Epistemic Uncertainty Sampling

Vu-Linh Nguyen[1(✉)], Sébastien Destercke[2], and Eyke Hüllermeier[1]

[1] Heinz Nixdorf Institute, Department of Computer Science, Paderborn University, Paderborn, Germany
vu.linh.nguyen@uni-paderborn.de, eyke@upb.de

[2] UMR CNRS 7253 Heudiasyc, Sorbonne Universités, Université de Technologie de Compiègne, Compiègne, France
sebastien.destercke@hds.utc.fr

Abstract. Various strategies for active learning have been proposed in the machine learning literature. In uncertainty sampling, which is among the most popular approaches, the active learner sequentially queries the label of those instances for which its current prediction is maximally uncertain. The predictions as well as the measures used to quantify the degree of uncertainty, such as entropy, are almost exclusively of a probabilistic nature. In this paper, we advocate a distinction between two different types of uncertainty, referred to as *epistemic* and *aleatoric*, in the context of active learning. Roughly speaking, these notions capture the reducible and the irreducible part of the total uncertainty in a prediction, respectively. We conjecture that, in uncertainty sampling, the usefulness of an instance is better reflected by its epistemic than by its aleatoric uncertainty. This leads us to suggest the principle of "epistemic uncertainty sampling", which we instantiate by means of a concrete approach for measuring epistemic and aleatoric uncertainty. In experimental studies, epistemic uncertainty sampling does indeed show promising performance.

Keywords: Active learning · Uncertainty sampling · Epistemic uncertainty · Aleatoric uncertainty

1 Introduction

The goal in standard supervised learning, such as binary or multi-class classification, is to learn models with high predictive accuracy from labelled training data [7,22]. However, labelled data does normally not come for free. On the contrary, labelling can be expensive, time-consuming, and costly. The ambition of *active learning*, therefore, is to exploit labelled data in the most effective way. More specifically, the idea is to let the learning algorithm itself decide which examples it considers to be most informative. Compared to random sampling, the hope is to achieve better performance with the same amount of training data, or to reach the same performance with less data [6,20].

The selection of training examples is often done in an iterative manner, i.e., the active learner alternates between re-training and selecting new examples. In each iteration, the usefulness of a candidate example is estimated in terms of a *utility score*, and the one with the highest score is queried. In this regard, the notion of utility typically refers to uncertainty reduction: To what extent will the knowledge about the label of a specific instance help to reduce the learner's uncertainty about the sought model? In *uncertainty sampling* [20], which is among the most popular approaches, utility is quantified in terms of predictive uncertainty, i.e., the active learner selects those instances for which its current prediction is maximally uncertain. The predictions as well as the measures used to quantify the degree of uncertainty, such as entropy, are almost exclusively of a probabilistic nature. Such approaches indeed proved to be successful in many applications.

Yet, as pointed out by [21], existing approaches can be criticized for not informing about the *reasons* for why an instance is considered uncertain, although this might be relevant for judging the usefulness of an example. In this paper, we advocate a distinction between two different types of uncertainty, referred to as *epistemic* and *aleatoric*—roughly speaking, these capture the reducible and the irreducible part of the total uncertainty in a prediction, respectively. The conjecture that, in uncertainty sampling, the usefulness of an instance is better reflected by its epistemic than by its aleatoric uncertainty leads us to the idea of "epistemic uncertainty sampling". Our approach, which builds on a formalization of epistemic and aleatoric uncertainty as proposed by [19], is generic in the sense that is can be instantiated for any learning algorithm; concretely, we present instantiations for a Parzen window classifier, decision tree learning, and logistic regression.

The rest of this paper is organized as follows. In the next section, we recall the general framework of uncertainty sampling and provide a brief survey of related work on active learning. In Sect. 3, we recall the approach of [19] for modeling epistemic and aleatoric uncertainty, and then present our idea of generalizing uncertainty sampling on the basis of this approach. Instantiations of our approach for local learning (Parzen window classifier), decision tree learning and logistic regression are presented in Sect. 4. Experimental evaluations are given in the Sect. 5. The paper concludes with a short summary and an outlook on future work in Sect. 6.

2 Uncertainty Sampling

As usual in active learning, we assume to be given a labelled set of training data $\mathbf{D}$ and a pool of unlabeled instances $\mathbf{U}$ that can be queried by the learner:

$$\mathbf{D} = \big\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\big\}, \quad \mathbf{U} = \big\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_J\big\}$$

Instances are represented as features vectors $\boldsymbol{x}_i = \big(x_i^1, \ldots, x_i^d\big) \in \mathcal{X} = \mathbb{R}^d$. In this paper, we only consider the case of binary classification, where labels y_i are taken from $\mathcal{Y} = \{0, 1\}$, leaving the more general case of multi-class classification for future work. We denote by $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ the underlying hypothesis space, i.e., the class of candidate models $h : \mathcal{X} \longrightarrow \mathcal{Y}$ the learner can choose from. Often, hypotheses are parametrized by a parameter vector $\theta \in \Theta$; in this case, we equate a hypothesis $h = h_\theta \in \mathcal{H}$ with the parameter θ, and the model space $\mathcal{H}$ with the parameter space Θ.

In uncertainty sampling, instances are queried in a greedy fashion. Given the current model θ that has been trained on $\mathbf{D}$, each instance $\boldsymbol{x}_j$ in the current pool $\mathbf{U}$ is assigned a *utility* score $s(\theta, \boldsymbol{x}_j)$, and the next instance to be queried is the one with the highest score [11,20,21]. The chosen instance is labelled (by an oracle or expert) and added to the training data $\mathbf{D}$, on which the model is then re-trained. The active learning process for a given budget B (i.e., the number of unlabelled instances to be queried) is summarized in Algorithm 1.

Algorithm 1: Uncertainty sampling

```
  Input: U, D, θ- initial pool, training data, classifier, and B-budget
  Output: U, D, θ - updated pool, training data, classifier
1 initialize $b = 0$;
2 while $b < B$ do
3     foreach $\boldsymbol{x} \in \mathbf{U}$ do
4         compute $s(\theta, \boldsymbol{x})$
5     query the label of the optimal instance $\boldsymbol{x}^*$ with respect to $s(\theta, \boldsymbol{x})$
      $\mathbf{D} = \mathbf{D} \cup \{\boldsymbol{x}^*, y^*\}$ ;
6     $\mathbf{U} = \mathbf{U} \setminus \{\boldsymbol{x}^*, y^*\}$ ;
7     train $\theta$ from $\mathbf{D}$;
8     $b = b + 1$;
9 Return U, D, θ;
```

Assuming a probabilistic model producing predictions in the form of probability distributions $p_\theta(\cdot \,|\, \boldsymbol{x})$ on $\mathcal{Y}$, the utility score is typically defined in terms of a measure of uncertainty. Thus, instances on which the current model is highly uncertain are supposed to be maximally informative [20,21]. Popular examples of such measures include

– the entropy:

$$s(\theta, \boldsymbol{x}) = -\sum_{\lambda \in \mathcal{Y}} p_\theta(\lambda \,|\, \boldsymbol{x}) \log p_\theta(\lambda \,|\, \boldsymbol{x}), \tag{1}$$

– the least confidence:

$$s(\theta, \boldsymbol{x}) = 1 - \max_{\lambda \in \mathcal{Y}} p_\theta(\lambda \,|\, \boldsymbol{x}), \tag{2}$$

– the smallest margin:

$$s(\theta, \boldsymbol{x}) = p_\theta(\lambda_n \,|\, \boldsymbol{x}) - p_\theta(\lambda_m \,|\, \boldsymbol{x}), \tag{3}$$

where $\lambda_m = \arg\max_{\lambda \in \mathcal{Y}} p_\theta(\lambda \,|\, \boldsymbol{x})$ and $\lambda_n = \arg\max_{\lambda \in \mathcal{Y} \setminus \lambda_m} p_\theta(\lambda \,|\, \boldsymbol{x})$.

All the three measures ought to be maximized. In the case of binary classification, i.e., $\mathcal{Y} = \{0,1\}$, all these measures rank unlabelled instances in the same order and look for instances with small difference between $p_\theta(0 \,|\, \boldsymbol{x})$ and $p_\theta(1 \,|\, \boldsymbol{x})$.

3 Epistemic and Aleatoric Uncertainty

A main building block of our approach to active learning is the distinction between the *epistemic* and *aleatoric* uncertainty involved in the prediction for an instance $\boldsymbol{x}$. Although this distinction is well accepted in the literature on uncertainty [8], it has been considered in machine learning only very recently [9,13,19]. Here, we adopt the formal model proposed by [19], which is based on the use of relative likelihoods, historically proposed by [2] and then justified in other settings such as possibility theory [23]. For the sake of completeness and self-containedness, we briefly recall the essence of this approach.

As before, we proceed from an instance space $\mathcal{X}$, an output space $\mathcal{Y} = \{0,1\}$ encoding the two classes, and a hypothesis space $\mathcal{H}$ consisting of probabilistic classifiers $h : \mathcal{X} \longrightarrow [0,1]$. We denote by $p_h(1 \,|\, \boldsymbol{x}) = h(\boldsymbol{x})$ and $p_h(0 \,|\, \boldsymbol{x}) = 1 - h(\boldsymbol{x})$ the (predicted) probability that instance $\boldsymbol{x} \in \mathcal{X}$ belongs to the positive and negative class, respectively. Given a set of training data $\mathbf{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$, the normalized likelihood of a model h is defined as

$$\pi_{\mathcal{H}}(h) = \frac{L(h)}{L(h^{ml})} = \frac{L(h)}{\max_{h' \in \mathcal{H}} L(h')}, \tag{4}$$

where $L(h) = \prod_{i=1}^N p_h(y_i \,|\, \boldsymbol{x}_i)$ is the likelihood of h, and $h^{ml} \in \mathcal{H}$ the maximum likelihood estimation on the training data. For a given instance $\boldsymbol{x}$, the degrees of support (plausibility) of the two classes are defined as follows:

$$\pi(1 \,|\, \boldsymbol{x}) = \sup_{h \in \mathcal{H}} \min \big[\pi_{\mathcal{H}}(h), p_h(1 \,|\, \boldsymbol{x}) - p_h(0 \,|\, \boldsymbol{x})\big], \tag{5}$$

$$\pi(0 \,|\, \boldsymbol{x}) = \sup_{h \in \mathcal{H}} \min \big[\pi_{\mathcal{H}}(h), p_h(0 \,|\, \boldsymbol{x}) - p_h(1 \,|\, \boldsymbol{x})\big]. \tag{6}$$

So, $\pi(1 \,|\, \boldsymbol{x})$ is high if and only if a highly plausible model supports the positive class much stronger (in terms of the assigned probability mass) than the negative class (and $\pi(0 \,|\, \boldsymbol{x})$ can be interpreted analogously)[1]. Note that, with $f(a) = 2a - 1$, we can also rewrite (5)–(6) as follows:

$$\pi(1 \,|\, \boldsymbol{x}) = \sup_{h \in \mathcal{H}} \min \big[\pi_{\mathcal{H}}(h), f(h(\boldsymbol{x}))\big], \tag{7}$$

$$\pi(0 \,|\, \boldsymbol{x}) = \sup_{h \in \mathcal{H}} \min \big[\pi_{\mathcal{H}}(h), f(1 - h(\boldsymbol{x}))\big]. \tag{8}$$

Given the above degrees of support, the degrees of epistemic uncertainty u_e and aleatoric uncertainty u_a are defined as follows:

$$u_e(\boldsymbol{x}) = \min \big[\pi(1 \,|\, \boldsymbol{x}), \pi(0 \,|\, \boldsymbol{x})\big], \tag{9}$$

$$u_a(\boldsymbol{x}) = 1 - \max \big[\pi(1 \,|\, \boldsymbol{x}), \pi(0 \,|\, \boldsymbol{x})\big]. \tag{10}$$

[1] Technically, we assume that, for each $\boldsymbol{x} \in \mathcal{X}$, there are hypotheses $h, h' \in \mathcal{H}$ such that $h(\boldsymbol{x}) \geq 0.5$ and $h'(\boldsymbol{x}) \leq 0.5$, which implies $\pi(1 \,|\, \boldsymbol{x}) \geq 0$ and $\pi(0 \,|\, \boldsymbol{x}) \geq 0$.

Thus, epistemic uncertainty refers to the case where both the positive and the negative class appear to be plausible, while the degree of aleatoric uncertainty (10) is the degree to which none of the classes is supported. These uncertainty degrees are completed with degrees $s_1(\boldsymbol{x})$ and $s_0(\boldsymbol{x})$ of (strict) preference in favor of the positive and negative class, respectively:

$$s_1(\boldsymbol{x}) = \begin{cases} 1 - (u_a(\boldsymbol{x}) + u_e(\boldsymbol{x})) & \text{if } \pi(1 \,|\, \boldsymbol{x}) > \pi(0 \,|\, \boldsymbol{x}), \\ \frac{1-(u_a(\boldsymbol{x})+u_e(\boldsymbol{x}))}{2} & \text{if } \pi(1 \,|\, \boldsymbol{x}) = \pi(0 \,|\, \boldsymbol{x}), \\ 0 & \text{if } \pi(1 \,|\, \boldsymbol{x}) < \pi(0 \,|\, \boldsymbol{x}). \end{cases}$$

With an analogous definition for $s_0(\boldsymbol{x})$, we have $s_0(\boldsymbol{x})+s_1(\boldsymbol{x})+u_a(\boldsymbol{x})+u_e(\boldsymbol{x}) \equiv 1$. Besides, it has the following properties:

- $s_1(\boldsymbol{x})$ ($s_0(\boldsymbol{x})$) will be high if and only if, for all plausible models, the probability of the positive (negative) class is significantly higher than the one of the negative (positive) class;
- $u_e(\boldsymbol{x})$ will be high if class probabilities strongly vary within the set of plausible models, i.e., if we are unsure how to compare these probabilities. In particular, it will be 1 if and only if we have $h(\boldsymbol{x}) = 1$ and $h'(\boldsymbol{x}) = 0$ for two totally plausible models h and h';
- $u_a(\boldsymbol{x})$ will be high if class probabilities are similar for all plausible models, i.e., if there is strong evidence that $h(\boldsymbol{x}) \approx 0.5$. In particular, it will be close to 1 if all plausible models allocate their probability mass around $h(\boldsymbol{x}) = 0.5$.

Roughly speaking, aleatoric uncertainty is due to influences on the data-generating process that are inherently random, whereas epistemic uncertainty is caused by a lack of knowledge. Or, stated differently, u_e and u_a measure the *reducible* and the *irreducible* part of the total uncertainty, respectively. It thus appears reasonable to assume that epistemic uncertainty is more relevant for active learning: While it makes sense to query additional class labels in regions where uncertainty can be reduced, doing so in regions of high aleatoric uncertainty appears to be less reasonable. This leads us to the principle of *epistemic uncertainty sampling*, which prescribes the selection

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathbf{U}} u_e(\boldsymbol{x}). \tag{11}$$

For comparison, we will also consider an analogous selection rule based on the aleatoric uncertainty, i.e.,

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathbf{U}} u_a(\boldsymbol{x}). \tag{12}$$

Let us note that the above approach is completely generic and can in principle be instantiated with any hypothesis space $\mathcal{H}$. The uncertainty measures (11–12) can be derived very easily from the support degrees (7–8). The computation of the latter may become difficult, however, as it requires the solution of an optimization problem, the properties of which depend on the choice of $\mathcal{H}$.

4 Instantiations of the General Approach

We are going to present practical methods to determine (7–8) for the cases of local learning and logistic regression in Sects. 4.1 and 4.2, respectively.

4.1 Local Learning

This section presents an instantiation of our approach for the case of local learning using a Parzen window classifier [4]. The method is then adapted to the case where the decision tree classifier [16,18] is employed as the based learner.

As already said, instantiating the approach essentially means to address the question of how to compute the degrees of support (7–8), from which everything else can easily be derived.

By local learning, we refer to a class of non-parametric models that derive predictions from the training information in a local region of the instance space, for example the local neighborhood of a query instance [3,5]. As a simple example, we consider the Parzen window classifier [4], to which our approach can be applied in a quite straightforward way. To this end, for a given instance $\boldsymbol{x}$, define the set of its neighbours as follows:

$$R(\boldsymbol{x}, \epsilon) = \big\{ (\boldsymbol{x}_i, y_i) \in \mathbf{D} \,|\, \|\boldsymbol{x}_i - \boldsymbol{x}\| \leq \epsilon \big\}, \tag{13}$$

where ϵ is the width of the Parzen window (a practical method to determine such a width will be given latter).

In binary classification, a local region R can be associated with a constant hypothesis h_θ, $\theta \in \Theta = [0,1]$, where $h_\theta(\boldsymbol{x}) \equiv \theta$ is the probability of the positive class in the region; thus, h_θ predicts the same probabilities $p_h(1 \,|\, \boldsymbol{x}) = \theta$ and $p_h(0 \,|\, \boldsymbol{x}) = 1 - \theta$ for all $\boldsymbol{x} \in R$. The underlying hypothesis space is given by $\mathcal{H} = \{h_\theta \,|\, 0 \leq \theta \leq 1\}$. With n and p the number of positive and negative instances, respectively, within a Parzen window $R(\boldsymbol{x}, \epsilon)$, the likelihood and the maximum likelihood estimate of θ are respectively given by

$$L(\theta) = \binom{n+p}{n} \theta^n (1-\theta)^p \text{ and } \hat{\theta} = \frac{n}{n+p}. \tag{14}$$

Therefore, the degrees of support for the positive and negative classes are

$$\pi(1 \,|\, \boldsymbol{x}) = \sup_{\theta \in [0,1]} \min \left(\frac{\theta^p (1-\theta)^n}{\left(\frac{p}{n+p}\right)^p \left(\frac{n}{n+p}\right)^n}, 2\theta - 1 \right), \tag{15}$$

$$\pi(0 \,|\, \boldsymbol{x}) = \sup_{\theta \in [0,1]} \min \left(\frac{\theta^p (1-\theta)^n}{\left(\frac{p}{n+p}\right)^p \left(\frac{n}{n+p}\right)^n}, 1 - 2\theta \right). \tag{16}$$

Solving (15) and (16) comes down to maximizing a scalar function over a bounded domain, for which standard solvers can be used. We applied Brent's method[2] (which is a variant of the golden section method) to find a local minimum in the interval $\theta \in [0, 1]$. From (15–16), the epistemic and aleatoric uncertainty associated with the region R can be derived according to (11) and (12), respectively. For different combinations of n and p, these uncertainty degrees can be pre-computed (cf. Fig. 1).

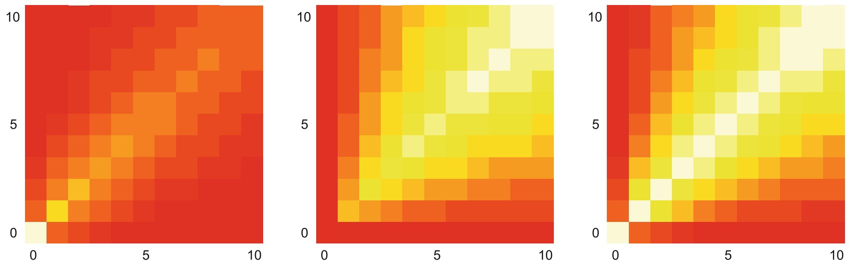

Fig. 1. From left to right: Epistemic, aleatoric, and total uncertainty (epistemic + aleatoric) as a function of the numbers $p, n \in \{0, 1, \ldots, 10\}$ of positive and negative examples in a region (Parzen window) of the instance space (lighter colors indicate higher values).

How to determine the width ϵ of the Parzen window? This value is difficult to assess, and an appropriate choice strongly depends properties of the data and the dimensionality of the instance space. Intuitively, it is even difficult to say in which range this value should lie. Therefore, instead of fixing ϵ, we fixed an absolute number K of neighbors in the training data, which is intuitively more meaningful and easier to interpret. A corresponding value of ϵ is then determined in such a way that the average number of nearest neighbours of instances $\boldsymbol{x}_i$ in the training data $\mathbf{D}$ is just K (see Algorithm 2). In other words, ϵ is determined indirectly via K.

Since K is an average, individual instances may have more or less neighbors in their Parzen windows. In particular, a Parzen window may also be empty. In this case, we set $u_e(\boldsymbol{x}) = 1$ by definition, i.e., we consider this as a case of full epistemic uncertainty. Likewise, the uncertainty is considered to be maximal for all other sampling techniques. If the accuracy of the Parzen classifier needs to be determined, we assume that it yields a wrong prediction.

In a similar way, the approach can be applied to decision tree learning [16, 18]. In fact recall that a decision tree partitions the instance space $\mathcal{X}$ into (rectangular) regions $R_1, \ldots, R_L$ (i.e., $\bigcup_{i=1}^{L} R_i = \mathcal{X}$ and $R_i \cap R_j = \emptyset$ for $i \neq j$) associated with corresponding leafs of the tree (each leaf node defines a region R). Again, in the case of binary classification, we can assume each region R to be associated with a constant hypothesis h_θ, $\theta \in \Theta = [0, 1]$, where $h_\theta(\boldsymbol{x}) \equiv \theta$ is the probability

[2] For an implementation in Python, see https://docs.scipy.org/doc/scipy-0.19.1/reference/generated/scipy.optimize.minimize_scalar.html.

Algorithm 2: Determining the width ϵ.

```
   Input: $\mathbf{D}$-normalized data, $K$-number
   Output: the local width $\epsilon_K$
1  foreach $\boldsymbol{x}_n \in \mathbf{D}$ do
2      foreach $\boldsymbol{x}_m \neq \boldsymbol{x}_n$ do
3          compute $d(\boldsymbol{x}_n, \boldsymbol{x}_m)$;
4      form $1 \times (n-1)$ vector $\mathbf{d}_n = \big(d(\boldsymbol{x}_n, \boldsymbol{x}_m) \mid n \neq m\big)$;
5      sort $\mathbf{d}_n$ by increasing order and determine the $K$-th element $\mathbf{d}_n^K$;
6  return $\epsilon_K = \frac{\sum_{n=1}^{|\mathbf{D}|} \mathbf{d}_n^K}{|\mathbf{D}|}$;
```

of the positive class. Therefore, degrees of epistemic and aleatoric uncertainty degrees can be derived in the same way as described above.

4.2 Logistic Regression

In this section, we present another instantiation of our approach for a commonly used learning algorithm, namely logistic regression. In contrast to nonparametric, local learning methods such as the Parzen window classifier, logistic regression is a parametric class of linear models, and hence coming with comparatively restrictive assumptions.

Recall that logistic regression assumes posterior probabilities to depend on feature vectors $\boldsymbol{x} = (x^1, \ldots, x^d) \in \mathbb{R}^d$ in the following way:

$$h(\boldsymbol{x}) = p(1 \mid \boldsymbol{x}) = \frac{\exp\left(\theta_0 + \sum_{i=1}^d \theta_i \, x^i\right)}{1 + \exp\left(\theta_0 + \sum_{i=1}^d \theta_i \, x^i\right)} \tag{17}$$

This means that learning the model comes down to estimating a parameter vector $\theta = (\theta_0, \ldots, \theta_d)$, which is commonly done through likelihood maximization [12]. To avoid numerical issues (e.g, having to deal with the exponential function for large θ) when maximizing the target function, we employ L_2-regularization. The corresponding version of the log-likelihood function (18) is strictly concave [17]:

$$\begin{aligned} l(\theta) = \log L(\theta) = & \sum_{n=1}^N y_n \left(\theta_0 + \sum_{i=1}^d \theta_i x_n^i\right) \\ & - \sum_{n=1}^N \ln\left(1 + \exp\left(\theta_0 + \sum_{i=1}^d \theta_i x_n^i\right)\right) - \frac{\gamma}{2} \sum_{i=0}^d \theta_i^2, \end{aligned} \tag{18}$$

where the regularization term γ will be fixed to 1.

We now focus on determining the degree of support (7) for the positive class, and then summarize the results for the negative class (which can be determined in a similar manner). Associating each hypothesis $h \in \mathcal{H}$ with a vector $\theta \in \mathbb{R}^{d+1}$, the degree of support (7) can be rewritten as follows:

$$\pi(1 \,|\, \boldsymbol{x}) = \sup_{\theta \in \mathbb{R}^{d+1}} \min \big[\pi(\theta), 2h(\boldsymbol{x}) - 1\big] \tag{19}$$

It is easy to see that the target function to be maximized in (19) is not necessarily concave. Therefore, we propose the following approach.

Let us first note that whenever $h(\boldsymbol{x}) < 0.5$, we have $2h(\boldsymbol{x}) - 1 \leq 0$ and $\min\big[\pi_{\mathcal{H}}(h), 2h(\boldsymbol{x}) - 1\big] \leq 0$. Thus the optimal value of the target function (7) can only be achieved for some hypotheses h such that $h(\boldsymbol{x}) \in [0.5, 1]$. For a given value $\alpha \in [0.5, 1]$, the set of hypotheses h such that $h(\boldsymbol{x}) = \alpha$ corresponds to the convex set

$$\theta^{\alpha} = \left\{\theta \,\Big|\, \theta_0 + \sum_{i=1}^{d} \theta_i x^i = \ln\left(\frac{\alpha}{1-\alpha}\right)\right\}. \tag{20}$$

The optimal value $\pi^*_{\alpha}(1 \,|\, \boldsymbol{x})$ that can be achieved within the region (20) can be determined as follows:

$$\pi^*_{\alpha}(1 \,|\, \boldsymbol{x}) = \sup_{\theta \in \theta^{\alpha}} \min \big[\pi(\theta), 2\alpha - 1\big] = \min \big[\sup_{\theta \in \theta^{\alpha}} \pi(\theta), 2\alpha - 1\big]. \tag{21}$$

Thus, to find this value, we maximize the concave log-likelihood over a convex set:

$$\theta^*_{\alpha} = \arg \sup_{\theta \in \theta^{\alpha}} l(\theta) \tag{22}$$

As the log-likelihood function (18) is concave and has second-order derivatives, we tackle the problem with a Newton-CG algorithm [14]. Furthermore, the optimization problem (22) can be solved using sequential least squares programming[3] [15]. Since regions defined in (20) are parallel hyperplanes, the solution of the optimization problem (7) can then be obtained by solving the following problem:

$$\sup_{\alpha \in [0.5,1)} \pi^*_{\alpha}(1|\boldsymbol{x}) = \sup_{\alpha \in [0.5,1)} \min \big[\pi(\theta^*_{\alpha}), 2\alpha - 1\big]. \tag{23}$$

Following a similar procedure, we can estimate the degree of support for the negative class (8) as follows:

$$\sup_{\alpha \in (0,0.5]} \pi^*_{\alpha}(0|\boldsymbol{x}) = \sup_{\alpha \in (0,0.5]} \min \big[\pi(\theta^*_{\alpha}), 1 - 2\alpha\big] \tag{24}$$

Note that limit cases $\alpha = 1$ and $\alpha = 0$ cannot be solved, since the region (20) is then not well-defined (as $\ln(\infty)$ and $\ln(0)$ do not exist). For the purpose of practical implementation, we handle (23) by discretizing the interval over α. That is, we optimize the target function for a given number of values $\alpha \in [0.5, 1)$

[3] For an implementation in Python, see https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html.

and consider the solution corresponding to the α with the highest optimal value of the target function $\pi^*_\alpha(1 \,|\, \boldsymbol{x})$ as the maximum estimator. Similarly, (24) can be handled over the domain $(0, 0.5]$.

In practice, we evaluate (23) and (24) on uniform discretizations of cardinality 50 of $[0.5, 1)$ and $(0, 0.5]$, respectively. We can further increase efficiency by avoiding computations for values of α for which we know that $2\alpha - 1$ and $1 - 2\alpha$ are lower than the current highest support value given to class 1 and 0, respectively. See Algorithm 3 for a pseudo-code description of the whole procedure.

Algorithm 3: Degrees of support for logistic regression

```
   Input: $Q$, $\mathbf{D}$, $\theta^{ml}$, $\boldsymbol{x}$- initial pool, training data, classifier, unlabelled instance
   Output: $\pi(1 \,|\, \boldsymbol{x})$, $\pi(0 \,|\, \boldsymbol{x})$ - degrees of support
1  initialize subsets $Q_p$, $Q_n$ of cardinality $Q$;
2  $\pi(1 \,|\, \boldsymbol{x}) = \max(2h^{ml}(\boldsymbol{x}) - 1, 0)$ , $\pi(0 \,|\, \boldsymbol{x}) = \max(1 - 2h^{ml}(\boldsymbol{x}), 0)$ ;
3  for $q = 1, \ldots, Q$ do
4      $\alpha_p = \max(Q_p)$; $\alpha_n = \min(Q_n)$ ;
5      if $2\alpha_p - 1 > \pi(1 \,|\, \boldsymbol{x})$ then
6          solve (22) for $\boldsymbol{x}$, $\alpha_p$ and return $\theta$;
7          $\pi(1 \,|\, \boldsymbol{x}) = \max(\pi(1 \,|\, \boldsymbol{x}), \min(\pi_{\mathcal{H}}(\theta), 2\alpha_p - 1))$ ;
8      if $1 - 2\alpha_n > \pi(0 \,|\, \boldsymbol{x})$ then
9          solve (22) for $\boldsymbol{x}$, $\alpha_n$ and return $\theta$;
10         $\pi(0 \,|\, \boldsymbol{x}) = \max(\pi(0 \,|\, \boldsymbol{x}), \min(\pi_{\mathcal{H}}(\theta), 1 - 2\alpha_p))$ ;
11     $Q_p = Q_p \setminus \{\alpha_p\}$, $Q_n = Q_n \setminus \{\alpha_n\}$ ;
12 Return $\pi(1 \,|\, \boldsymbol{x})$, $\pi(0 \,|\, \boldsymbol{x})$ ;
```

5 Experimental Results

To illustrate the performance of our uncertainty measures in active learning, we conducted experiments on data sets from the UCI repository[4], the main properties of which are summarized in Table 1.

5.1 Local Learning

We follow a 10-fold cross-validation procedure, considering each fold as the test set, while the other folds are used for learning. The latter is randomly split into a training data set and a pool set. The proportions of training/pool/test sets are 10/80/10% and accuracies are averaged. The budget of the active learner is fixed to be 30% of the original data.

[4] http://archive.ics.uci.edu/ml/index.php.

Table 1. Data sets used in the experiments

#	Name	# instances	# features	Attributes
1	Parkinsons	197	22	Real
2	Vertebral-column	310	6	Real
3	Ionosphere	351	34	Real
4	Climate-model	540	18	Real
5	Breast-cancer	569	30	Real
6	Blood-transfusion	748	5	Real
7	QSAR	1055	41	Integer, real
8	Banknote-authentication	1372	4	Real

After each query, we update the data sets and, correspondingly, the classifiers. The improvements of the classifiers are compared for four different uncertainty measures, i.e., uncertainty sampling (following the strategy presented in Algorithm 1) based on four measures for selecting unlabelled instances: random sampling, standard uncertainty (2), epistemic uncertainty (9), aleatoric uncertainty (10).

To reduce the computational efforts, in each iteration, the learner is allowed to evaluate and query instances from a randomly selected subset consisting of 10% of the data in the pool. Since we are not, in the first place, interested in maximizing performance, but in analyzing the effectiveness of active learning approaches, we simply fix the neighborhood size K as the square root of the size of the data set (number of instances in the initial training set and pool) [10].

As can be seen in Fig. 2, the results are nicely in agreement with our expectations: Epistemic uncertainty sampling performs the best and aleatoric uncertainty sampling the worst. Moreover, standard uncertainty sampling and random sampling are in-between the two. This supports our conjecture that, from an active learning point of view, epistemic uncertainty is the more useful information. Even if the improvements compared to standard uncertainty sampling are not huge, they are still visible and quite consistent.

The results for decision tree learning (cf. Fig. 3) are quite similar and again in agreement with our expectations.

5.2 Logistic Regression

For logistic regression, we start with a relatively small amount of initial training data, thereby making improvements in the beginning more visible. More specifically, the proportions of training/pool/test set are 1/89/10%, and the accuracies are averaged. The budget is fixed to be 20% of the original data, and in each iteration, the learner is allowed to evaluate and query instances from a (randomly) subset consisting of 10% data of the pool.

In the case of logistic regression, the improvements through epistemic uncertainty sampling are less pronounced—on the contrary, the performance of epis-

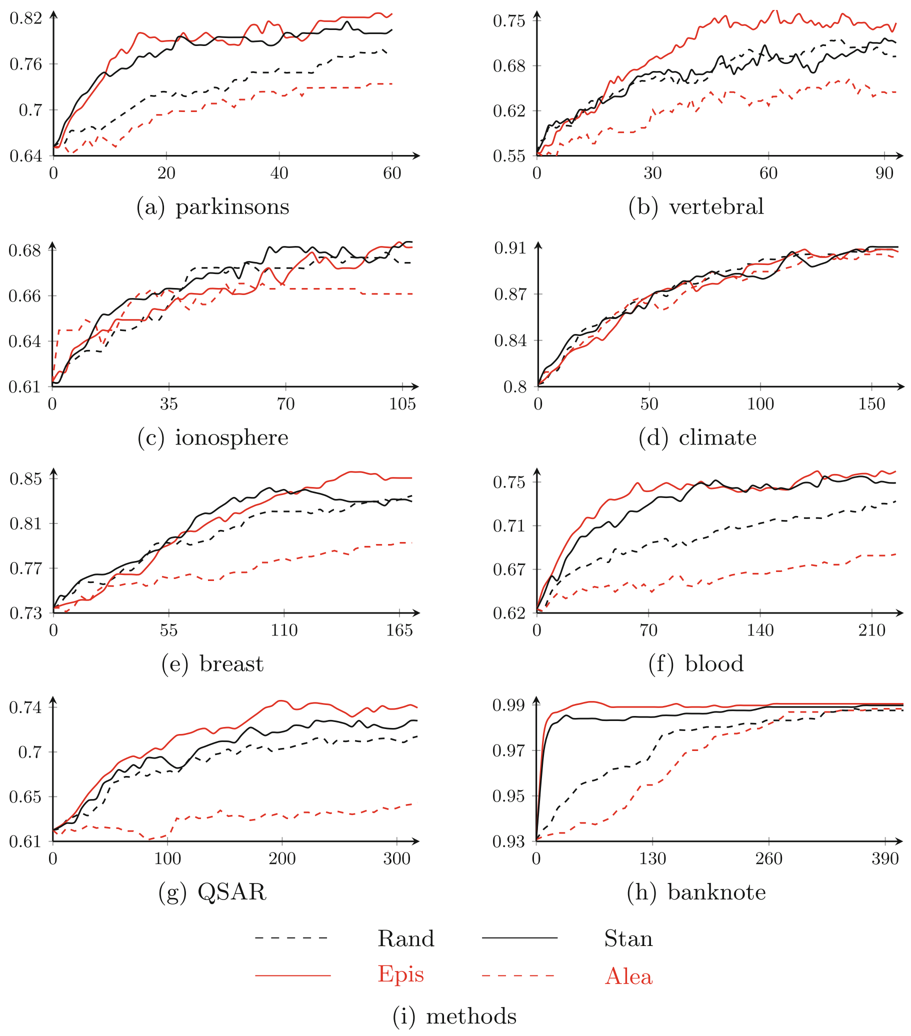

Fig. 2. Average accuracies (y-axis) for the Parzen window classifier as a function of the number of examples queried from the pool (x-axis).

temic and standard uncertainty sampling is quite comparable. Two examples, which are quite representative, are shown in Fig. 4. As a plausible explanation, note that logistic regression comes with a very strong learning bias in the form of a linearity assumption. Therefore, the epistemic (or model) uncertainty disappears quite quickly.

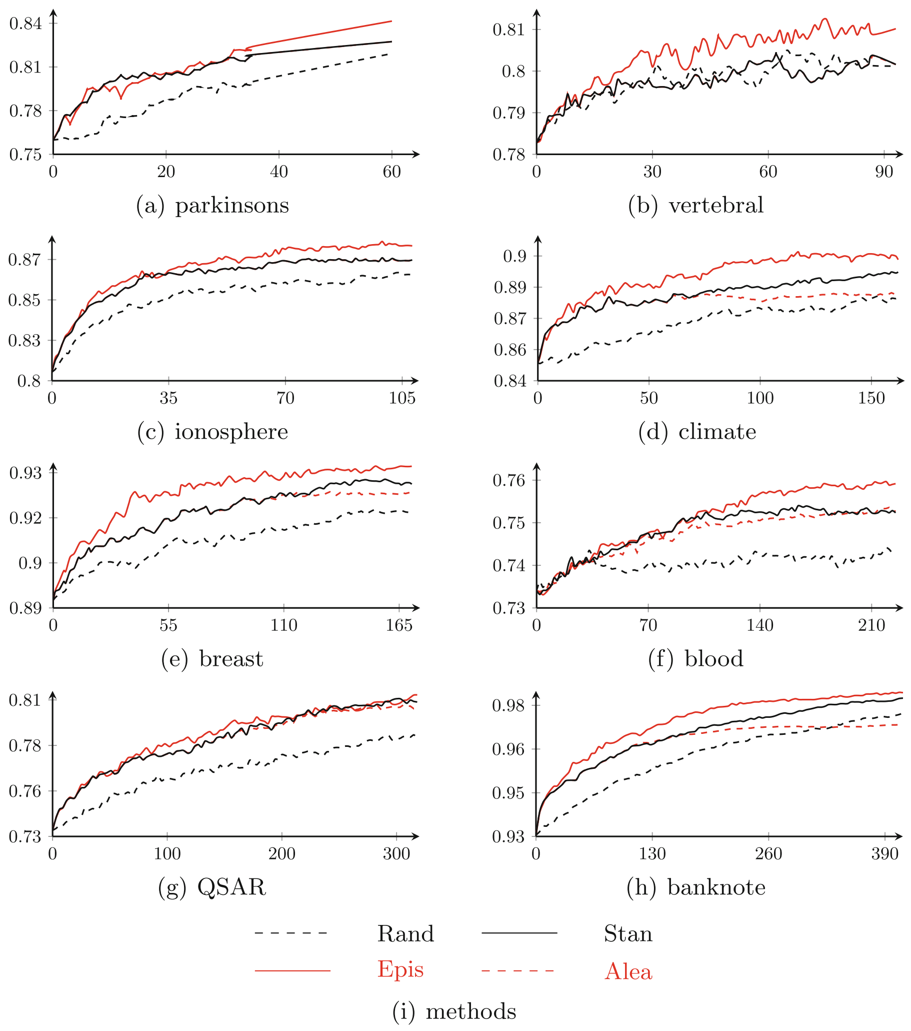

Fig. 3. Average accuracies (y-axis) for the decision tree classifier as a function of the number of examples queried from the pool (x-axis).

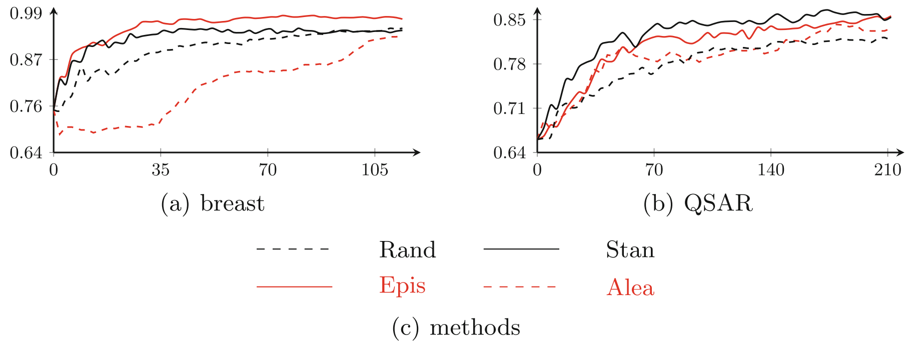

Fig. 4. Average accuracies (y-axis) for logistic regression as a function of the number of examples queried from the pool (x-axis).

6 Conclusion

This paper reconsiders the principle of uncertainty sampling in active learning from the perspective of uncertainty modeling. More specifically, it starts from the supposition that, when it comes to the question of which instances to select from a pool of candidates, a learner's predictive uncertainty due to "not knowing" should be more relevant than its uncertainty due to inherent randomness.

To corroborate this conjecture, we proposed *epistemic uncertainty sampling*, in which standard uncertainty measures such as entropy are replaced by a novel measure of epistemic uncertainty. The latter is borrowed from a recent framework for uncertainty modeling, in which epistemic uncertainty is distinguished from aleatoric uncertainty [19]. We interpret our experimental results, especially those for local learning (Parzen window classifier and decision trees) as evidence in favor of our conjecture. They clearly show that a separation of the total uncertainty (into epistemic and aleatoric) is effective, and that the epistemic part is the better criterion for selecting instances to be queried. This was the main purpose of the paper.

Given this affirmation, we are now encouraged to elaborate on epistemic uncertainty sampling in more depth, and to develop it in more sophistication. This includes an extension to other learning algorithms and more general learning problems (such as multi-class classification), as well as a comparison to other variants of uncertainty sampling, such as [1] and [21].

Acknowledgements. This work was supported by the German Research Foundation (DFG) and the French National Agency for Research (Labex MS2T).

References

1. Antonucci, A., Corani, G., Gabaglio, S.: Active learning by the naive credal classifier. In: Proceedings of the Sixth European Workshop on Probabilistic Graphical Models (PGM), pp. 3–10 (2012)

2. Birnbaum, A.: On the foundations of statistical inference. J. Am. Stat. Assoc. **57**(298), 269–306 (1962)
3. Bottou, L., Vapnik, V.: Local learning algorithms. Neural Comput. **4**(6), 888–900 (1992)
4. Chapelle, O.: Active learning for Parzen window classifier. In: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS), vol. 5, pp. 49–56 (2005)
5. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
6. Fu, Y., Zhu, X., Li, B.: A survey on instance selection for active learning. Knowl. Inf. Syst. **35**(2), 249–283 (2013)
7. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. Math. Intelligencer **27**(2), 83–85 (2005)
8. Hora, S.C.: Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management. Reliab. Eng. Syst. Saf. **54**(2–3), 217–223 (1996)
9. Kendall, A., Gal, Y.: What uncertainties do we need in Bayesian deep learning for computer vision? In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS) (2017)
10. Lall, U., Sharma, A.: A nearest neighbor bootstrap for resampling hydrologic time series. Water Resour. Res. **32**(3), 679–693 (1996)
11. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Croft, B.W., van Rijsbergen, C.J. (eds.) SIGIR 1994, pp. 3–12. Springer, London (1994). https://doi.org/10.1007/978-1-4471-2099-5_1
12. Menard, S.: Applied Logistic Regression Analysis, vol. 106. Sage, Thousand Oaks (2002)
13. Nguyen, V.L., Destercke, S., Masson, M.H., Hüllermeier, E.: Reliable multi-class classification based on pairwise epistemic and aleatoric uncertainty. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), pp. 5089–5095. AAAI Press (2018)
14. Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer, New York (2006). https://doi.org/10.1007/978-0-387-40065-5
15. Philip, E., Elizabeth, W.: Sequential quadratic programming methods. UCSD Department of Mathematics, Technical report NA-10-03 (2010)
16. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
17. Rennie, J.D.: Regularized logistic regression is strictly convex. Technical report, MIT (2005)
18. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. IEEE Trans. Syst. Man. Cybern. **21**(3), 660–674 (1991)
19. Senge, R., et al.: Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. Inf. Sci. **255**, 16–29 (2014)
20. Settles, B.: Active learning literature survey. Technical report, University of Wisconsin, Madison, vol. 52, no. 55–66, p. 11 (2010)
21. Sharma, M., Bilgic, M.: Evidence-based uncertainty sampling for active learning. Data Min. Knowl. Disc. **31**(1), 164–202 (2017)
22. Vapnik, V.N.: An overview of statistical learning theory. IEEE Trans. Neural Networks **10**(5), 988–999 (1999)
23. Walley, P., Moral, S.: Upper probabilities based only on the likelihood function. J. Roy. Stat. Soc. Ser. B (Stat. Methodol.) **61**(4), 831–847 (1999)