Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Hardness Sampling: Exploring Instance Hardness in Pool-based Active Learning

Gabriel da S. C. Nogueira

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Luís Paulo Faina Garcia

Brasília
2024

CIP - Catalogação na Publicação

da Silva Corvino Nogueira, Gabriel.
dN778h Hardness Sampling: Exploring Instance Hardness in Pool-based Active Learning / Gabriel da Silva Corvino Nogueira; orientador Luís Paulo Faina Garcia. -- Brasília, 2024.
47 p.

Monografia (Graduação - Ciência da Computação (Bacharelado)) -- Universidade de Brasília, 2024.

1. Pool-based Active Learning. 2. Instance Hardness. 3. Hardness Measures. 4. Query Strategies. 5. Machine Learning. I. Faina Garcia, Luís Paulo, orient. II. Título.

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Hardness Sampling: Exploring Instance Hardness in Pool-based Active Learning

Gabriel da S. C. Nogueira

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Luís Paulo Faina Garcia (Orientador)
CIC/UnB

Prof. Dr. Vinícius Ruela Pereira Borges
CIC/UnB

Prof. Dr. Pedro Garcia Freitas
CIC/UnB

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 08 de novembro de 2024

Dedicatória

Eu dedico este trabalho aos meus pais, Jane e Marcelo.

Agradecimentos

Este trabalho encerra meu longo ciclo como aluno de graduação na Universidade de Brasília. Foi um período de muitas descobertas, que certamente impactaram profundamente na minha formação pessoal e profissional. Dessa forma, gostaria de agradecer a todos que fizeram parte desse processo.

Em primeiro lugar, agradeço ao Prof. Luís Paulo pela orientação excepcional. Sou muito grato por ele ter sempre acreditado em mim e por ser tão paciente e atencioso. Mesmo dominando o conteúdo, ele sempre se demonstrou uma pessoa humilde, e a convivência com ele reforçou meu desejo de seguir na produção científica.

Agradeço também ao Prof. Davi (UTFPR), que me auxiliou no processo de revisão deste trabalho. Foi uma honra contar com as opiniões de alguém tão experiente tanto na área de aprendizado ativo quanto no processo de redação científica.

Agradeço aos membros do projeto KnEDLe, responsável pelo meu primeiro contato direto com aprendizado de máquina. Em especial, ao Prof. Vinícius, que me apresentou à área de aprendizado ativo e me incentivou a continuar no projeto, mesmo quando eu duvidava de minhas capacidades. Também agradeço ao Prof. Ricardo Marcacini (USP), por me proporcionar a experiência de trabalhar diretamente com aprendizado ativo em problemas reais e por possibilitar minha primeira publicação científica.

Agradeço à minha mãe, Jane, que recentemente se tornou doutora. Ela é um exemplo para mim, tanto na vida acadêmica quanto pessoal, sendo a pessoa mais doce e empática que conheço. Sou realizado por tê-la ao meu lado para me lembrar de que certas coisas, mesmo que comuns, jamais devem ser tratadas com banalidade.

Agradeço ao meu meu pai, Marcelo, que sempre me apoiou e me incentivou a tomar as decisões corretas, por mais difíceis que fossem. Sou muito grato por ter alguém com uma visão de mundo tão rica, que frequentemente consegue enxergar o que está além do meu alcance.

Agradeço à minha irmã, Rafaela, por simplesmente existir. Foram anos pedindo uma irmã, e sinto-me privilegiado por acompanhar seu crescimento e pela pessoa incrível que ela está se tornando.

Agradeço profundamente minha namorada, Maria Vitória, que neste momento está do outro lado do Atlântico. Sua presença foi essencial para me lembrar da importância de aproveitar os momentos de lazer e valorizar as pequenas coisas da vida.

Antes de concluir, gostaria de expressar minha gratidão ao curso de Engenharia Mecatrônica, que me abriu as portas para o mundo da computação, e ao curso de Bacharelado em Ciência da Computação, no qual tenho muito orgulho de estar me formando.

Por fim, deixo meu agradecimento a todas as demais pessoas que passaram pela minha vida durante esse período. Cada uma delas contribuiu de alguma forma para a pessoa que sou hoje e, consequentemente, para a realização deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Hardness Sampling: Exploring Instance Hardness in Pool-based Active Learning

Gabriel da S. C. Nogueira[a,*], Davi P. dos Santos[b], Luís P. F. Garcia[a]

[a] *University of Brasília, Computer Science Department, Campus Universitário Darcy Ribeiro, Brasília, 70910-900, Distrito Federal, Brazil*
[b] *Federal Technological University of Paraná, Medianeira Campus, Academic Department of Computing, Av. Brasil, 4232 - Independência, Medianeira, 85884-000, Paraná, Brazil*

Abstract

Active Learning (AL) techniques enable the creation of efficient models with minimal annotation effort by deciding which portions of the available data are worth learning. Pool-based AL (PAL) is a specific scenario in which instances within a pool of unlabeled data must be selected, labeled by an oracle, and incorporated into a subset of the pool to be used as a training set. The goal of PAL is to build a growing subset that is increasingly more representative of the problem at hand. However, the proper strategy for an optimal query of such instances is still an open question. In this paper, we resort to Hardness Measures (HMs) to enrich the current repertoire of PAL strategies available to address this question. HMs are metrics that employ the Instance Hardness (IH) concept to identify instances with a higher probability of being misclassified and have been successfully applied in areas such as meta-learning and explainable AI. Likewise, this study adds to this collective effort by exploring the use of IH in the context of AL, examining HMs as informativeness criteria for PAL, which led to a new PAL strategy called Hardness Sampling (HardS). We tested HardS across multiple datasets and learners, demonstrating its competitive performance compared to classical strategies such as Uncertainty Sampling, Expected Error Reduction, and Density-weighted methods. The results also highlighted the success of neighborhood-based measures, especially the ratio of the intra-class and

*Corresponding author
Email addresses: gabrielscnogueira@gmail.com (Gabriel da S. C. Nogueira), davip@utfpr.edu.br (Davi P. dos Santos), luis.garcia@unb.br (Luís P. F. Garcia)

extra-class distances at an instance level. Additionally, some tree-based and likelihood-based measures also showed promising performance.

Keywords: Pool-based active learning, Instance hardness, Hardness measures, Hardness sampling, Query strategies

1. Introduction

The task of labeling instances is increasingly important as new data has been generated across a wider range of applications each day. However, building datasets can be expensive depending on the labeling costs. For instance, a physician may be hired to correctly determine the class of each patient whose data will be used to train a Machine Learning (ML) algorithm. In the more general case, such a specialist is called an *oracle*, which is often a highly skilled person [1]. Notwithstanding, any tool or process able to provide a ground truth for the problem at hand can be the oracle: a machine, a chemical process, a probe in space or under the ocean waters, an expensive calculation, among others.

Active Learning (AL) techniques can be employed to address the issue of managing the inherent costs of building new datasets. They provide the learning algorithm with a strategy of curiosity analogously to human active learning, i.e., the power to efficiently select which parts of the available data are more interesting or worth learning. Consequently, AL allows such algorithms to use less training data without affecting performance [2]. In this study, we focus on Pool-based AL (PAL) [3], which deals with a small labeled data set $\mathcal{L}$ and a large pool of unlabeled data $\mathcal{U}$. A typical PAL process selects a subset of instances contained in $\mathcal{U}$ to be labeled by the oracle to increase the size of $\mathcal{L}$. The method used to select these instances relies on the *query strategy* applied, which, in turn, defines the criteria for categorizing a certain instance as critical or not to induce a predictive model able to generalize beyond the available data for the problem.

Additionally, PAL strategies can be classified as *agnostic* or *non-agnostic* regarding their dependency on the learner's predictions. Strategies following the agnostic approach do not assume the correctness of the separation surface established by the predictive model induced. In contrast, non-agnostic strategies rely on the decision boundary estimated by the active learner to assign a specific degree of informativeness to the unlabeled instances [4]. Thus, due to their reliance on the learner's estimated classification, non-agnostic

strategies tend to be more prospective. Namely, they focus on the most promising regions of the input space [4]. As a result, these strategies lean to be useful in the later stages of the sampling process, as the decision boundary is generally more reliable with a larger number of labeled instances. However, an exploratory bias in PAL involves querying instances from dense and unknown regions of the feature space [4]. Consequently, combining this bias with prospective approaches may lead to better results.

Another important concept that forms the basis for this work is Instance Hardness (IH). Specifically in classification problems, IH can be referred to as a property that indicates the probability of each instance in the dataset being misclassified [5]. However, according to the work of Smith et al. [5], this property can only be obtained through the performance evaluation of various classifiers on a given instance and it does not help identify reasons for such hardness. With this in mind, the authors propose Hardness Measures (HM), which are intended to indicate the reason why an instance is harder to classify. In addition, recent studies [6, 7] have also summarized instance hardness approaches and introduced new HMs.

Although HMs have been applied for different purposes in literature [6, 8, 9, 10, 11, 12], their applicability within PAL remains unexplored to the extent of our knowledge. To address this gap, this work presents the following contributions:

- Explore the use of HMs as informativeness measures in PAL.
- Propose Hardness Sampling (HardS), a non-agnostic query strategy for PAL based on HMs.
- Perform a comprehensive evaluation of the HardS strategy on a diverse set of datasets and learning algorithms, comparing its performance with classical strategies from the literature.

These contributions represent an innovative approach to PAL. Specifically, our strategy selects instances based on their potential hardness, i.e. the value attributed by some HM to instances in $\mathcal{U}$. Since the measures require labeled instances, we address this limitation by assigning labels according to the learner's predictions on those instances. Through the application of HardS, we aim to determine whether HMs can serve as effective informativeness measures in the domain of PAL and whether they offer a viable alternative to other non-agnostic strategies found in the literature. Furthermore, we aim to investigate the individual behavior of these measures in

terms of performance and their influence on the balance between exploratory and prospective sampling biases.

Given that many query strategies reported in the literature are often evaluated on specific datasets that highlight their effectiveness [13], we set out to conduct a more detailed and comprehensive evaluation of our approach. To this end, the HardS strategy was evaluated on 90 classification datasets, considering all combinations of 19 HMs and 4 different learning algorithms. Moreover, methods within classical frameworks such as Uncertainty Sampling [14], Expected Error Reduction [15] and Density Weighted Methods [16, 17] were also included for comparison purposes. The performance of each method was evaluated based on the overall mean rank achieved by them, as well as through the graphical analysis of their ranking curves [18].

The results suggest that while HardS is a competitive strategy, its performance depends on both the group of the HM used and the specific HM itself. Markedly, measures belonging to the neighborhood-based, tree-based, and likelihood-based groups presented the best average ranks. On top of that, some measures stood out within these groups, showing potential for balancing prospection and exploration across learners.

Regarding the structure of this work, Section 2 will present the group of non-agnostic query strategies for PAL, which we classify as classical in this study, along with the main methods for their implementation. This section will also discuss other relevant works in the field and recent advances. Section 3 will define the concept of instance hardness and introduce the HMs used in the experiments. Section 4 formally introduces the HardS strategy, while Section 5 describes the methodology employed to conduct the experiments. Section 6 presents the results obtained. Finally, Section 7 concludes this study by summarizing the findings and offering suggestions for future research.

2. Non-agnostic Query Strategies for PAL

ML predictive models can mimic the human curiosity of a student by querying the instances where they are least certain or, generally, querying the instances that are more informative in a given dataset. In this way, it can establish a "line of inquiry" that can accelerate the generalization process for a particular problem [19] while avoiding the costs of unnecessary labels. In a pool-based scenario [3], an AL strategy must be able to select the most informative instance(s) from a pool of unlabeled data $\mathcal{U}$, so that

the oracle can be queried about its true label. However, the approach to querying the oracle directly depends on the strategy employed to characterize an informative instance.

More formally, given the instance space $\mathcal{X}$ and the class set $\mathcal{Y}$, consider the dataset $\mathcal{D} = \{\langle \mathbf{x}_i, y_i \rangle \ | \mathbf{x}_i \in \mathcal{X} \wedge y_i \in \mathcal{Y}\}$, which maps instances in $\mathcal{X}$ to their respective classes in $\mathcal{Y}$. Additionally, let there be a pool of unlabeled instances $\mathcal{U} \subset \mathcal{X}$ and a labeled set of instances $\mathcal{L} \subset \mathcal{D}$. A generic AL query strategy can then be described by Equation 1.

$$\mathbf{x}^S = \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \, S(\mathbf{x}_i) \tag{1}$$

In this case, $\mathbf{x}^S$ represents the most informative example in $\mathcal{U}$ based on some utility measure S. This utility measure assigns a degree of informativeness to all instances in the pool. Therefore, $\mathbf{x}^S$ can be queried for label annotation, and then a new labeled set $\mathcal{L}' = \langle \mathbf{x}^S, y^S \rangle \cup \mathcal{L}$ is produced. Hence, the set of strategies dependent on S is as diverse as there are different choices for S.

Based on the previous formulation, this section discusses several non-agnostic strategies in the literature and explains the rationale for including them in the analysis. Specifically, Section 2.1 provides a more detailed explanation of the main methods that comprise the Uncertainty Sampling strategy. While Section 2.2 discusses density-weighted methods, Section 2.3 presents the Expected Error Reduction strategy. Finally, Section 2.4, highlights other relevant strategies within PAL, as well as advancements made regarding the comparison of methods.

2.1. Uncertainty Sampling

Uncertainty Sampling (US) [3] is possibly the most popular AL strategy in practice [20]. Its core idea is to make the utility measure a function of the model's confidence. Therefore, to estimate the confidence of model θ, generated from $\mathcal{L}$, in its predictions, US-based methods use its predictive distribution P_θ. By doing so, the probability $P_\theta(y|\mathbf{x_i})$ represents the model's uncertainty regarding $\mathbf{x}_i$ having the label y. Consequently, the learner can avoid querying instances it is already confident about, allowing it to focus on the more challenging ones [20].

Equation 2 presents a basic query strategy that employs the Least Confident (LC) utility measure. Given that $\hat{y} = \operatorname{argmax}_y P_\theta(y|\mathbf{x}_i)$, this strategy aims to query instances whose predicted output is least confident [20].

$$\mathbf{x}^{LC} = \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \left[1 - P_\theta(\hat{y}|\mathbf{x}_i) \right] \tag{2}$$

However, in a multiclass problem, LC is limited in measuring the uncertainty of ambiguous instances, which have similar probabilities assigned for their most likely classes. Such a limitation is addressed by Margin Sampling (MS), introduced by Equation 3. This method takes into account the difference between the probabilities assigned to the two most likely classes, $\hat{y_1}$ and $\hat{y_2}$, respectively [20].

$$\mathbf{x}^{M} = \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \left[P_\theta(\hat{y_2}|\mathbf{x}_i) - P_\theta(\hat{y_1}|\mathbf{x}_i) \right] \tag{3}$$

Finally, the notion presented in MS can be generalized using the entropy function [21] as a utility measure. This leads to the Entropy Sampling (ES) method, expressed by Equation 4.

$$\mathbf{x}^{E} = \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \left[- \sum_{y \in \mathcal{Y}} P_\theta(y|\mathbf{x}_i) \log P_\theta(y|\mathbf{x}_i) \right] \tag{4}$$

Other PAL strategies also aim to select instances based on model uncertainty. Some of these approaches leverage specific models to exploit their properties for selecting instances, like SVM-based strategies [22, 23]. Others adopt a model-agnostic stance, such as Query by Committee [24], where multiple models are trained simultaneously, and the divergence among predictions reflects collective uncertainty over an instance. Nevertheless, the US strategy likely owes its popularity to being intuitive and easy to implement, while adding minimal overhead to the AL process [20]. Therefore, it is important to compare it with the strategy presented in this work to see if they behave similarly or if our strategy can overcome some of US's shortcomings, such as its poor performance when dealing with a small amount of labeled data.

2.2. Density Weighted Methods

Some AL strategies are limited by the fact that the instances are analyzed individually, running the risk of selecting poor query choices, such as outliers [25]. To mitigate this risk, Settles and Craven [16] presented the information density framework. The methods within this framework assume

that informative instances should not only have a high information content but also be representative of the data distribution [25].

Equation 5 presents a generic method for the framework. The main idea of this approach is to increase the informativeness value of the most representative instances in $\mathcal{U}$.

$$\mathbf{x}^{ID} = \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \left[\Phi_A(\mathbf{x}_i) \times \left(\frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}_j \in \mathcal{U} \setminus \{\mathbf{x_i}\}} sim(\mathbf{x}_i, \mathbf{x}_j) \right)^{\beta} \right] \tag{5}$$

Therefore, the utility $\Phi_A(\cdot)$ of an instance $\mathbf{x}_i$ according to method A should be weighted by the average similarity of $\mathbf{x}_i$ relative to the other unlabeled instances. Moreover, the function sim represents any similarity measure that can be derived from cosine similarity, Euclidean distance, Pearson's correlation coefficient, etc.

Equation 6 extends the generic method through the use of the training utility (TU) function [17]. In addition to considering the similarity between instances in $\mathcal{U}$, the strategy inversely weights the value of the instances by their similarity to labeled data. This creates a more exploratory bias, encouraging queries to shift away from previously queried regions.

$$\mathbf{x}^{TU} = \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \left[\Phi_{ID}(\mathbf{x}_i) \times \left(\frac{1}{|\mathcal{L}|} \sum_{\mathbf{x}_j \in \mathcal{L}} sim(\mathbf{x}_i, \mathbf{x}_j) \right)^{-\delta} \right] \tag{6}$$

Besides agnostic strategies like Hierarchical Sampling [26], and Coreset [27], other strategies aim to incorporate the structure of unlabeled data into the sampling process. For instance, Adaptive Active Learning [28] seeks to address this need through a self-adjusting mechanism, while Active Learning by Learning [29] is designed to choose the most appropriate strategy for a given problem. Strategies such as Graph Density [30], Querying Informative and Representative Examples [31] and Representative Sampling [32] also consider both informativeness and representativeness. However, given the simplicity of ID and TU, it is appropriate to evaluate the performance of any new strategy against them to determine whether the proposed approach offers a more effective solution.

2.3. Expected Error Reduction

Although effective in many cases, the presented strategies do not aim to optimize the learner's performance directly. In fact, uncertainty and density-based methods query instances that are considered to be informative regardless of whether labeling these instances will result in improved model performance. Conversely, methods based on expected error reduction (EER) [15] aim to select instances that, once labeled, have a higher probability of reducing the future error [33]. Thus, given that $P(y|\mathbf{x}_i)$ is the unknown conditional distribution of the input data $\mathbf{x}_i \in \mathcal{X}$ and classes $y \in \mathcal{Y}$ and $P(\mathbf{x}_i)$ is the distribution of the input data, the expected error of the learner can be defined in function of $P_\theta(y|\mathbf{x}_i)$ by Equation 7:

$$E_{P_\theta} = \int_{\mathbf{x}_i} L(P(y|\mathbf{x}_i), P_\theta(y|\mathbf{x}_i)) \times P(\mathbf{x}_i) \tag{7}$$

where L is any loss function responsible for measuring the degree of disappointment. However, not all distributions presented in the formulation above are known during the AL process. Therefore, instead of estimating the error over the complete distribution $P(\mathbf{x}_i)$, Roy and McCallum [15] propose measuring it over the sample in the pool. Additionally, they also suggest using the distribution $P_\theta(y|\mathbf{x}_i)$ to estimate $P(y|\mathbf{x}_i)$, so that the learner's expected error can be approximated by:

$$\tilde{E}_{P_\theta} = \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}_i \in \mathcal{U}} L(P_\theta(y|\mathbf{x}_i), P_\theta(y|\mathbf{x}_i)) \tag{8}$$

Based on this formulation, the EER strategy aims to select instances that have a lower expected value for the learner's expected error:

$$\mathbf{x}^{EER} = \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \left[- \sum_{y \in \mathcal{Y}} P_\theta(y|\mathbf{x}_i) \times \tilde{E}_{P_{\theta'}} \right] \tag{9}$$

where $P_{\theta'}$ represents the posterior distribution predicted by the learner θ' generated from the set $\mathcal{L}^+ = \mathcal{L} \cup \langle \mathbf{x}_i, y \rangle$. The main difference between EER methods is the loss function L. In their work, Roy and McCallum [15] adopted the binary loss and log loss functions (equations 10 and 11) as loss functions.

$$L_{bin} = \sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_i)(1 - \delta(y, \underset{y' \in \mathcal{Y}}{\operatorname{argmax}} P_\theta(y'|\mathbf{x}_i))) \tag{10}$$

$$L_{ent} = \sum_{y \in \mathcal{Y}} P(y|\mathbf{x}_i) \log(P_\theta(y|\mathbf{x}_i)) \tag{11}$$

In Equation 10, δ represents the Kronecker delta, which equals 1 if its two arguments are equal and 0 otherwise. Notably, the use of log loss in Equation 8 results in estimating the expected error based on the entropy of distribution $P_\theta(y|\mathbf{x}_i)$.

Other variants of EER aim to minimize the model's variance rather than the expected error [34]. Additionally, Konyushkova et al. [35], focuses on predicting the expected error reduction of an instance by treating the query procedure as a regression problem. Still, the near-optimal nature of EER makes it a significant benchmark to surpass. However, due to its computational complexity, the strategy can be infeasible in certain scenarios [33]. Therefore, a more practical strategy that still matches its performance would represent a notable improvement.

2.4. Novel Approaches and Comparative Experiments

Recent studies [36, 37] have introduced the use of meta-learning to enhance the process of selecting unlabeled instances for annotation. By leveraging experience acquired from previous tasks, meta-learning adapts the ML process to develop efficient models and solutions [38]. Additionally, Zhu et al. [39] proposed a novel approach that aims to reformulate AL by unifying the instance selection and model training stages to optimize a single objective for statistical learning.

Regarding strategy analysis, Pereira-Santos et al. [4] compared the performance of several strategies across a large number of datasets, considering different learning algorithms, and highlighted the existence of a relationship between the chosen strategy and the learning algorithm used. Conversely, Nguyen et al. [40] analyzed the use of a variety of uncertainty measures inside the US framework. Finally, Zhan et al. [41] seek to establish the limits of studies in PAL by developing a benchmark with a variety of datasets and a quantitative metric. Similarly, Lu et al. [42] present a new transparent and reproducible benchmark for the community, aiming to address the shortcomings of existing benchmarks.

3. Hardness Measures

In their work, Smith et al. [5] aim to identify a property that makes an instance difficult to classify in ML problems. This property, referred to as IH, is based on the probability of an instance being misclassified by a generic classifier [5]. Formally, given the data set $\mathcal{D}$, we define a hypothesis as a function of the type $h : \mathcal{X} \rightarrow \mathcal{Y}$, with $\mathcal{H}$ being the set of all possible hypotheses. Thus, Equation 12 defines the hardness value of an instance $\langle \mathbf{x_i}, y_i \rangle \in t$ concerning a given hypothesis h.

$$IH_h(\langle \mathbf{x_i}, y_i \rangle) = 1 - p(y_i|\mathbf{x_i}, h) \tag{12}$$

This value is essentially determined by the probability that the instance is classified incorrectly by the hypothesis in question, given that its true label is known. However, Smith et al. [5] suggest that the dependence of IH based on a specific hypothesis could be hypothetically lessened by aggregating IH values across all members of the set $\mathcal{H}$ in order to better understand what affects IH in general. This is achieved in Equation 13 by summing IH values associated with each hypothesis, weighted by their probabilities.

$$IH(\langle \mathbf{x_i}, y_i \rangle) = 1 - \sum_{h \in \mathcal{H}} p(y_i|\mathbf{x_i}, h) p(h|\mathcal{D}) \tag{13}$$

However, since hypothesis h would be generated by an algorithm a, applied to the data set $\mathcal{D}$, using a set of hyperparameters α (i.e. $h = a(\mathcal{D}, \alpha)$), calculating the presented IH measure becomes impractical due to its dependence on the entire set $\mathcal{H}$ — specifically, all combinations between algorithms and respective hyperparameters [5]. Notwithstanding, Smith et al. [5] assume that IH can be estimated by focusing on a carefully selected set of algorithms (and hyperparameters) $\mathcal{A}$. Accordingly, Equation 14 defines the hardness of an instance concerning the set $\mathcal{A}$, where a_j represents a selected algorithm and α_j its respective hyperparameters.

$$IH_{\mathcal{A}}(\langle \mathbf{x_i}, y_i \rangle) = 1 - \frac{1}{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{A}|} p(y_i|\mathbf{x_i}, a_j(\mathcal{D}, \alpha_j)) \tag{14}$$

It is important to notice that, although the elements in $\mathcal{A}$ are selected based on their utility and adoption degree, the set is constantly evolving and there is no definitive solution for it [5]. Based on this, we will use the terms IH and $\text{IH}_{\mathcal{A}}$ interchangeably through this work, for simplicity's sake.

While calculating IH is ideal for identifying instances that are hard to classify, it does not explain hardness. Therefore, Smith et al. [5] also proposed a set of HMs that assess aspects contributing to why an instance might be misclassified, thereby offering key insights for its analysis, detection, and processing [5]. Additionally, Arruda et al. [7] expanded this set of measures by adapting metrics originally designed to assess dataset complexity in classification problems [43] to the domain of IH. Furthermore, Lorena et al. [6] reviewed these measures and categorized them into 5 categories, as shown in Table 1. Accordingly, this section will briefly describe these categories and their corresponding measures.

Category	**Measure**	**Acron.**
Neighborhood	k-Disagreeing Neighbors	*kDN*
	Frac. nearby instances different class at instance-level	*N1I*
	Ratio of intra-extra class distances at instance level	*N2I*
	Local set cardinality at instance-level	*LSCI*
	Local set radius	*LSR*
	Usefulness	*U*
	Harmfulness	*H*
Likelihood	Class Likelihood	*CL*
	Class Likelihood Difference	*CLD*
Feature-based	Frac. features in overlapping areas	*F1I*
	Min. distance to overlapping areas of features	*F2I*
	Mean distance to overlapping areas of features	*F3I*
	Max. distance to overlapping areas of features	*F4I*
Tree-based	Disjunct Size	*DS*
	Disjunct Class Percentage	*DCP*
	Tree Depth (pruned)	TD_p
	Tree Depth (unpruned)	TD_u
Class Balance	Class Balance	*CB*
	Majority Value	*MV*

Table 1: Hardness measures as categorized by Lorena et al. [6].

3.1. Neighborhood-based

Neighborhood-based measures rely on the instance's neighbors to determine its hardness. In more detail, instances surrounded by examples from other classes in the input space are harder to classify than those located in regions with a higher density of their own class [6].

A prime example of a neighborhood-based measure is the k-disagreeing neighbors (kDN) measure [5]. This measure represents the percentage of

the k-nearest neighbors of an instance that do not share the same label, as described by Equation 15, where $kNN(\mathbf{x}_i)$ denotes the set of k-nearest neighbors of instance $\mathbf{x}_i$ and y_i represents the label of instance $\mathbf{x_i}$.

$$kDN(\mathbf{x_i}) = \frac{|\{\mathbf{x_j}|\mathbf{x_j} \in kNN(\mathbf{x_i}) \wedge y_j \neq y_i\}|}{k} \tag{15}$$

Similarly, Arruda et al. [7] took advantage of the fraction of nearby instances of different classes at an instance level to produce the N1I measure. Unlike kDN, this measure employs a minimum spanning tree MST constructed from $\mathcal{D}$, where an instance $\mathbf{x_i}$ corresponds to a vertex, and two instances are connected based on their distance. Given that nearby elements in the input space are likely to be connected in this structure, N1I returns the percentage of instances from different classes among the connections of $\mathbf{x_i}$ in the tree [6] as shown in Equation 16. Furthermore, high values for $N1I(\mathbf{x_i})$ indicate that $\mathbf{x_i}$ is close to instances from a different class, either because it lies on a classification boundary or due to being a noisy instance.

$$N1I(\mathbf{x_i}) = \frac{|\{\mathbf{x_j}|(\mathbf{x_i}, \mathbf{x_j}) \in MST(\mathcal{D}) \wedge y_j \neq y_i\}|}{|\{\mathbf{x_j}|(\mathbf{x_i}, \mathbf{x_j}) \in MST(\mathcal{D})\}|} \tag{16}$$

Given that the instance $\mathbf{x_j}$ closest to $\mathbf{x_i}$, whose class y_j differs from y_i, is called the *nearest enemy* of $\mathbf{x_i}$ ($NE(\mathbf{x_i})$), the N2I measure [7] considers the ratio of the intra-class and extra-class distances, defined by the function presented in Equation 17. In this context, d represents a distance function (e.g. Euclidean distance), while $NN(\mathbf{x}_i) \in y_i$ denotes the nearest example to $\mathbf{x_i}$ belonging to the same class.

$$\text{IntraInter}(\mathbf{x_i}) = \frac{d(\mathbf{x_i}, NN(\mathbf{x_i}) \in y_i)}{d(\mathbf{x_i}, NE(\mathbf{x_i}))} \tag{17}$$

Based on that, Equation 18 defines the N2I measure relying on the value of Equation 17 for $\mathbf{x_i}$. This approach ensures that its computed value falls within the $[0, 1]$ interval and that higher values represent instances that are harder to classify. Furthermore, the metric suggests that the harder instances are those closer to examples from opposing classes (enemies) or farther away from instances of its own class [6].

$$N2I(\mathbf{x_i}) = 1 - \frac{1}{\text{IntraInter}(\mathbf{x_i}) + 1} \tag{18}$$

In addition to the HMs presented so far, more measures can be proposed based on the local set concept. Equation 19 describes the local set of an instance $\mathbf{x_i}$ as the set of instances whose distance to $\mathbf{x_i}$ is smaller than the distance of $\mathbf{x_i}$ to its nearest enemy.

$$LS(\mathbf{x_i}) = \{\mathbf{x_j} | d(\mathbf{x_i}, \mathbf{x_j}) < d(\mathbf{x_i}, NE(\mathbf{x_i}))\} \tag{19}$$

From this conception, a natural extension is to use the cardinality of the $LS(\mathbf{x_i})$ set to measure the hardness of instance $\mathbf{x_i}$. Hence, the local set cardinality at an instance level (LSCI) measure [7], defined in Equation 20, is based on the complement of the relative cardinality of an instance's local set [6]. As a result, an easy instance will present lower values for LSCI, since most members of class y_i will belong to its local set.

$$LSCI(\mathbf{x_i}) = 1 - \frac{|LS(\mathbf{x_i})|}{|\{\mathbf{x_j} | y_i = y_j\}|} \tag{20}$$

A smoother version of LSCI is the local set radius (LSR) measure [7]. It takes the normalized radius of the local set, by calculating the distance of $\mathbf{x_i}$ to its nearest enemy:

$$LSR(\mathbf{x_i}) = 1 - \min\left\{1, \frac{d(\mathbf{x_i}, NE(\mathbf{x_i}))}{\max(d(\mathbf{x_i}, \mathbf{x_j}) | y_i = y_j)}\right\} \tag{21}$$

In contrast to LSCI, the usefulness (U) measure [7] considers the relationship of $\mathbf{x_i}$ with the local sets of other instances in its class. Equation 22 presents the measure as $U(\mathbf{x_i})$, which corresponds to the complement of its usefulness, i.e. the fraction of instances that include $\mathbf{x_i}$ in their local set [6]. This approach assumes that more useful instances are easier to classify, as they are closer to instances of their own class. Thus, high values of U can indicate outliers in the data set [6].

$$U(\mathbf{x}_i) = 1 - \frac{|\{\mathbf{x_j} | d(\mathbf{x_i}, \mathbf{x_j}) < d(\mathbf{x_j}, NE(\mathbf{x_j})\}|}{|\{\mathbf{x_j} | y_i = y_j\}|} \tag{22}$$

Finally, the harmfulness (H) measure [7], defined in Equation 23 is related to the number of instances that have $\mathbf{x_i}$ as their nearest enemy. In this definition, outliers located in regions dominated by examples of another class and instances near classification boundaries will tend to have higher H values and are therefore harder to classify [6].

$$H(\mathbf{x_i}) = \frac{|\{\mathbf{x_j}|NE(\mathbf{x_j}) = \mathbf{x_i})\}|}{|\{\mathbf{x_j}|y_i \neq y_j\}|} \tag{23}$$

3.2. Likelihood-based

Class likelihood-based HMs leverage the degree of similarity between an instance and the general patterns associated with its class to assess its hardness [6]. These measures use the posterior probability $P(\mathbf{x}_i|y_i)$ to perform this calculation. This probability, in turn, is calculated similarly to the approach used in the Naive Bayes classification algorithm, where the likelihood of an instance belonging to a class is derived from the probability distributions generated for each of its features, which are analyzed independently [44, 6].

Therefore, Equation 24 presents the class likelihood (CL) measure [5], which identifies instances with a low probability of belonging to their own class as hard instances. In this setting, for a given number of classes c, $P(y_i)$ is the prior of class y_i, and it is assumed to be $\frac{1}{c}$ for all instances. When this posterior probability is low, it indicates that the instance $\mathbf{x_i}$ does not conform to the patterns exhibited by most instances in class y_i. Consequently, $\mathbf{x_i}$ is considered hard to classify, resulting in a high $CL(\mathbf{x}_i)$ value.

$$CL(\mathbf{x}_i) = 1 - P(\mathbf{x}_i|y_i)P(y_i) \tag{24}$$

Similarly, another approach known as class likelihood difference (CLD) [5] is described by Equation 25, where the conditional probabilities are also estimated by considering each of the input features independent from each other. This measure involves analyzing the difference between the likelihood of $\mathbf{x}_i$ belonging to its own class and the maximum likelihood of $\mathbf{x}_i$ belonging to any other class. Therefore, easier instances are those that not only align with the patterns expressed by the majority of cases in their class but also lack significant similarity with instances from any other class [6]. Additionally, according to Smith et al. [5], the prior presented in equations 24 and 25 can be ignored to prevent class skewness from influencing the results, although the overall effect remains [6].

$$CLD(\mathbf{x}_i) = \frac{1 - (P(\mathbf{x}_i|y_i)P(y_i) - \max_{y_j \neq y_i}[P(\mathbf{x}_i|y_j)P(y_j)])}{2} \tag{25}$$

3.3. Feature-based

Feature-based HMs try to quantify the fact that instances with feature values outside the overlapping regions of the classes are generally easy to

classify [6]. With this in mind, Arruda et al. [7] suggested using the fraction of features in overlapping areas to measure the hardness of an instance through the F1I measure, as presented in Equation 28. In this formulation, m corresponds to the number of features. At the same time, x_{ij} represents the value of the j-th feature of instance $\mathbf{x}_i$ and $\mathbf{f}_j$ is the vector containing all values taken by this feature in the dataset. Nevertheless, the limits of the overlapping areas for feature j are obtained by the functions defined in equations 26 and 27, where $\max(\mathbf{f}_j^{y_i})$ and $\min(\mathbf{f}_j^{y_i})$ are the maximum and minimum values in $\mathbf{f}_j$ for class $y_i \in c_1, c_2$, respectively. Therefore, higher values of F1I are assigned to instances with many features lying in overlapping regions.

$$\text{min\_max}(\mathbf{f}_j) = \min(\max(\mathbf{f}_j^{c_1}), \max(\mathbf{f}_j^{c_2})) \tag{26}$$

$$\text{max\_min}(\mathbf{f}_j) = \max(\min(\mathbf{f}_j^{c_1}), \min(\mathbf{f}_j^{c_2})) \tag{27}$$

$$F1I(\mathbf{x}_i) = \frac{\sum_{j=1}^{m} \delta(x_{ij} \geq \text{max\_min}(\mathbf{f}_j) \wedge x_{ij} \leq \text{min\_max}(\mathbf{f}_j))}{m} \tag{28}$$

Additionally, three other measures proposed by Arruda et al. [7] rely on calculating the distance of each instance from the central points of the feature overlap regions. This calculation is accomplished by using the formula presented in Equation 29, which is transformed by Equation 30 to ensure that maximum hardness values are obtained for instances located at the center of an overlapping region.

$$d_o(\mathbf{x}_i, \mathbf{f}_j) = \frac{\text{min\_max}(\mathbf{f}_j) - x_{ij}}{\text{min\_max}(\mathbf{f}_j) - \text{max\_min}(\mathbf{f}_j)} \tag{29}$$

$$d_o^t(\mathbf{x}_i, \mathbf{f}_j) = \frac{1}{(1 + |0.5 - d_o(\mathbf{x}_i, \mathbf{f}_j)|)} \tag{30}$$

Thus, the F2I, F3I e F4I measures [7] are presented by equations 31, 32 and 33. They calculate the minimum, average, and maximum distances of instance $\mathbf{x}_i$ to the center of the overlapping regions.

$$F2I(\mathbf{x}_i) = \min_{j=1}^{m} d_o^t(\mathbf{x}_i, \mathbf{f}_j) \tag{31}$$

$$F3I(\mathbf{x}_i) = \frac{1}{m}\sum_{j=1}^{m} d_o^t(\mathbf{x}_i, \mathbf{f}_j) \tag{32}$$

$$F4I(\mathbf{x}_i) = \max_{j=1}^{m} d_o^t(\mathbf{x}_i, \mathbf{f}_j) \tag{33}$$

Finally, although feature-based measures assume the presence of only two classes, multiclass problems can be addressed by using the one-versus-one strategy [45].

3.4. Tree-based

Tree-based HMs are derived from some decision tree [46] DT generated from $\mathcal{D}$. These measures are based on the assumption that harder instances lead to a greater number of decisions to classify them and, consequently, a larger number of splits in the tree [6]. In this context, the disjunct size (DS) measure [5] stipulates IH based on the relative size of the disjunct (leaf node) where this instance is placed. This is presented in Equation 34, where $Disjunct(\mathbf{x}_i)$ represents the set of instances contained in the leaf node where $\mathbf{x}_i$ is located. Following this definition, easier instances are located in larger disjuncts, while the harder ones tend to fall into smaller disjuncts due to the greater number of splits required for correctly classifying them.

$$DS(\mathbf{x}_i) = 1 - \frac{|Disjunct(\mathbf{x}_i)|}{\max_{\mathbf{x_j}\in\mathcal{D}} |Disjunct(\mathbf{x}_j)|} \tag{34}$$

On the other hand, the disjunct class percentage (DCP) measure [5], is obtained by fitting a pruned decision tree on $\mathcal{D}$ and estimating the percentage of instances that have the same class as $\mathbf{x}_i$ in its disjunct, as stated in Equation 35. In this manner, instances that are easier to classify tend to form a clear majority within the disjunct used for their classification, thereby exhibiting lower DCP values [6].

$$DCP(\mathbf{x}_i) = 1 - \frac{|\{\mathbf{x}_j | \mathbf{x}_j \in Disjunct(\mathbf{x}_j) \wedge y_j = y_i\}|}{|Disjunct(\mathbf{x}_i)|} \tag{35}$$

Finally, the tree depth (TD) measure [5] considers the depth of the node that classifies $\mathbf{x}_i$ in the decision tree DT. This depth is calculated by the $depth_{DT}(\mathbf{x}_i)$ function, which is then normalized by the maximum depth of the tree, as shown in Equation 36. In this context, the hardness of an instance

is determined by the proximity of the disjunct classifying it to the deepest level of the tree. Therefore, instances closer to this level exhibit higher TD values [6]. Moreover, this measure can be presented in two versions, TD_P and TD_U, which differ based on whether DT is pruned or unpruned.

$$TD(\mathbf{x}_i) = \frac{depth_{DT}(\mathbf{x}_i)}{\max_{\mathbf{x}_j \in \mathcal{D}}(depth_{DT}(\mathbf{x}_j))} \tag{36}$$

3.5. Class Balance

Another factor to consider when it comes to IH is the distribution of the instance's class in the dataset. Instances belonging to a minority class are considered harder to classify, as they are more susceptible to classification errors, while instances belonging to a majority class tend to be easier to classify [6]. Therefore, Smith et al. [5] introduced two measures that characterize the difficulty of an example based on its class distribution.

Equation 37 presents the class balance (CB) measure [5] adapted by Lorena et al. [6] to fit within the $[0, 1]$ interval in such a way that harder instances have higher CB values. This metric takes into account all C classes that make up the dataset $\mathcal{D}$, interpreting all instances in the dataset as very easy to classify (e.g. $CB(\mathbf{x}_i) \approx 0$) if the problem is balanced.

$$CB(\mathbf{x}_i) = \left(1 - \frac{|\{\mathbf{x}_j | y_j = y_i\}|}{n} + \frac{1}{C}\right)\left(\frac{C}{C+1}\right) \tag{37}$$

In contrast, the majority value (MV) measure [5], presented in Equation 38, considers only the proportion between the number of instances that have the same class as $\mathbf{x}_i$ and the number of instances belonging to the most representative class in $\mathcal{D}$. As a result, higher values are assigned to points belonging to rarer classes.

$$MV(\mathbf{x}_i) = 1 - \frac{|\{\mathbf{x}_j | y_j = y_i\}|}{\max_{y_i \in \mathcal{Y}} |\{\mathbf{x}_j | \mathbf{x}_j \in y_j\}|} \tag{38}$$

4. Hardness Sampling

To explore the use of HMs in the context of PAL, we propose HardS, a new query strategy designed to sample instances that are potentially hard to classify. This strategy extends Equation 1 by making use of some hardness

measure HM as a utility function representing the degree of informativeness of a given instance $\mathbf{x}_i$, as stated in Equation 39.

$$\mathbf{x}^{HardS} = \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}}\ HM(\mathbf{x}_i) \tag{39}$$

Notwithstanding, it is important to notice that all HMs presented in Section 3 rely on knowledge about the true label y_i of instance $\mathbf{x}_i$, which limits their applicability in PAL. To overcome this issue, we have opted to use the predictions made by the active learner as instance labels. By doing so, we expect HardS to sample instances based on the model's assumptions about the hardness of unlabeled data. Moreover, the hardness degree of instance $\mathbf{x}_i$ depends not only on its predicted label $\hat{y}_i$, but also on the labels $\hat{y}_j$ assigned to all other instances $\mathbf{x}_j \in \mathcal{U}$. Consequently, the proposed strategy distinguishes itself from most classical approaches, presented in Section 2, by considering model predictions within their complex interdependence — not in isolation. Rather than using the predicted label likelihoods directly, the strategy leverages these labels to compute a certain HM, reflecting a more integrated use of predictions. Therefore, in the sampling context, the confidence concerning an instance's true label is less relevant than its potential hardness, the degree of hardness for any model to correctly classify an example, given that the learner's assumptions about the problem are correct. Although this approach may be more susceptible to bias introduced by the learner, the resulting shift in the model's interpretation over successive iterations is expected to positively impact its application in PAL.

Another constraint that arises from the application of HMs in query strategies is that they assume that the problem at hand involves more than one class. However, when it comes to AL, the learner is prone to classify all unlabeled instances as belonging to the same class. This happens more frequently in the early stages of the process when the initial amount of information is limited. To address this issue, random sampling (Rnd) was adopted as a fallback method for cases where the learner's predictions prevent HMs from being extracted. Since this method is based only on the random selection of unlabeled instances, it avoids adding more bias to the process. As a result, Rnd is frequently employed as a baseline when comparing different PAL methods.

5. Methodology and Experimental Configuration

To appropriately evaluate HardS, comparing it against some benchmarks is crucial. To this end, we assessed its performance alongside methods employing diverse strategies across various datasets and learning algorithms. Specifically, we adopted a collection of datasets represented by $\mathcal{S}$, a set $\mathcal{A}$ of learning algorithms, and a set $\mathcal{Q}$ of distinct PAL methods. As a result, the tuple $(\mathcal{D}, a, q) \in \mathcal{S} \times \mathcal{A} \times \mathcal{Q}$ represents a specific configuration.

Regarding the collection $\mathcal{S}$, we chose one similar to that employed in the work of Pereira-Santos et al. [4]. This selection evaluated the strategies under various conditions, such as dataset size and class imbalance. In particular, 90 datasets from the UCI Machine Learning Repository [47] were chosen. Table 2 provides an overview of such datasets and their key characteristics.

The classification problems ranged from 2 to 30 classes, with the majority containing up to 5. Dataset sizes vary significantly, spanning from hundreds to tens of thousands of instances. In terms of attributes, the datasets feature between 2 and 167 attributes, with most consisting primarily of numerical attributes. To ensure compatibility with the learning algorithms employed, nominal attributes were converted to numerical form through one-hot encoding.

As for the set $\mathcal{A}$, four learning algorithms were deliberately selected for their diversity. The chosen algorithms were Gaussian Naive Bayes (NB) [48]; Support Vector Machines (SVM) [49], with the RBF kernel, $C = 1$ and γ set to the inverse of the product between the number of features and the variance of the data; K-nearest Neighbors [48] with $k = 5$ (5NN) and Euclidean distance as the distance metric; and Classification and Regression Trees (CART) [50] with no pruning and the Gini impurity as the splitting criterion. The default hyperparameters from the adopted implementation were kept for each algorithm.

Finally, the $\mathcal{Q}$ set was structured to include methods representing various classical strategies, along with those introduced by the HardS strategy. More precisely, the set can be decomposed as $\mathcal{Q} = \mathcal{Q}_C \cup \mathcal{Q}_{HardS}$, where $\mathcal{Q}_C$ contains methods associated with classical strategies, and $\mathcal{Q}_{HardS}$ encompasses the methods derived from HardS. To form $\mathcal{Q}_C$, methods representing strategies discussed in Section 2 were gathered. For the US strategy, the MS method, presented in Equation 3, was chosen. In the domain of density-weighted methods, ID and TU (equations 5 e 6) were selected, with both parameters α and β set to 1, along with the use of Euclidean distance to measure sim-

Name	#in	#cl	#at	#no	Name	#in	#cl	#at	#no
1-abalone-3class	4177	3	8	1	46-ozone-eighthr	2534	2	72	0
2-artificial-charac...	10218	10	7	0	47-page-blocks	5473	5	10	0
3-autoUniv-au1-1000	1000	2	20	0	48-parkinsons	195	2	22	0
4-autoUniv-au6-cd1-...	400	8	40	3	49-pendigits	10992	10	16	0
5-autoUniv-au7-300-...	1100	5	12	4	50-phoneme	5404	2	5	0
6-autoUniv-au7-700	700	3	12	4	51-pima-indians-dia...	768	2	8	0
7-autoUniv-au7-cpd1...	500	5	12	4	52-qsar-biodegradat...	1055	2	41	0
8-balance-scale	625	3	4	0	53-ringnorm	7400	2	20	0
9-banana	5300	2	2	0	54-robot-failure-lp5	164	5	90	0
10-banknote-authent...	1372	2	4	0	55-robot-nav-sensor...	5456	4	2	0
11-bupa	345	2	6	0	56-saheart	462	2	9	1
12-car-evaluation	1728	4	6	6	57-seeds	210	3	7	0
13-cardiotocography...	2126	3	35	0	58-spambase	4601	2	57	0
14-climate-simulati...	540	2	20	0	59-spect-heart	267	2	22	22
15-connectionist-mi...	208	2	60	0	60-statlog-australi...	690	2	14	6
16-connectionist-vo...	990	11	13	0	61-statlog-german-c...	1000	2	20	13
17-ecoli	336	8	7	0	62-statlog-heart	270	2	13	0
18-eeg-eye-state	14980	2	14	0	63-statlog-image-se...	2310	7	18	0
19-first-order-theo...	6118	6	51	0	64-statlog-vehicle-...	846	4	18	0
20-flare	1389	6	12	2	65-steel-plates-fau...	1941	2	33	0
21-glass	214	6	9	0	66-systhetic-control	600	6	60	0
22-habermans-survival	306	2	3	0	67-texture	5500	11	40	0
23-heart-disease-pr...	303	5	13	2	68-thyroid-ann	3772	3	21	0
24-heart-disease-pr...	294	2	13	0	69-thyroid-hypothyr...	3163	2	25	18
25-heart-disease-pr...	200	5	13	0	70-thyroid-newthyroid	215	3	5	0
26-hepatitis	155	2	19	13	71-thyroid-sick-eut...	3163	2	25	18
27-hill-valley-with...	1212	2	100	0	72-tic-tac-toe	958	2	9	9
28-horse-colic-surg...	300	2	27	14	73-turkiye-student	5820	13	32	0
29-indian-liver-pat...	583	2	10	1	74-twonorm	7400	2	20	0
30-ionosphere	351	2	33	0	75-user-knowledge	403	5	5	0
31-iris	150	3	4	0	76-vertebra-column-2c	310	2	6	0
32-kr-vs-kp	3196	2	36	36	77-vertebra-column-3c	310	3	6	0
33-leaf	340	30	15	0	78-volcanoes-a3	1521	5	3	0
34-lymphography	148	4	18	15	79-volcanoes-b5	9989	5	3	0
35-magic	19020	2	10	0	80-volcanoes-d1	8753	5	3	0
36-mammographic-mass	961	2	5	0	81-volcanoes-e1	1183	5	3	0
37-mfeat-fourier	2000	10	76	0	82-voting	435	2	16	16
38-molecular-splice...	3190	3	60	60	83-waveform-v2	5000	3	40	0
39-monks1	556	2	6	0	84-wdbc	569	2	30	0
40-monks3	554	2	6	0	85-wholesale-channel	440	2	7	0
41-movement-libras	360	15	90	0	86-wilt	4839	2	5	0
42-mushroom	8124	2	21	21	87-wine	178	3	13	0
43-musk	6598	2	167	1	88-wine-quality-red	1599	6	11	0
44-nursery	12960	5	8	8	89-wine-quality-whi...	4873	5	11	0
45-optdigits	5620	10	62	0	90-yeast-4class	1299	4	8	0

Table 2: Datasets used in the experiments. The first column shows the dataset name, followed by columns that represent the number of instances, the number of classes, the total number of attributes, and the number of nominal attributes.

ilarity between instances. Moreover, the EER strategy was represented by the EER_{ent} method, which minimizes the entropy-based loss function defined in Equation 11. As a baseline, Rnd was included to assess whether any of the methods presented offered an advantage over the random selection of unlabeled instances. In addition, $\mathcal{Q}_{HardS}$ was composed through the use of HMs presented in Table 1 as utility measures, as outlined in Equation 39.

Therefore, Figure 1 shows the steps taken to evaluate a given configuration $(\mathcal{D}, a, q)$. First, the dataset $\mathcal{D}$ was split into $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$, where the training set $\mathcal{D}_{train}$ was employed to simulate the PAL process, while the test set $\mathcal{D}_{test}$ served to evaluate its respective performance. To ensure a more re-

liable evaluation of each configuration, stratified 5-fold cross-validation was adopted, where one fold was used for testing and the remaining folds for training.

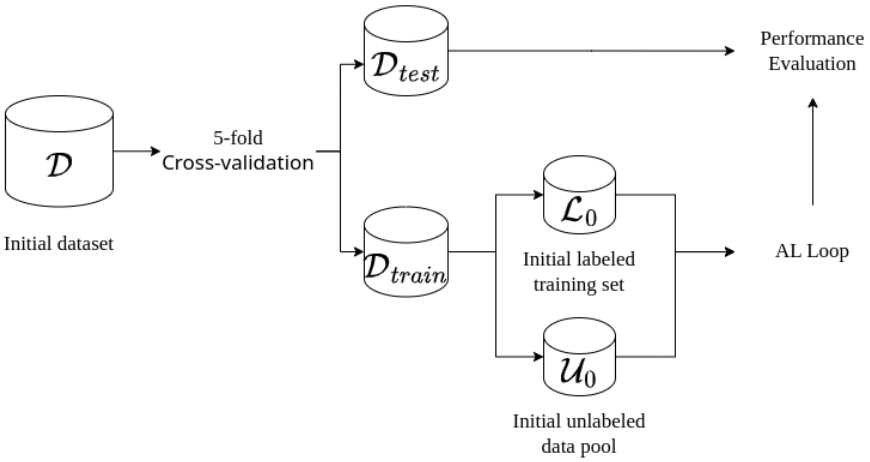

Figure 1: Diagram illustrating the steps to perform the experiment on dataset $\mathcal{D}$.

Regarding the initial set of labeled data $\mathcal{L}_0$, it was established that it should contain at least one example from each class, such that $|\mathcal{L}_0| = \max(5, c)$. Moreover, the decision to have a minimum of 5 labeled instances in total — regardless of class — at the beginning of the PAL process was influenced by the inclusion of the 5NN algorithm in the set $\mathcal{A}$.

Given the initial conditions for each configuration, the AL loop was carried out over 100 iterations. For each iteration i, where $0 \leq i < 100$, a new instance $\mathbf{x}^q$ was selected from the pool $\mathcal{U}_i$ based on method q. Subsequently, the label y^q for this instance was revealed to mimic the annotation process, resulting in an updated labeled dataset $\mathcal{L}_{i+1} = \mathcal{L}_i \cup \{(\mathbf{x}^q, y^q)\}$.

As a new labeled dataset $\mathcal{L}_i$ was obtained, a new model θ_i was induced by algorithm a. The performance of θ_i was then evaluated on $\mathcal{D}_{test}$ using Cohen's Kappa coefficient [51], denoted as κ_i, which ranges from -1 to 1. This metric was chosen due to its robustness, as it accounts for the possibility of θ_i achieving correct predictions by chance while considering the true and predicted labels as distinct distributions [52]. Values close to 1 indicate strong agreement between the model's predictions and the true labels, while values near 0 suggest that any similarity is likely due to chance. Moreover, negative values reflect a disagreement between predictions and actual labels.

Accordingly, the performance scores were recorded for $i \in [1, 100]$, with κ_0 not being recorded. After evaluating a single configuration, 100 performance records were available for each test fold. Consequently, the κ_i values for all folds could be summarized by $\bar{\kappa}_i$, which represents the average performance of the learner on the i-th query of the AL loop. From that, it was possible to identify the performance associated with a certain configuration given the number of queries made. Moreover, we were able to obtain the learning curve associated with a specific configuration by treating $\bar{\kappa}_i$ as a function of the number of queries made.

To ensure a more accurate comparison, we chose to use the average rank as an evaluation metric, similar to the approach taken by Pereira-Santos et al. [4]. Although measures like the area under the learning curve (ALC) are useful for assessing individual configurations, they are not ideal for evaluating overall performance across the entire collection. This is because comparisons of average ALC values obtained by a pair method-algorithm can be unreliable due to the varying difficulty levels presented by each dataset [4]. Additionally, ranking curves [18] were employed to visually compare the performance of different methods applied with various learners across multiple datasets.

The experiments were conducted on the Ubuntu 22.04 LTS GNU/Linux operating system, with kernel version 6.8.0-36-generic, using Python 3.8.19. The learning algorithms were sourced from the `scikit-learn` library [53], while most of the methods in $\mathcal{Q}_C$ were implemented by the `modAL` [54] module. Additional extensions were developed to fully implement the TU and ID methods. The HardS strategy was applied using `modAL` in combination with the HMs provided by the `PyHard` package [10]. All experiments used the default parameters from the respective libraries, ensuring consistency in the implementation.

6. Results and Discussion

In this Section, we analyze the performance scores for each method-algorithm pair evaluated. First, we split the presentation into a general, higher-level quantitative analysis (Section 6.1) and a more detailed qualitative view of how the methods' performance curves compare visually to each other (Section 6.2). The former focuses on average rank positions as the end result of PAL, while the latter focuses on the number of queries to reach competitive prediction performance earlier in the PAL process. In Section 6.3, we highlight the relationship between HMs groups and learning algorithms

through an extended version of the ranking curves showing details like the best and worst performers within each group of interest. Finally, Section 6.4 makes a general overview by explaining the observed behaviors, concluding with a hypothesis on how HardS may contribute to regulating sampling bias in PAL.

6.1. General Analysis

To establish an overview of the performance of each method-algorithm pair, we begin by summarizing their ranks across all queries and datasets. Therefore, Table 3 presents the average ranking and standard deviation for each pair, divided into two halves. For each algorithm, the best and worst rankings are highlighted in bold. For distinction purposes, the best ranks are also marked with an asterisk. The occurrences of random sampling, the baseline technique, are underlined for an easier comparison to the others. Additionally, for variants based on HardS, the Group column specifies the group to which the employed HM belongs. As for rows describing classical methods, the Group column is filled with the '-' character.

The table suggests that the choice of the learning algorithm significantly impacts performance. In greater detail, methods paired with CART and NB tended to achieve better average ranks than those paired with 5NN and SVM, regardless of the strategy in use. All pairs that employed CART secured positions in the first half of the table, with ranks falling between those presented by the TU and MS methods. In the case of NB, Rnd was depicted as the best approach whereas ID had the lowest performance, being one of the few methods to appear in the second half of the table for this algorithm. Conversely, most pairs containing 5NN fell into the second half of the table, with ID presenting the worst rank overall. However, a small number of exceptions is displayed at the bottom of the first half, where Rnd emerged as the best approach for 5NN. Although, all SVM pairs are located in the second half of the table, with N2I and ID achieving the best and worst average ranks, respectively.

With the exception of the N2I-SVM pair, all HardS variants had an average performance below the baseline. Nevertheless, they were still capable of outperforming at least one of their classical alternatives for a given algorithm. For CART, only TU and Rnd achieved better average ranks than those assigned to HardS. This pattern is demonstrated by a broad range of ranks displayed for CART in the first half of the table, which includes only HardS methods and is led by N2I, LSR, and kDN. For NB, a similar pattern

Avg. Rank Pos.	Method-Learner	Group	Avg. Rank Pos.	Method-Learner	Group
25.65 ± 23.38	**TU-CART***	-	48.41 ± 28.59	EER_{ent}-NB	-
26.18 ± 21.55	Rnd-CART	-	48.85 ± 29.75	**N2I-SVM***	Neighbor-based
27.09 ± 24.72	**Rnd-NB***	-	48.91 ± 27.81	CB-NB	Class-Balance
27.88 ± 24.38	N2I-CART	Neighbor-based	49.48 ± 24.54	MS-5NN	-
30.79 ± 26.76	TU-NB	-	50.65 ± 24.46	CLD-5NN	Likelihood-based
31.10 ± 23.17	LSR-NB	Neighbor-based	50.71 ± 24.19	H-5NN	Neighbor-based
31.81 ± 25.08	N2I-NB	Neighbor-based	51.18 ± 25.30	LSR-5NN	Neighbor-based
32.66 ± 25.23	LSR-CART	Neighbor-based	51.38 ± 28.94	MS-SVM	-
33.77 ± 24.11	kDN-CART	Neighbor-based	51.39 ± 24.07	kDN-5NN	Neighbor-based
34.54 ± 26.63	H-CART	Neighbor-based	53.57 ± 20.70	TD_P-5NN	Tree-based
35.23 ± 24.80	CLD-CART	Likelihood-based	53.61 ± 28.97	H-SVM	Neighbor-based
35.95 ± 24.38	kDN-NB	Neighbor-based	53.62 ± 29.47	Rnd-SVM	-
36.21 ± 23.32	TD_U-NB	Tree-based	53.70 ± 28.24	CLD-SVM	Likelihood-based
36.32 ± 23.38	N1I-CART	Neighbor-based	54.70 ± 29.10	**ID-NB**	-
36.32 ± 23.78	LSCI-CART	Neighbor-based	54.73 ± 24.23	DS-5NN	Tree-based
36.44 ± 24.69	CL-CART	Likelihood-based	54.73 ± 28.01	TD_U-SVM	Tree-based
36.46 ± 24.10	U-CART	Neighbor-based	54.75 ± 23.75	LSCI-5NN	Neighbor-based
36.68 ± 23.21	F4I-CART	Feature-based	54.92 ± 30.20	LSR-SVM	Neighbor-based
37.53 ± 24.62	LSCI-NB	Neighbor-based	55.30 ± 30.62	kDN-SVM	Neighbor-based
38.60 ± 27.20	H-NB	Neighbor-based	55.37 ± 24.39	CL-5NN	Likelihood-based
38.85 ± 25.74	U-NB	Neighbor-based	55.50 ± 24.61	N1I-5NN	Neighbor-based
39.83 ± 26.77	CLD-NB	Likelihood-based	55.67 ± 27.42	CL-SVM	Likelihood-based
39.91 ± 25.02	N1I-NB	Neighbor-based	56.37 ± 24.14	DCP-5NN	Tree-based
40.11 ± 23.98	MV-CART	Class-Balance	57.23 ± 23.60	U-5NN	Neighbor-based
40.54 ± 24.53	F2I-CART	Feature-based	57.85 ± 22.38	CB-5NN	Class-Balance
40.61 ± 24.16	CB-CART	Class-Balance	58.03 ± 23.82	F4I-5NN	Feature-based
41.30 ± 22.85	DS-CART	Tree-based	58.49 ± 28.21	DS-SVM	Tree-based
41.39 ± 25.92	DS-NB	Tree-based	58.51 ± 28.77	U-SVM	Neighbor-based
41.45 ± 24.43	DCP-NB	Tree-based	58.77 ± 25.52	TU-5NN	-
41.57 ± 25.28	TD_P-NB	Tree-based	58.91 ± 23.22	MV-5NN	Class-Balance
41.60 ± 23.60	TD_U-CART	Tree-based	59.00 ± 26.12	TD_P-SVM	Tree-based
41.65 ± 26.04	F4I-NB	Feature-based	59.17 ± 29.51	N1I-SVM	Neighbor-based
41.90 ± 27.03	CL-NB	Likelihood-based	59.92 ± 28.57	LSCI-SVM	Neighbor-based
42.06 ± 25.86	F3I-CART	Feature-based	60.84 ± 23.27	F3I-5NN	Feature-based
42.31 ± 22.12	F1I-CART	Feature-based	60.97 ± 23.84	F1I-5NN	Feature-based
42.82 ± 24.23	TD_P-CART	Tree-based	61.34 ± 23.92	F2I-5NN	Feature-based
43.27 ± 24.73	DCP-CART	Tree-based	61.63 ± 28.10	DCP-SVM	Tree-based
43.80 ± 25.94	**Rnd-5NN***	-	62.82 ± 22.87	EER_{ent}-5NN	-
44.43 ± 24.47	N2I-5NN	Neighbor-based	63.56 ± 24.77	EER_{ent}-SVM	-
45.26 ± 26.54	F3I-NB	Feature-based	63.66 ± 26.12	F4I-SVM	Feature-based
45.62 ± 28.48	MS-NB	-	64.37 ± 25.61	MV-SVM	Class-Balance
46.22 ± 24.93	EER_{ent}-CART	-	65.10 ± 24.02	CB-SVM	Class-Balance
46.29 ± 25.38	F2I-NB	Feature-based	65.11 ± 23.51	F2I-SVM	Feature-based
46.58 ± 26.88	ID-CART	-	65.94 ± 21.28	TU-SVM	-
47.81 ± 22.87	TD_U-5NN	Tree-based	66.10 ± 24.73	F1I-SVM	Feature-based
47.92 ± 26.87	F1I-NB	Feature-based	66.40 ± 22.75	F3I-SVM	Feature-based
47.95 ± 25.22	**MS-CART**	-	70.79 ± 21.14	**ID-SVM**	-
48.24 ± 24.72	MV-NB	Class-Balance	74.77 ± 21.44	**ID-5NN**	-

Table 3: Average ranking and standard deviation obtained by each method-algorithm pair across all datasets. The best and worst rankings are highlighted in bold face for each algorithm, while the best is also marked with an asterisk (*). Occurrences of the baseline method (random sampling) are underlined. For variants of the HardS strategy, the group to which the employed HM belongs is shown instead of a method name.

occurs: Rnd and TU held the top positions, followed by a series of HardS methods. In this case, LSR, N2I, and kDN ranked highest within this series. However, unlike with CART, this sequence is interrupted at the final positions by MS, as seen at the bottom of the table's first half. Even so, the remaining HardS methods managed to outperform EER or ID.

For 5NN, N2I achieved second place, with a slight difference from the baseline, followed by TD_U, in the third position. Although the remaining

HardS's methods were surpassed by MS, they consistently outperformed the remaining classical approaches. A small exception is observed with TU, which managed to surpass only the four lowest-ranked HardS methods. Finally, for SVM, two methods from our strategy — namely N2I and H — exceeded the baseline. Additionally, N2I emerged as an outstanding measure, placing MS in second. Although surpassed by Rnd, the remaining HardS methods consistently occupied the subsequent positions. At the final ranks, only six of our methods were surpassed by EER_{ent} or TU. Still, all of them managed to outperform ID.

6.2. Ranking Curves Analysis

Despite introducing a good overview of pairs' performance, Table 3 does not consider the various stages of PAL. To address this issue, figures 2 to 5 present ranking curves for the different learners tested. These curves complement the results by providing a graphical overview of the methods' behavior across all iterations of the AL loop. In all these figures, each line represents the average rank achieved by a method in a given stage, which is described by the number of queries already made. Although each figure refers to a specific learner, the mean ranking position presented on the y-axis pertains to the overall average ranking of a pair. Therefore, the integration of these graphs reflects the performance of all pairs tested in our experiments. In these figures, Rnd is depicted with dashed lines, while classical methods are shown with dash-dotted lines. Methods that compose HardS are represented by solid lines with markers, where curves sharing the same marker shape denote methods using HMs from the same group. Additionally, a moving average was applied to all curves to improve visualization. A sliding window width of 5 queries was chosen to provide a good balance between smoothness and detail preservation.

Figure 2 exhibits the ranking curves recorded for the CART algorithm. All curves show a declining trend, which is often expected due to the overall improvement of other competing methods as the training set increases in size. The competition between TU and Rnd for the top positions is evident. Additionally, N2I's curve remains relatively close to them, even securing the highest scores at the beginning and near the 80th query, while maintaining a considerable distance from the other curves. The steadier trend of N2I causes this gap to widen as the number of queries increases, allowing N2I to re-enter competition with Rnd and TU around the 75th query. In the

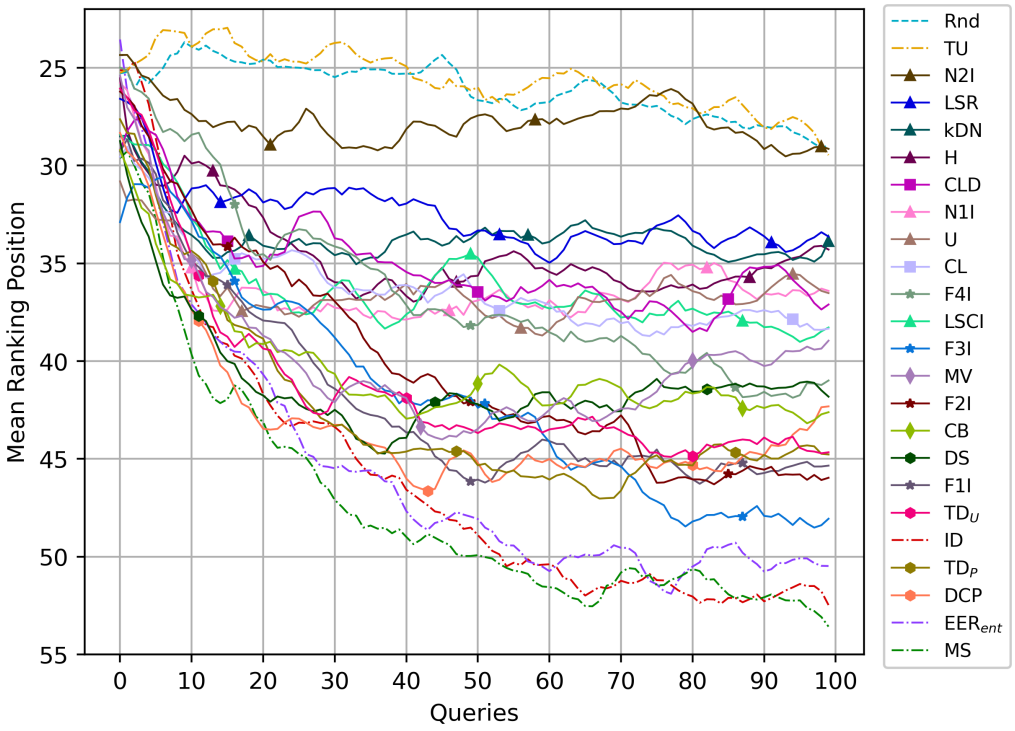

Figure 2: Ranking curves for methods tested with the CART algorithm. Curves with the same marker shape present HMs from the same group. Dash-dotted lines depict classical methods, and the cyan dashed line shows random sampling. The legend is sorted by the ALC scores.

end, N2I finishes on par with the other two alternatives, despite showing a sharper downward trend after the 80th query.

The chart also highlights the poor execution of the other classical methods tested, which lagged behind HardS for most queries. In contrast, the HardS measures exhibited highly variable performance, occupying a wide range in the central portion of the graph. At the top of this region, methods derived from neighborhood-based and likelihood-based HMs were predominant. Meanwhile, feature-based, tree-based, and class balance measures appeared more frequently at the lower end.

Regarding the overall method performance of NB, the curves in Figure 3 show a general trend of an initial rise in average ranks, followed by a prolonged stabilization phase. Up to about the 10th query, LSR stood out as the leading candidate. After that, its improvement rate slows, and it is quickly

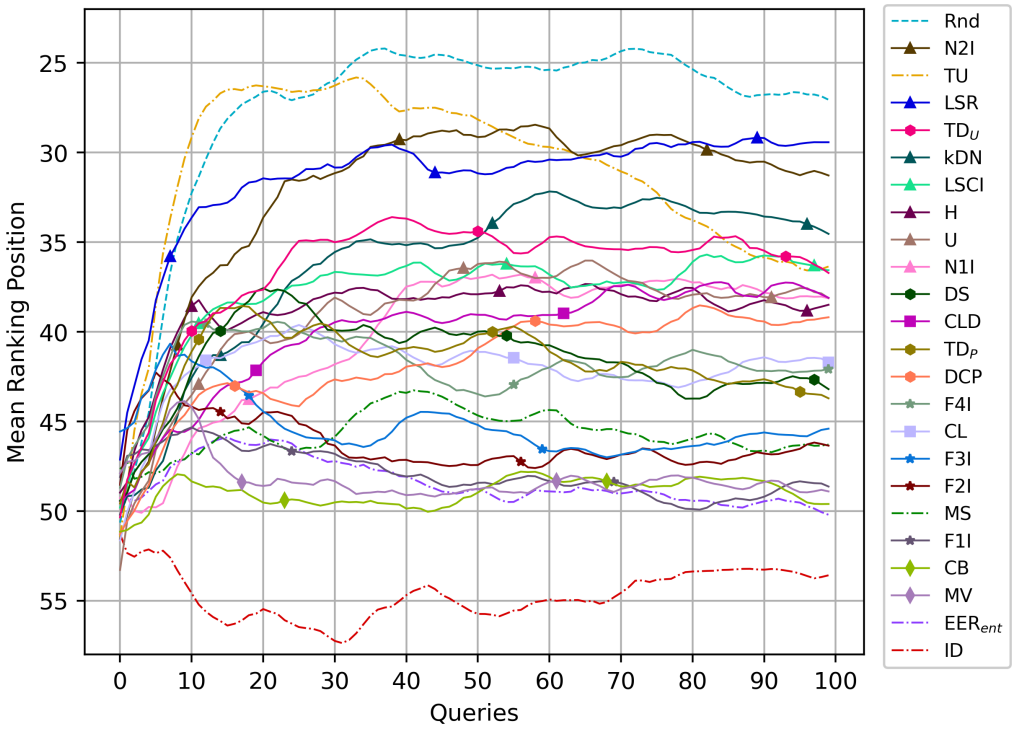

Figure 3: Ranking curves for methods tested with the NB algorithm. Curves with the same marker shape present HMs from the same group. Dash-dotted lines depict classical methods, and the cyan dashed line shows random sampling. The legend is sorted by the ALC scores.

surpassed by TU and Rnd, with TU taking the lead. However, around the 20th query, TU and Rnd enter a brief stabilization phase before their paths diverge. While Rnd's curve resumes an upward trend, taking the leading position, TU's curve defies the general trend and presents a declining behavior that persists through the end of the process. This decline allowed N2I and LSR, whose ranks had already been stabilized, to surpass TU near the 50th and 60th queries, respectively.

Similar to the CART scenario, there was an extensive range of HM-based measures in the central portion of the graph. In this case, however, two classical methods also occupied this region, though at its lower end. Most HardS methods surpassed MS, although it managed to consistently outperform a small group of them, which included only feature-based and class balance methods. On the other hand, EER$_{ent}$ is present at the bottom of the chart,

competing with the CB and MD methods. Moreover, ID exhibited the worst performance, trailing significantly behind the other curves.

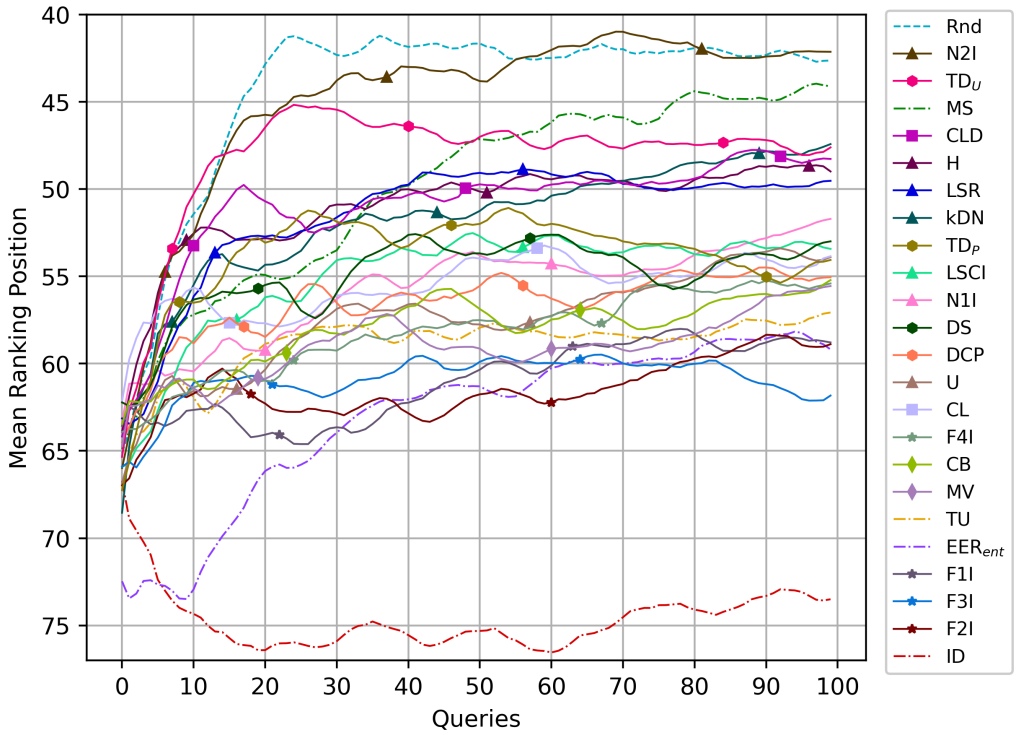

Figure 4: Ranking curves for methods tested with the 5NN algorithm. Curves with the same marker shape present HMs from the same group. Dash-dotted lines depict classical methods, and the cyan dashed line shows random sampling. The legend is sorted by the ALC scores.

Figure 4 presents the ranking curves for 5NN, where an upward trend is noticeable right at the beginning. This trend gradually diminishes across all curves, with some stabilizing at a certain point. In the first 10 queries, H, TD$_U$, and Rnd competed for the top position. Although TD$_U$ initially showed a slight advantage, it is rapidly surpassed by Rnd and loses second place to N2I. However, TD$_U$ maintained an upward trend until near the 25th query, where it experienced a slight decline and further stabilization. Simultaneously, N2I closes in on Rnd, overtaking it for the first time around the 50th query and maintaining a narrow lead until around the 80th query, where both methods reached similar ranks. Meanwhile, as TU$_U$ had stabi-

lized, MS continued to improve, eventually surpassing it near the 55th query. Nevertheless, TD_U achieved a higher ALC than MS by the end of the process.

The figure also shows that most methods within HardS outperformed TU. Some of these methods had even achieved better ranks than MS at the beginning but were surpassed by it around queries 30 and 45. Even so, they managed to maintain positions very close to TD_U Additionally, the poor performance of EER_{ent} and ID is evident at the bottom part of the graph.

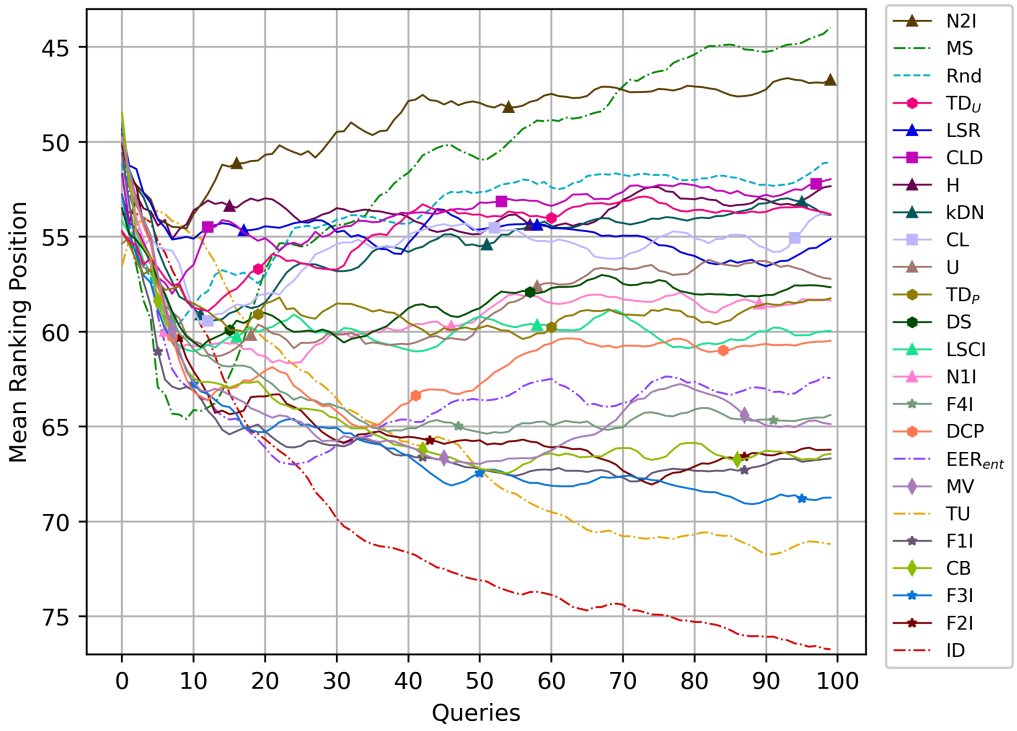

Figure 5: Ranking curves for methods tested with the SVM algorithm. Curves with the same marker shape present HMs from the same group. Dash-dotted lines depict classical methods, and the cyan dashed line shows random sampling. The legend is sorted by the ALC scores.

The behavior of the strategies concerning the SVM learner is depicted in Figure 5. All methods show an initial downward trend, likely influenced by the model's initial predictions generated by SVM. However, after this initial drop, the methods' behavior varied significantly. N2I, which already ranked higher in the early stages, exhibited the fastest growth and maintained the top positions throughout most of the graph, despite being overtaken by

MS at the final queries. Even with its poor early performance, MS had a good recovery, eventually surpassing the baseline and all HardS methods. Notwithstanding the good initial performance of CLD, LSR, and H, they remained in a similar ranking range and were eventually surpassed by the baseline. Rnd, which had a lower performance compared to figures 2 to 4, showed a consistent upward trend, frequently acting as an upper limit for the remaining methods evaluated.

Some proposed methods, exhibited similar behavior to Rnd, staying relatively close to its trajectory, while others remained closer to the middle portion of the graph. Additionally, at the lower end of the graph, some curves were unable to recover from the initial downward trend or showed only minimal recovery. For instance, EER_{ent}'s curve took time to regain upward momentum, ultimately failing to surpass other HardS methods beyond the feature-based and class-balance approaches. As for TU and ID, their ranks consistently dropped since the early stages, being sooner or later overtaken by the others.

6.3. HMs Groups and Learning Algorithms

The results presented so far suggest that more promising methods stem from HMs belonging to neighborhood-based, likelihood-based, or tree-based groups. In contrast, methods derived from HMs within the feature-based or class-balance groups tend to present lower average ranks. To tackle these aspects in more detail and to better asses whether the effectiveness of a certain group depends on the chosen learning algorithm, Figure 6 provides an alternative visualization of the previously presented ranking curves. In this visualization, each band represents the best and worst rankings achieved by methods derived from HMs within the same group for a given learner whereas alternatives outside HardS are represented by dotted lines.

Although the figure shows superior performance from methods derived from neighborhood-based HMs, the width of the bands assigned to this group is boosted by ranks achieved by N2I. Similar behavior occurred with tree-base and likelihood-based bands, where TD_U and CLD were responsible for drawing its upper limits. The figure also indicates greater overlap among the average rankings obtained by methods within the neighborhood-based, likelihood-based, and tree-based groups for the 5NN and SVM algorithms. For the CART algorithm, tree-based HMs did not perform well, even being surpassed by the feature-based group, which was among the worst-performing groups for all learners. In contrast, for NB, the neighborhood-based measures

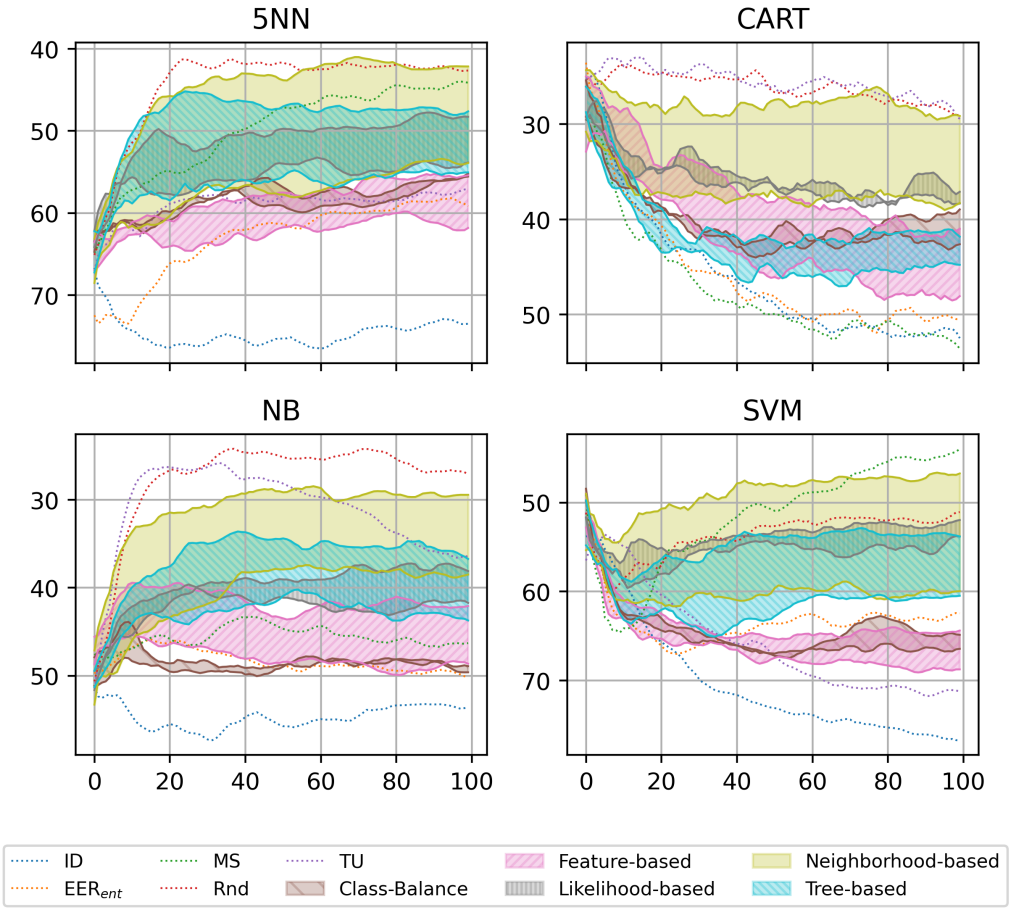

Figure 6: Comparison of average rankings across different HM groups for different learning algorithms. Each plot provides an alternative view of Figures 4, 2, 3, and 5, respectively. The bands represent the rankings curves obtained by each HardS method derived from an HM within a certain group. The upper and lower bounds indicate the best and worst average rankings achieved by these methods. Other tested methods as well as random sampling are represented by dotted lines.

prevailed, although there is some overlapping concerning some measures from the tree-based group.

Another notable observation is that the bands corresponding to the Likelihood-based methods tend to remain within the bound set by the tree-based ones for 5NN, NB, and SVM, although they are more centered within the band for 5NN, lower for NB, and higher for SVM. However, in the case of CART, the tree-based measures underperformed, while the Likelihood-based methods stayed within the limits established by the neighborhood-based band, demonstrating more robust behavior.

Despite outperforming some of the classical strategies tested, measures from the feature-based and class-balance groups showed the worst average ranks during the experiments. Nevertheless, it is important to highlight the performance of the F4I measure, which stood out compared to others within its group, being the primary contributor to the higher ranks assigned to the feature-based group in Figure 6, especially for CART.

6.4. Discussion

The presented results show that the HardS derived from measures within the neighborhood-based, tree-based, and likelihood-based groups achieved the highest rankings. This trend can be attributed to an increased effectiveness in identifying informative instances when certain data characteristics are considered. Moreover, methods derived from feature-based and class-balance groups could still surpass some of the classic methods used. While feature-based measures may have been affected by the high dimensionality of the experimental data, the performance of class balance measures was likely influenced by the arbitrary selection of instances from the minority class.

Some measures performed even better than their group's average. Methods associated with measures N2I, LSR, and kDN achieved some of the highest positions overall, while also demonstrating strong performance within the neighborhood-based group. Notably, N2I reached the top average rankings among all methods within HardS. Furthermore, the TD_U measure from the tree-based group and the CLD measure from the likelihood-based group also stood out for their consistently high average ranks. Finally, within the feature-based group, the F4 measure achieved performance levels comparable to those of the top-performing groups.

Additionally, we observed that the learning algorithm influenced the performance of certain HMs. For tree-based measures, a decline in performance was noted in conjunction with CART. A similar, though less pronounced,

effect was seen with likelihood-based measures when paired with NB. Both incidents are likely due to the similarities between the algorithms' operations and the measures in these groups. On the other hand, the neighborhood-based measure H performed significantly better paired with 5NN and SVM. Such improvement may be due to H's ability to select instances that are potentially decisive for the classification of the neighboring instances. Since both SVM and 5NN classify data based on the delimitation of regions in the input space, discovering the labels of these instances might have contributed to model performance.

During the experiments, Rnd often outperformed most other methods tested. One possible explanation for this performance could be the complexity of the classification problems used, as well as the number of queries made. Nevertheless, since Rnd is inherently exploratory — meaning it tends to cover a wide range of the data space without any specific bias — its early-stage proximity of certain methods within HardS, especially N2I, may suggest progress in controlling prospective and exploratory biases. In greater detail, by assuming Rnd's bias, we expect methods that effectively incorporate some exploratory bias to initially achieve similar ranks to it. Ineffective methods though, are prone to negatively impact initial model performance, as observed with TU when applied to SVM and 5NN. Furthermore, more prospective methods like MS and EER_{ent} tend to take longer to make proper choices, starting to achieve better ranks as they gather more information. Still, the same HardS methods also proved competitive with them during the later stages, which further supports our hypothesis. Nonetheless, additional research is necessary to fully understand the behavior and impact of these methods in different contexts.

7. Conclusion and Further Work

PAL is a specific scenario within AL where data is sampled from a pool of unlabeled instances to be labeled by an oracle, with the goal of creating efficient models using minimal annotated data to be used by ML algorithms. To achieve this, it focuses on instances that have a greater capacity to generalize the problem, making it crucial to establish a metric that distinguishes between more and less informative instances within the pool. Since information is a wide concept, only certain characteristics of an instance ought to be considered, as well as how to measure them. In contrast, HMs aim to identify why certain instances are harder to classify than others, rather than simply

marking an instance as hard to classify. Each group of HMs focuses on evaluating specific aspects that might increase the likelihood of misclassification, with different measures within each group employing distinct approaches to assess the same aspect.

This study proposes to investigate the use of HMs as information measures in the PAL context. Additionally, it aimed to analyze the individual behavior of these measures, asses their potential advantages over existing strategies, and explore their performance regarding the balance between prospection and exploitation biases. With this in mind, classical strategies for PAL and their corresponding methods were revisited, as well as the HMs found in the current literature. Based on this, a new query strategy called Hardness Sampling (HardS) was proposed. This strategy is based on the use of a fixed HM to employ the potential hardness of an instance as an informativeness measure for instances inside the pool. This value is then calculated through the use of the labels assigned by the learner to the unlabeled data. Subsequently, experiments were conducted using multiple datasets and four different learners to evaluate HardS. Furthermore, the strategy was also compared with methods from other classical strategies and random sampling to assess its performance.

The results indicate that the group to which the employed HM belongs directly affects HardS's performance. One possible explanation is that, in this context, a group represents the aspect that determines whether unlabeled instances are informative once labeled by the model. Thus, the aspect analyzed by neighborhood-based measures proved to be the most suitable. Additionally, tree-based and likelihood-based measures also demonstrated satisfactory performance, despite being more sensitive to the learning algorithm used. On the other hand, feature-based and class-balance measures showed inferior performance compared to other groups, indicating that the aspects analyzed by them may not be the most reliable for indicating the informativeness of an instance in PAL.

Furthermore, the specific HM used within a group proved to be another influential factor, since its goal is to summarize a certain aspect. Measures such as N2I, TD_U, CLD, and F4I presented themselves as the best in capturing such aspects within their groups. Additionally, some methods were able to select instances that ensured the best performance for the respective leaner, while in others, they remained close to the top performers. Nevertheless, all measures proved to be better or at least comparable alternatives to at least one of the classical methods discussed, indicating that HardS is a com-

petitive strategy. Additionally, the stable performance of the top-performing methods suggests that their use may indicate progress in regulating the balance between prospective and exploratory biases, although we do note that further studies are needed to confirm this hypothesis.

Further work might compare HardS with a broader range of query strategies for PAL, as well as evaluate it on a more diverse collection of datasets, ML algorithms, and additional evaluation metrics. This would provide a clear picture of the true impact of our approach and assist in confirming our hypothesis regarding its ability to balance prospection and exploration. Moreover, it would be valuable to conduct these experiments using other benchmarks from the community.

Acknowledgments

The first author would like to thank the Foundation for Support of Research of the Federal District (FAPDF) for the financial support provided for this research project (grant number: 857A).

CRediT authorship contribution statement

Gabriel da S. C. Nogueira: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft. **Davi P. dos Santos:** Data curation, Resources, Supervision, Validation, Writing - review & editing. **Luís P. F. Garcia:** Conceptualization, Methodology, Project administration, Resources, Supervision, Validation, Writing - review & editing.

References

[1] B. Settles, Active Learning, Synthesis Lectures on Artificial Intelligence and Machine Learning, Springer International Publishing, 2012. doi:10.1007/978-3-031-01560-1.

[2] B. Settles, Active Learning Literature Survey, Technical Report, University of Wisconsin-Madison Department of Computer Sciences, 2009. URL: https://minds.wisconsin.edu/handle/1793/60660.

[3] D. D. Lewis, W. A. Gale, A Sequential Algorithm for Training Text Classifiers, in: Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1994, pp. 3–12. doi:10.1007/978-1-4471-2099-5_1.

[4] D. Pereira-Santos, R. B. C. Prudêncio, A. C. P. L. F. de Carvalho, Empirical investigation of active learning strategies, Neurocomputing 326-327 (2019) 15–27. doi:10.1016/j.neucom.2017.05.105.

[5] M. R. Smith, T. Martinez, C. Giraud-Carrier, An instance level analysis of data complexity, Machine Learning 95 (2014) 225–256. doi:10.1007/s10994-013-5422-z.

[6] A. C. Lorena, P. Y. A. Paiva, R. B. C. Prudêncio, Trusting My Predictions: On the Value of Instance-Level Analysis, ACM Computing Surveys 56 (2024) 167:1–167:28. doi:10.1145/3615354.

[7] J. L. M. Arruda, R. B. C. Prudêncio, A. C. Lorena, Measuring Instance Hardness Using Data Complexity Measures, in: 9th Brazilian Conference on Intelligent Systems (BRACIS 2020), 2020, pp. 483–497. doi:10.1007/978-3-030-61380-8_33.

[8] M. G. Valeriano, P. Y. A. Paiva, C. R. V. Kiffer, A. C. Lorena, A framework for characterizing what makes an instance hard to classify, in: 12th Brazilian Conference on Intelligent Systems (BRACIS 2023), 2023, pp. 353–367. doi:10.1007/978-3-031-45389-2_24.

[9] C. Lancho, M. C. P. de Souto, A. C. Lorena, I. Martín de Diego, Complexity-driven sampling for bagging, in: Intelligent Data Engineering and Automated Learning (IDEAL 2023), 2023, pp. 15–21. doi:10.1007/978-3-031-48232-8_2.

[10] P. Y. A. Paiva, C. C. Moreno, K. Smith-Miles, M. G. Valeriano, A. C. Lorena, Relating instance hardness to classification performance in a dataset: a visual approach, Machine Learning 111 (2022) 3085–3123. doi:10.1007/s10994-022-06205-9.

[11] M. G. Valeriano, J. L. J. Pereira, C. R. V. Kiffer, A. C. Lorena, Explaining instances in the health domain based on the exploration of a

dataset's hardness embedding, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, (GECCO 2024), 2024, pp. 1598–1606. doi:10.1145/3638530.3664113.

[12] R. B. C. Prudêncio, A. C. Lorena, T. Silva-Filho, P. Drapal, M. G. Valeriano, Assessor models for explaining instance hardness in classification problems, in: 2024 International Joint Conference on Neural Networks (IJCNN), 2024, pp. 1–8. doi:10.1109/IJCNN60899.2024.10651521.

[13] X. Zhan, H. Liu, Q. Li, A. B. Chan, A comparative survey: Benchmarking for pool-based active learning, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, 2021, pp. 4679–4686. doi:10.24963/IJCAI.2021/634.

[14] D. D. Lewis, J. Catlett, Heterogeneous Uncertainty Sampling for Supervised Learning, in: Machine Learning Proceedings 1994, Morgan Kaufmann, San Francisco, California, USA, 1994, pp. 148–156. doi:10.1016/B978-1-55860-335-6.50026-X.

[15] N. Roy, A. McCallum, Toward optimal active learning through sampling estimation of error reduction, in: Proceedings of the Eighteenth International Conference on Machine Learning, 2001, p. 441–448.

[16] B. Settles, M. Craven, An analysis of active learning strategies for sequence labeling tasks, in: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, 2008, pp. 1070–1079.

[17] A. Fujii, K. Inui, T. Tokunaga, H. Tanaka, Selective sampling for example-based word sense disambiguation, Comput. Linguistics 24 (1998) 573–597.

[18] D. P. Dos Santos, A. C. de Carvalho, Selectively inhibiting learning bias for active sampling, in: 2015 Brazilian Conference on Intelligent Systems (BRACIS), 2015, pp. 62–67. doi:10.1109/BRACIS.2015.17.

[19] B. Settles, Automating Inquiry, in: Active Learning, Springer International Publishing, 2012, pp. 1–9. doi:10.1007/978-3-031-01560-1_1.

[20] B. Settles, Uncertainty Sampling, in: Active Learning, Springer International Publishing, 2012, pp. 11–20. doi:10.1007/978-3-031-01560-1_2.

[21] C. E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (1948) 379–423. doi:10.1002/j.1538-7305.1948.tb01338.x.

[22] J. Kremer, K. Steenstrup Pedersen, C. Igel, Active learning with support vector machines, WIREs Data Mining and Knowledge Discovery 4 (2014) 313–326. doi:10.1002/widm.1132.

[23] G. Schohn, D. Cohn, Less is more: Active learning with support vector machines, in: Proceedings of the Seventeenth International Conference on Machine Learning, 2000, p. 839–846.

[24] H. S. Seung, M. Opper, H. Sompolinsky, Query by committee, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT '92), 1992, p. 287–294. doi:10.1145/130385.130417.

[25] B. Settles, Exploiting Structure in Data, in: Active Learning, Springer International Publishing, 2012, pp. 47–54. doi:10.1007/978-3-031-01560-1_5.

[26] S. Dasgupta, D. Hsu, Hierarchical sampling for active learning, in: Proceedings of the 25th international conference on Machine learning (ICML '08), 2008, pp. 208–215. doi:10.1145/1390156.1390183.

[27] O. Sener, S. Savarese, Active learning for convolutional neural networks: A core-set approach, arXiv (2018). arXiv:arxiv:1708.00489.

[28] X. Li, Y. Guo, Adaptive Active Learning for Image Classification, in: 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 859–866. doi:10.1109/CVPR.2013.116.

[29] W.-N. Hsu, H.-T. Lin, Active Learning by Learning, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, volume 29, 2015, pp. 2659–2665. doi:10.1609/aaai.v29i1.9597.

[30] S. Ebert, M. Fritz, B. Schiele, RALF: A reinforced active learning formulation for object class recognition, in: 2012 IEEE Conference

on Computer Vision and Pattern Recognition, 2012, pp. 3626–3633. doi:10.1109/CVPR.2012.6248108.

[31] S.-J. Huang, R. Jin, Z.-H. Zhou, Active Learning by Querying Informative and Representative Examples, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2014) 1936–1949. doi:10.1109/TPAMI.2014.2307881.

[32] Z. Xu, K. Yu, V. Tresp, X. Xu, J. Wang, Representative Sampling for Text Classification Using Support Vector Machines, in: 25th European Conference on IR Research (ECIR 2003), 2003, pp. 393–407. doi:10.1007/3-540-36618-0_28.

[33] B. Settles, Minimizing Expected Error and Variance, in: Active Learning, Springer International Publishing, 2012, pp. 37–46. doi:10.1007/978-3-031-01560-1_4.

[34] D. A. Cohn, Z. Ghahramani, M. I. Jordan, Active Learning with Statistical Models, Journal of Artificial Intelligence Research 4 (1996) 129–145. doi:10.1613/jair.295.

[35] K. Konyushkova, S. Raphael, P. Fua, Learning active learning from data, in: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), 2017, pp. 4228–4238.

[36] I. Mas, R. Morros, V. Vilaplana, Picking Groups Instead of Samples: A Close Look at Static Pool-Based Meta-Active Learning, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019, pp. 1354–1362. doi:10.1109/ICCVW.2019.00171.

[37] S. Flesca, D. Mandaglio, F. Scala, A. Tagarelli, A meta-active learning approach exploiting instance importance, Expert Systems with Applications 247 (2024) 123320. doi:10.1016/j.eswa.2024.123320.

[38] P. Brazdil, J. N. van Rijn, C. Soares, J. Vanschoren, Metalearning: Applications to Automated Machine Learning and Data Mining, Springer International Publishing, 2022. doi:10.1007/978-3-030-67024-5_1.

[39] D. Zhu, Z. Li, X. Wang, B. Gong, T. Yang, A Robust Zero-Sum Game Framework for Pool-based Active Learning, in: Proceedings of the

Twenty-Second International Conference on Artificial Intelligence and Statistics, volume 89, 2019, pp. 517–526.

[40] V.-L. Nguyen, M. H. Shaker, E. Hüllermeier, How to measure uncertainty in uncertainty sampling for active learning, Machine Learning 111 (2022) 89–122. doi:10.1007/s10994-021-06003-9.

[41] X. Zhan, H. Liu, Q. Li, A. B. Chan, A Comparative Survey: Benchmarking for Pool-based Active Learning, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence(IJCAI-21), 2021, pp. 4679–4686. doi:10.24963/ijcai.2021/634.

[42] P.-Y. Lu, C.-L. Li, H.-T. Lin, Re-Benchmarking Pool-Based Active Learning for Binary Classification, arXiv (2023). doi:10.48550/ARXIV.2306.08954.

[43] A. C. Lorena, L. P. F. Garcia, J. Lehmann, M. C. P. Souto, T. K. Ho, How Complex Is Your Classification Problem? A Survey on Measuring Classification Complexity, ACM Computing Surveys 52 (2019) 107:1–107:34. doi:10.1145/3347711.

[44] H. Zhang, The Optimality of Naive Bayes., in: Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004), 2004, pp. 562–567.

[45] J. A. Sáez, M. Galar, J. Luengo, F. Herrera, Analyzing the presence of noise in multi-class problems: alleviating its influence with the One-vs-One decomposition, Knowledge and Information Systems 38 (2014) 179–206. doi:10.1007/s10115-012-0570-1.

[46] J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106. doi:10.1007/BF00116251.

[47] M. Kelly, R. Longjohn, K. Nottingham, The UCI Machine Learning Repository, 2023. URL: https://archive.ics.uci.edu, accessed: 2024-09-05.

[48] T. M. Mitchell, Machine learning, International Edition, McGraw-Hill Series in Computer Science, McGraw-Hill, 1997.

[49] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (1995) 273–297. doi:10.1007/BF00994018.

[50] L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, Classification and Regression Trees, Taylor & Francis, 1984.

[51] J. Cohen, A Coefficient of Agreement for Nominal Scales, Educational and Psychological Measurement 20 (1960) 37–46. doi:10.1177/001316446002000104.

[52] R. Artstein, M. Poesio, Inter-Coder Agreement for Computational Linguistics, Computational Linguistics 34 (2008) 555–596. doi:10.1162/coli.07-034-R2.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[54] T. Danka, P. Horvath, modAL: A modular active learning framework for Python, arXiv (2018). doi:10.48550/arXiv.1805.00979.