$ and its competitor

Area under the learning curve

Table 3 describes the averaged area under the learning curve including standard deviations and significance testing with the Wilcoxon signed rank test. The notation is similar to the one from the paper.

Detailed ranking plots for different parameters

Figure 13 is the detailed version of Fig. 7 (right) in the original paper.

Execution times and computing infrastructure

Table 4 provides an overview of the execution times over the different selection strategies and datasets. The execution times are averaged over 100 repeated runs each with a maximum number of 200 instance selections. A single execution time entry indicates the average time in seconds to select a single instance for a given dataset and selection strategy. The execution times are primarily depended on the number of instances but also on aspects like the number of features and classes as calculations might become more complex.

All experiments were run on an heterogeneous computer cluster which might lead to irregular results as the speed between the cluster nodes vary.

Table 4 Execution times in seconds for one single instance averaged over all repetitions and acquisitions

dataset	instances	xPAL ($\alpha = 10^{-3}$)	PAL	OPT	US	ALCE	QBC	EER	RAND
iris (v1)	150	0.0763	0.0089	0.0124	0.0009	0.0034	0.0329	0.0961	0.0001
wine (v1)	178	0.0824	0.0105	0.0160	0.0010	0.0038	0.0349	0.1184	0.0001
parkinsons (v1)	195	0.0726	0.0101	0.0234	0.0020	0.0029	0.0354	0.0994	0.0002
prnn_crabs (v1)	200	0.0717	0.0063	0.0193	0.0010	0.0030	0.0349	0.0919	0.0001
sonar (v1)	208	0.0865	0.0137	0.0459	0.0018	0.0032	0.0365	0.1631	0.0002
seismic-bumps (v1)	210	0.0941	0.0113	0.0179	0.0012	0.0042	0.0342	0.0305	0.0001
seeds (v1)	210	0.0991	0.0126	0.0186	0.0010	0.0041	0.0352	0.1459	0.0001
glass (v1)	214	0.3196	0.0616	0.0193	0.0011	0.0067	0.0355	0.2980	0.0001
vertebra-col- umn (v1)	310	0.1737	0.0189	0.0308	0.0013	0.0056	0.0369	0.2515	0.0001
ecoli (v1)	336	0.5243	0.2307	0.0466	0.0028	0.0064	0.0388	0.8007	0.0002
ionosphere (v1)	351	0.2497	0.0525	0.1307	0.0095	0.0046	0.0404	0.7586	0.0002
user-knowl- edge (v1)	403	0.4468	0.0858	0.0524	0.0015	0.0094	0.0425	0.7438	0.0001
chscase_vine2 (v2)	468	0.2468	0.0182	0.0730	0.0017	0.0046	0.0457	0.4156	0.0001
kc2 (v1)	522	0.4760	0.1127	0.2549	0.0173	0.0053	0.0491	2.9201	0.0003
wdbc (v1)	569	0.4808	0.1655	0.3257	0.0258	0.0054	0.0523	4.8052	0.0005
balance-scale (v1)	625	0.6702	0.0664	0.1512	0.0045	0.0074	0.0531	0.4069	0.0003
blood-transfu- sion-s. (v1)	748	0.8058	0.0801	0.3079	0.0128	0.0056	0.0574	1.3400	0.0006
diabetes (v1)	768	0.8174	0.1688	0.4503	0.0287	0.0059	0.0594	8.3075	0.0009
vehicle (v1)	846	2.2947	0.5192	1.9738	0.0384	0.0122	0.0664	35.5409	0.0010
qsar-biodeg (v1)	1055	1.9238	0.3374	1.2020	0.0712	0.0077	0.0789	38.2082	0.0016
banknote-auth. (v1)	1372	1.9130	0.3116	5.4191	0.0996	0.0085	0.0868	69.2271	0.0023
steel-plates- fault (v1)	1941	5.8124	0.6423	26.3854	0.2050	0.0118	0.1242	38.3739	0.0022

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Baram, Y., Yaniv, R. E., & Luz, K. (2004). Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5, 255–291.
- Beyer, C., Krempel, G., & Lemaire, V. (2015). How to select information that matters: A comparative study on active learning strategies for classification. In *Proceedings of the 15th international conference on knowledge technologies and data-driven business, association for computing machinery, i-KNOW '15*, New York, NY, USA.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
- Bondu, A., Lemaire, V., & Boullé, M. (2010). Exploration vs. exploitation in active learning: A Bayesian approach. In *International joint conference on neural networks (IJCNN)* (pp. 1–7). IEEE.
- Brinker, K. (2003). Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th international conference on machine learning (ICML)* (pp. 59–66).
- Calma, A., Reitmaier, T., & Sick, B. (2018). Semi-supervised active learning for support vector machines: A novel approach that exploits structure information in data. *Information Sciences*, 456, 13–33.
- Chapelle, O. (2005). Active learning for parzen window classifier. In *Proceedings of the 10th international workshop on artificial intelligence and statistics (AISTATS)* (Vol. 5, pp. 49–56).
- Chaudhuri, A., Kakde, D., Sadek, C., Gonzalez, L., & Kong, S. (2017). The mean and median criteria for kernel bandwidth selection for support vector data description. In *International conference on data mining workshops (ICDMW)* (pp. 842–849). IEEE.
- Cuong, N. V., Lee, W. S., & Ye, N. (2014). Near-optimal adaptive pool-based active learning with general loss. In *Proceedings of the 30th conference on uncertainty in artificial intelligence (UAI)* (pp. 122–131).
- Dasgupta, S. (2009). The two faces of active learning. In *International conference on discovery science* (pp. 35–35). Springer.
- Donmez, P., Carbonell, J. G., & Bennett, P. N. (2007). Dual strategy active learning. In *Proceedings of the European conference on machine learning (ECML)* (pp. 116–127). Springer.
- Golovin, D., & Krause, A. (2010). Adaptive submodularity: A new approach to active learning and stochastic optimization. In *Proceedings of the 23rd conference on algorithmic learning theory (ALT)* (pp. 333–345).
- Guillory, A., & Bilmes, J. (2010). Interactive submodular set cover. In *Proceedings of the 27th International conference on machine learning (ICML)*.
- Hernández-González, J., Rodríguez, D., Inza, I., Harrison, R., & Lozano, J. A. (2018). Two datasets of defect reports labeled by a crowd of annotators of unknown reliability. *Data in Brief*, 18, 840–845.
- Houlsby, N., Huszár, F., Ghahramani, Z., & Lengyel, M. (2011). Bayesian active learning for classification and preference learning. [arXiv:1112.5745](https://arxiv.org/abs/1112.5745) [stat.ML].
- Huang, K., & Lin, H. (2016). A novel uncertainty sampling algorithm for cost-sensitive multiclass active learning. In *Proceedings of the 16th international conference on data mining (ICDM)* (pp. 925–930). IEEE.
- Hutter, F., Xu, L., Hoos, H. H., & Leyton-Brown, K. (2014). Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence*, 206, 79–111.
- Japkowicz, N., & Shah, M. (2011). *Evaluating learning algorithms: A classification perspective*. Cambridge University Press.
- Konyushkova, K., Sznitman, R., & Fua, P. (2018). Discovering general purpose active learning strategies. [arXiv:1810.04114v2](https://arxiv.org/abs/1810.04114v2) [cs.LG].
- Kottke, D., Krempel, G., Lang, D., Teschner, J., & Spiliopoulou, M. (2016). Multi-class probabilistic active learning. In *Proceedings of the European conference on artificial intelligence (ECAI)* (pp. 586–594). IOS Press.
- Kottke, D., Herde, M., Minh, T. P., Benz, A., Mergard, P., Roghman, A., Sandrock, C., & Sick, B. (2021). scikit-activeml: A library and toolbox for active learning algorithms. Preprints, 2021030194.
- Krempel, G., Kottke, D., & Lemaire, V. (2015). Optimised probabilistic active learning (OPAL). *Machine Learning*, 100(2–3), 449–476.
- Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international conference on research and development in information retrieval (SIGIR)* (pp. 3–12). Springer.
- Murphy, K. P. (2006). *Binomial and multinomial distributions*. Technical report, University of British Columbia.
- Nguyen, H. T., & Smeulders, A. (2004). Active learning using pre-clustering. In *Proceedings of the 21st international conference on machine learning (ICML)* (pp. 79–86). ACM Press.

- Osugi, T., Kim, D., & Scott, S. (2005). Balancing exploration and exploitation: A new algorithm for active machine learning. In *Proceedings of the 5th international conference on data mining (ICDM)* (pp. 330–337). IEEE.
- Roy, N., & McCallum, A. (2001). Toward optimal active learning through Monte Carlo estimation of error reduction. In *Proceedings of the 18th international conference on machine learning (ICML)* (pp. 441–448).
- Settles, B. (2009). *Active learning literature survey*. Technical report, University of Wisconsin-Madison Department of Computer Sciences.
- Settles, B. (2012). *Active learning. No. 18 in Synthesis lectures on artificial intelligence and machine learning*. Morgan and Claypool Publishers.
- Seung, H. S., Opper, M., & Sompolinsky, H. (1992). Query by committee. In *Proceedings of the 5th annual workshop on computational learning theory (COLT)* (pp. 287–294). ACM.
- Shi, S., Liu, Y., Huang, Y., Zhu, S., & Liu, Y. (2008). Active learning for kNN based on bagging features. In *Proceedings of the 4th international conference on natural computation* (pp. 61–64), Jinan, China.
- Thrun, S. B., & Möller, K. (1992). Active exploration in dynamic environments. In *Advances in neural information processing systems* (pp. 531–538).
- Vanschoren, J., van Rijn, J. N., Bischl, B., & Torgo, L. (2013). Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2), 49–60.
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer.
- Wei, K., Iyer, R., & Bilmes, J. (2015). Submodularity in data subset selection and active learning. In *Proceedings of the 32nd international conference on machine learning (ICML)* (pp. 1954–1963).
- Weigl, E., Heidl, W., Lughofer, E., Radauer, T., & Eitzinger, C. (2015). On improving performance of surface inspection systems by online active learning and flexible classifier updates. *Machine Vision and Applications*, 27(1), 103–127.
- Xu, Z., Akella, R., & Zhang, Y. (2007). Incorporating diversity and density in active learning for relevance feedback. In *Proceedings of the European conference on information retrieval (ECIR)* (pp. 246–257). Springer.
- Zoller, T., & Buhmann, J. M. (2000). Active learning for hierarchical pairwise data clustering. In *Proceedings 15th international conference on pattern recognition (ICPR)* (pp. 186–189). IEEE.
- Žliobaitė, I., Bifet, A., Pfahringer, B., & Holmes, G. (2014). Active learning with drifting streaming data. *Transactions on Neural Networks and Learning Systems*, 25(1), 27–39.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.