Uncertainty-Aware Classifier Accuracy Prediction under Prior Probability Shift

Lorenzo Volpi, Alejandro Moreo, and Fabrizio Sebastiani

Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
{firstname.lastname}@isti.cnr.it

Abstract. *Classifier accuracy prediction* (CAP) is the task of estimating the accuracy that a trained classifier will have on a set of datapoints characterized by *dataset shift*, i.e., data sampled from a distribution different from the distribution from where the training data originated. We here propose three new algorithms for CAP under *prior probability shift*, an important type of dataset shift; our first two algorithms are built on top of surrogate quantifiers, while the third is a Bayesian algorithm. Importantly, these methods do not merely return a point estimate (i.e., the estimated accuracy of the classifier on the unlabelled data), but also return a measure of their uncertainty in this estimate. Systematic experiments on a number of datasets and against a number of baseline CAP algorithms show that our methods are competitive with strong baselines, with one of them frequently achieving the best balance between point estimation accuracy and interval quality.

Keywords: Classifier accuracy prediction, Dataset shift, Prior probability shift, Label shift, Quantification, Point estimates, Uncertainty quantification, Confidence intervals

1 Introduction

The accuracy that a classifier will obtain when applied to new unlabelled data is usually estimated by cross-validation. However, the assumption that this procedure returns *reliable* estimates rests on the assumption that the labelled test data used in the cross-validation process are representative of the target unlabelled data. In other words, cross-validation relies on the *IID assumption*, which states that the joint distribution $P(X, Y)$ from which the labelled data are sampled and the joint distribution $Q(X, Y)$ from which the unlabelled data are sampled, are the same. When $P(X, Y) \neq Q(X, Y)$ we say we are instead in the presence of *dataset shift* [21, 25], and *classifier accuracy prediction* (CAP, also known as *unsupervised accuracy estimation* [5, 18], or *out-of-distribution error*

prediction [31]) can no more be carried out by cross-validation, thus becoming an open problem.[1]

Until a few years ago, little research had been conducted on CAP under dataset shift, but in recent years the scientific community has investigated the problem more vigorously, and to deliver algorithms that address it [9, 11, 17, 18, 28, 29]. Most solutions proposed so far (and the present paper is no exception) address *prior probability shift* (PPS, a.k.a. *label shift*), a type of dataset shift that has sometimes been called *the paradigmatic* case of dataset shift [33].

Most CAP works published so far return, given a classifier h and a set σ of unlabelled datapoints, a *point estimate* of the accuracy that h is going to have on σ. However, since the returned estimate is characterized by uncertainty, it would be desirable to *quantify* this uncertainty [14]; measuring the uncertainty of the produced estimate is one of the goals of the present paper.

The contributions of this paper are threefold. First, we propose two new methods for CAP under PPS, dubbed *SQS-H* and *SQS-S*; these methods use a *quantifier* (i.e., a model trained to estimate the prevalence values of the classes in a set of unlabelled datapoints; see [7]) as an auxiliary model for selecting validation sets that resemble the target distribution. Second, we propose *BayCAP*, a Bayesian CAP method that samples sets from a posterior distribution over target contingency tables and turns these sets into posterior sets of classifier accuracy values. Third, we analyse the ability of these methods to measure the uncertainty of the returned classifier accuracy estimate.

Our paper is organized as follows. Section 2 summarizes related work in CAP under dataset shift, and in uncertainty quantification for CAP. In Section 3 we set the notation we use, and review our underlying distributional assumptions and the reasons why cross-validated estimation of accuracy is unreliable under prior probability shift. In Section 4, we describe two new CAP methods that rely on surrogate quantifiers, while in Section 5 we propose a fully Bayesian approach to CAP. Section 6 describes the experiments we have carried out, while Section 7 wraps up, pointing at avenues for further research.

2 Related work

Research on CAP under dataset shift has grown substantially in recent years. Earlier proposals include methods based on confidence scores, domain-shift diagnostics, nearest-neighbour structure, or distribution matching [2–6, 9, 12, 15, 20, 26, 31, 32]. We focus here on the methods that are closest to our setting, i.e., CAP under PPS, and on CAP methods that either estimate contingency tables or quantify uncertainty.

The *Difference of Confidence* (DoC) method [11] estimates the accuracy of a classifier from changes in the confidence scores it returns. DoC constructs shifted validation sets, measures how the average confidence changes with respect to the

[1] Throughout this paper, by CAP we will indicate a task where no access to labels from distribution $Q(X, Y)$ is assumed. Other works (e.g., [30]) assume instead that a limited number of labels from $Q(X, Y)$ are available to help in the prediction.

original validation set, and trains a regressor to map these confidence differences to differences in accuracy.

More recent work has moved from directly estimating the value of a single accuracy measure, to estimating the values of the entries of the contingency table produced by the classifier. This is useful because several evaluation measures can then be computed from the same estimated table entry values. QuAcc [29] treats the entries of the contingency table as classes and estimates their prevalence by means of a PPS-robust quantifier. LEAP [28] instead solves a system of linear equations that the contingency table entry values must satisfy under PPS, also using a PPS-robust quantifier for estimating class prevalence values required by the equations.

Confidence-Based Performance Estimation (CBPE – [17]) also targets the contingency table, but models its entries as random variables whose values derive from the classifier's confidence scores. This gives CBPE a natural mechanism for measuring the uncertainty of the prediction. However, its guarantees rely on the assumption that the classifier is calibrated on the target distribution. CBPE is therefore the closest existing work to the uncertainty-aware perspective adopted in the present paper, since it models uncertainty over quantities from which evaluation measures can be derived. Nevertheless, its uncertainty model is frequentist rather than Bayesian, and thus does not provide a natural mechanism for incorporating prior knowledge about the target contingency table or the desired evaluation measures. To the best of our knowledge, no Bayesian treatment of uncertainty in CAP has been proposed before the present paper.

3 Preliminaries

3.1 Background and notation

In this section we introduce basic concepts and notation on classifier accuracy prediction and uncertainty quantification.

Let $\mathcal{X} = \mathbb{R}^d$ denote the input space (a space of vectors of covariates), and let $\mathcal{Y} = \{\omega_1, \ldots, \omega_K\}$ denote the output space (a space of classes). We assume we are in the presence of prior probability shift (PPS), i.e., that the training and test data are drawn from distributions P and Q, respectively, such that $P(Y) \neq Q(Y)$ and $P(X|Y) = Q(X|Y)$.

In this paper we assume by default that our classifiers are probabilistic, i.e., that they generate a vector of class specific posterior probability estimates (one for each class). This is not too restrictive, since most classifiers generate a vector of class-specific scores that can be converted into posterior probabilities estimates via calibration. In this paper, though, we do not make any hard assumption on how well-calibrated these classifiers are.

Let $h : \mathcal{X} \to \Delta^{K-1}$ denote a probabilistic classifier mapping input $x \in \mathcal{X}$ into a vector $\hat{p} = h(x) = (\hat{p}_1, \ldots, \hat{p}_K)$ of posterior probability estimates, where $\hat{p}_k =$

$\hat{P}(Y = \omega_k|X = x)$, and where Δ^{K-1} denotes the standard probability simplex

$$\Delta^{K-1} = \{p \in \mathbb{R}^K : p_k \geq 0, \sum_{k=1}^{K} p_k = 1\}$$

and where one can obtain a crisp prediction by simply applying the Maximum a Posteriori (MAP) rule $\hat{y} = \arg\max_{\omega_k \in \mathcal{Y}} \hat{P}(\omega_k|x)$.

A key property, which will be instrumental in the following sections, is that for any deterministic and measurable function f on $\mathcal{X}$, the stationarity $P(X|Y) = Q(X|Y)$ implies that $P(f(X)|Y) = Q(f(X)|Y)$; see also [19, Lemma 1] and [27]. As a direct consequence, if we take f to be a classifier h, this property implies the stationarity of the (crisp and soft) classifier predictions, i.e., it implies that $P(\hat{Y}|Y) = Q(\hat{Y}|Y)$ and $P(Z|Y) = Q(Z|Y)$, with $Z = h(X)$. We write C to denote the matrix of class-conditional classification rates, with $C_{ij} = P(\hat{Y} = \omega_i|Y = \omega_j)$. We denote by $\pi^P = (\pi_1^P, \ldots, \pi_K^P) \in \Delta^{K-1}$ the class prevalence values of the source distribution, with $\pi_k^P = P(Y = \omega_k)$; analogously, we write $\pi^Q = (\pi_1^Q, \ldots, \pi_K^Q) \in \Delta^{K-1}$ for the target distribution, with $\pi_k^Q = Q(Y = \omega_k)$.

We assume that classifier h (i.e., the classifier whose accuracy we aim to estimate) has already been trained on a set L of labelled datapoints which may no longer be available. We assume access to a validation set V of labelled datapoints (which will allow us to train our accuracy predictors), with L and V drawn from the same distribution P. Given a measure of accuracy A, CAP consists of predicting the accuracy of h on an unlabelled test set U drawn from a distribution Q related to P via PPS. We write $A(h, U)$ to denote the empirical value of measure A on set U. We restrict our attention to evaluation measures A that can be computed from a contingency table T, and write $A(T)$ when the measure is applied directly to such a table.

Note that our goal is to estimate the performance of an already trained classifier h, and not to improve its performance on the target distribution Q. In this paper, we are interested not only in the accuracy of classifier accuracy point estimates, but also in characterizing the uncertainty associated with such estimates.

3.2 Why Can't We Trust Cross-Validation Estimates Under Prior Probability Shift?

Assume our classifier accuracy measure is ("vanilla") accuracy (in symbols: Acc), i.e., the fraction of classification decisions that are correct, and assume we have an arbitrarily good, unbiased estimate $\widehat{\text{Acc}}$ of the classifier's accuracy obtained by means of cross-validation on training data L. Assume our labelled and unlabelled data are drawn from distributions P and Q, respectively, that are related via PPS: can we trust our $\widehat{\text{Acc}}$ estimate? This amounts to asking whether our estimator is unbiased under PPS, i.e., whether

$$\text{Bias}(\widehat{\text{Acc}}) \equiv \mathbb{E}[\widehat{\text{Acc}}] - Q(\hat{Y} = Y) = 0 \tag{1}$$

where $\mathbb{E}$ indicates expectation (taken over all possible sets L of labelled datapoints) and $\hat{Y}$ is a random variable ranging on the predicted class labels. Since our training data is drawn from distribution P and our estimate is unbiased, asymptotically it holds that $\mathbb{E}[\widehat{\text{Acc}}] = P(\hat{Y} = Y)$. Note that

$$P(\hat{Y} = Y) = \sum_{\omega_k \in \mathcal{Y}} P(\hat{Y} = \omega_k, Y = \omega_k) = \sum_{\omega_k \in \mathcal{Y}} P(\hat{Y} = \omega_k | Y = \omega_k) P(Y = \omega_k)$$
$$Q(\hat{Y} = Y) = \sum_{\omega_k \in \mathcal{Y}} Q(\hat{Y} = \omega_k, Y = \omega_k) = \sum_{\omega_k \in \mathcal{Y}} Q(\hat{Y} = \omega_k | Y = \omega_k) Q(Y = \omega_k)$$

The stationarity of the class-conditional classifier predictions under PPS (Section 3.1) implies that the class-conditional error rates are stationary as well. Let C denote the matrix with entries $C_{ij} = P(\hat{Y} = \omega_i | Y = \omega_j)$, and let π^P and π^Q be the training and test class prevalence values, respectively. We can now rewrite our above equations as

$$P(\hat{Y} = Y) = \text{diag}(C)^\top \pi^P \qquad Q(\hat{Y} = Y) = \text{diag}(C)^\top \pi^Q$$

where $\text{diag}(M) \in \mathbb{R}^n$ denotes the vector of diagonal entries of $M \in \mathbb{R}^{n \times n}$. Letting $d = \text{diag}(C)$, it follows that

$$\text{Bias}(\widehat{\text{Acc}}) = d^\top \pi^P - d^\top \pi^Q = d^\top (\pi^P - \pi^Q) \tag{2}$$

PPS means that $\pi^P \neq \pi^Q$, which thus implies that $\text{Bias}(\widehat{\text{Acc}}) = 0$ holds only if $d = \mathbf{0}$, or if $d \perp (\pi^P - \pi^Q)$.

The first condition ($d = \mathbf{0}$) corresponds to a degenerate case in which the classifier has zero probability of making a correct prediction, and can thus be safely excluded. The second condition ($d \perp (\pi^P - \pi^Q)$) deserves further attention, though. Note that, as both π^P and π^Q lie on the probability simplex Δ^{K-1}, if this condition is verified the vector $(\pi^P - \pi^Q)$ lies in the linear subspace parallel to the simplex, namely $\{x \in [-1, 1]^K : \sum_{i=1}^K x_i = 0\}$. Since d contains the true positive rates of all classes, all its components are non-negative, and orthogonality between d and $(\pi^P - \pi^Q)$ is thus highly restrictive and unlikely to arise in realistic settings.

For example, one case in which $\text{Bias}(\widehat{\text{Acc}}) = 0$ irrespective of π^P and π^Q is when the true positive rates for all the classes are equal, i.e., when $d = \beta \mathbf{1}$ for some $\beta \in \mathbb{R}$. Unless the classifier is a perfect one, such a configuration becomes highly unlikely. However, since this condition does not depend on the class prevalence values (and more importantly, on the unknown π^Q), we can check its validity by inspecting C, i.e., using only training data. If the condition is approximately satisfied, we may safely use our cross-validated estimate of accuracy, even under suspected PPS.

In sum, the accuracy estimates obtained by cross-validation are generally unreliable under PPS.

4 Classifier accuracy prediction via surrogate quantifiers

The methods we propose in this section rely on the same underlying rationale. Since, as a result of Equation 2, accuracy estimates obtained via cross-validation are biased because of the prevalence shift ($\pi^P - \pi^Q$), our goal is to estimate classifier performance using new sets drawn from the validation data, whose class prevalence values resemble those of the target test distribution. In order to achieve this, we explore new CAP methods that rely on quantification as a surrogate model, i.e., on methods specialised in estimating the class prevalence values π^U of sets U of unlabelled datapoints drawn from the target distribution Q.

A quantifier can be formalized as a function $q : 2^{\mathcal{X}} \to \Delta^{K-1}$, mapping a set σ of instances from $\mathcal{X}$ to a vector of class prevalence estimates $\hat{\pi}^Q = q(\sigma) = (\hat{\pi}^Q_1, \ldots, \hat{\pi}^Q_K)$, where $\hat{\pi}^Q_k = \hat{Q}(Y = \omega_k)$ denotes the estimated prevalence of class ω_k in σ. The error we incur in estimating the true prevalence π^Q as $\hat{\pi}^Q$ is typically measured in terms of a divergence function $\mathcal{D}$.

As discussed in Section 3.1, under PPS it holds that $P(f(X)|Y) = Q(f(X)|Y)$, where function f acts as a *representation function*, typically instantiated as a probabilistic classifier. This allows quantifiers to model distributions over posterior probabilities rather than to directly model distributions of covariates, which is infeasible in high-dimensional spaces. This property underlies the broad class of *aggregative quantifiers* [§4.2 in 7], which includes those adopted in this paper.

In the methods we propose, we *do not* use the target classifier h as the classifier on which our aggregative quantifiers rely. Since the quantifier is used as an auxiliary instrument to estimate the accuracy of h, using h also as the surrogate classifier would make the measurement depend on the very classifier being measured, thus establishing a vicious circle. We thus keep the surrogate classifier used for quantification distinct from the classifier whose accuracy is to be predicted. As a consequence, the surrogate classifier must be trained from the validation data used to train the CAP method. Since in-sample posterior probability estimates would tend to be overconfident, we instead obtain out-of-sample estimates by applying cross-validation within the validation set.

In what follows, we present two new uncertainty-aware CAP methods that use quantification as a surrogate task.

4.1 SQS-H: Surrogate quantifier-based sampling with hard prevalence matching

Our first proposed CAP method based on quantification is *Surrogate Quantifier-based Sampling with Hard prevalence matching* (SQS-H). This method consists of these steps:

1. Train a quantifier q robust to PPS using V as the training data;
2. Compute the class prevalence value estimates $\hat{\pi}^Q = q(U)$ for the unlabelled set U;

3. By drawing (with replacement) labelled datapoints uniformly at random from validation set V, generate m sets of datapoints $\{\sigma_i\}_{i=1}^m$ with class prevalence values $\pi^{(i)}$ (approximately) equal to the estimated class distribution $\hat{\pi}^Q$;[2]
4. Compute the accuracy of h on each of the m sets, as $\{A(h, \sigma_i)\}_{i=1}^m$;
5. Use these accuracy values to compute $100(1-\alpha)\%$ empirical percentile intervals, and return the empirical mean as the point estimate, where α is the significance level.

The rationale of this algorithm is the following. Step 3 induces a new distribution P' over the selected validation sets[3], with $P'(Y) \neq P(Y)$. However, Step 3 does not change the class-conditional distributions, since the selected sets are still made of instances drawn from the same conditional distributions as the original validation data, i.e., $P'(X|Y) = P(X|Y)$. Under PPS between validation and target data, it holds that $P(X|Y) = Q(X|Y)$, and thus it holds that $P'(X|Y) = Q(X|Y)$ as well. Since the selection is designed to make the class prior of the selected validation sets close to that of the target set, we also have $P'(Y) \approx Q(Y)$. If PPS is the only source of shift, it follows that $P'(X, Y) \approx Q(X, Y)$ (since $P'(X, Y)$ is factorized as $P'(X|Y)P'(Y)$ and since both factors are similar to the counterparts in Q), so the accuracy of h estimated on sets from P' is a good approximation of its accuracy on the target distribution Q.

4.2 SQS-S: Surrogate quantifier-based sampling with soft prevalence matching

A limitation of SQS-H is that, while we can observe the true prevalence values $\pi^{(i)}$ of our validation sets, we cannot observe the true prevalence π^Q of U, since we can observe only an estimate $\hat{\pi}^Q$, which is affected by quantification error.

The second quantification-based CAP method we propose is called *SQS-S* (standing for *Surrogate Quantifier-based Sampling with Soft prevalence matching*), and is aimed at mitigating this limitation. The key idea behind SQS-S is to inject the same type of estimation error before the selection of validation sets. The rationale is that, whenever the quantifier q returns similar *estimated* class prevalence values $\hat{\pi}^{(i)}$ and $\hat{\pi}^Q$ for sets σ_i and U, respectively, their true class prevalence values $\pi^{(i)}$ and π^Q are also likely to be similar.

A fundamental difference between SQS-S and SQS-H is that the quantifier q cannot be trained on the same validation instances from which sets are later drawn, since this would inevitably lead to overconfident prevalence estimates. Following this rationale, we reserve part of the validation data for quantifier training. A second important difference is that the validation sets employed to

[2] The fact that we draw with replacement implies that the σ_i's are *bags*, and not sets. However, this is inessential, since no known quantification method depends on the fact that the datapoints in σ_i are all distinct, which means that quantifiers can indifferently be applied to sets or bags.

[3] Note that P and P' are affected by a different type of dataset shift, namely *sampling selection bias*.

estimate accuracy cannot be selected directly, as in SQS-H, since the selection criterion now relies on similarity between *predicted* prevalence values. To address this issue, we first draw n initial sets whose prevalence values approximately cover the probability simplex uniformly, and then retain the fraction γ whose predicted prevalence values are closest to $\hat{\pi}^Q$. Note that this algorithm introduces two hyperparameters: the fraction $\gamma \in [0, 1]$ and the criterion of dissimilarity (divergence) $\mathcal{D}$.

SQS-S consists of the following steps:

1. Randomly partition the validation set V into V_1 and V_2 via stratification, with $|V_1| \approx |V_2|$;
2. Train a quantifier q robust to PPS using V_1;
3. Estimate the class prevalence of the unlabelled set $\hat{\pi}^Q = q(U)$;
4. Extract n random sets σ_i by drawing with replacement from V_2, with each set characterised by different class prevalence values uniformly covering the probability simplex;
5. Estimate class prevalence values in $\hat{\pi}^{(i)} = q(\sigma_i)$ for all sets σ_i;
6. Take the $m = \lceil \gamma\, n \rceil$ sets σ_i whose predicted prevalence is closest to $\hat{\pi}$ in terms of a divergence function $\mathcal{D}$. We here adopt the L1 distance, i.e., given $r_i = ||\hat{\pi}^Q - \hat{\pi}^{(i)}||_1$ and $r_{(m)}$ the m-th closest such distance, we keep $I_\gamma = \{i : r_i \leq r_{(m)}\}$;
7. Compute the accuracy of h $\{A(h, \sigma_j)\}_{j \in I_\gamma}$ in each of the selected sets;
8. Use these accuracy values to compute $100(1 - \alpha)\%$ empirical percentile intervals, and return the empirical mean as the point estimate, where α is the significance level.

5 Bayesian posterior sampling of accuracy

Bayesian methods provide a natural framework for uncertainty-aware classifier accuracy prediction. In contrast to point-estimation approaches, they return a posterior distribution over the quantities of interest and allow prior information to be incorporated explicitly. In our setting, the quantity of interest is the performance of a fixed classifier h on an unlabelled target set U.

Rather than specifying a probabilistic model directly for a particular evaluation measure, we model the posterior distribution of the target contingency table. This choice makes the method metric-agnostic: once posterior sets of the contingency table are available, posterior sets of any contingency-table-based evaluation measure can be obtained by applying the corresponding functional to each sampled table. As from Section 3.1, we use C to denote the classifier prediction rates with (estimated) entries $C_{ij} = P(\hat{Y} = \omega_i \mid Y = \omega_j)$, where $\hat{Y} = h(X)$; this matrix is shared by distributions P and Q under PPS.

We adopt the BayesianCC model of [33], originally proposed as a quantification method, which performs Bayesian inference over the source prevalences, target prevalences, and the conditional prediction matrix. In particular, the posterior is given by

$$p(\pi^P, \pi^Q, C \mid V, U) \propto p(V, U \mid \pi^P, \pi^Q, C)\, p(\pi^P, \pi^Q, C)$$

Following [33], Dirichlet priors can be placed on the simplex-valued parameters π^P, π^Q, and on the columns of C. The model assumes the sampling process

$$\begin{aligned}(n^s_{Y=\omega_1}, \ldots, n^s_{Y=\omega_K}) \mid \pi^P &\sim \text{Multinomial}(N^P, \pi^P) \\ (f_{1j}, \ldots, f_{Kj}) \mid n^s_{Y=\omega_j}, C &\sim \text{Multinomial}(n^s_{Y=\omega_j}, C_{:j}) \quad j = 1, \ldots, K \\ (n^t_{\hat{Y}=\omega_1}, \ldots, n^t_{\hat{Y}=\omega_K}) \mid \pi^Q, C &\sim \text{Multinomial}(N^Q, C\pi^Q)\end{aligned}$$

where $N^P = |V|$ and $N^Q = |U|$. Here, $n^s_{Y=\omega_j}$ denotes the number of validation instances whose true class is ω_j; f_{ij} denotes the number of validation instances with predicted class ω_i and true class ω_j; and $n^t_{\hat{Y}=\omega_i}$ denotes the number of target instances predicted as class ω_i.

After drawing m posterior sets $\{(\pi^Q_{(r)}, C_{(r)})\}_{r=1}^m \sim p(\pi^Q, C \mid V, U)$, we construct, for each draw, the corresponding normalized target contingency table $T_{(r)} = C_{(r)} \operatorname{diag}(\pi^Q_{(r)})$, where entry (i, j) in $T_{(r)}$ represents one posterior draw of the joint probability $Q(\hat{Y} = \omega_i, Y = \omega_j)$. Given any evaluation measure A expressible as a function of a contingency table, we obtain posterior samples of the corresponding target performance as $\{A(T_{(r)})\}_{r=1}^m$. Thus, the posterior distribution over the evaluation measure is the *pushforward* of the posterior distribution over target contingency tables through the metric functional A.

We use the resulting empirical posterior distribution to compute $100(1-\alpha)\%$ posterior credible intervals and report its mean as a point estimate.

6 Experiments

In this section, we evaluate the proposed uncertainty-aware CAP methods on binary and multiclass tabular datasets under PPS, considering both point-estimation accuracy and interval quality.

Datasets: We evaluate the CAP methods considered in this paper on tabular classification datasets from the UCI Machine Learning Repository [16]. We restrict our attention to those accessible through QuaPy [22], and select the 5 largest binary and 10 largest multiclass datasets. The number of instances ranges from 830 (mammographic) to 1,024,985 (poker_hand); the number of features ranges from 5 (mammographic) to 617 (isolet); the number of classes ranges from 2 (all binary datasets) to 26 (isolet and letter).

Classifier-Training Algorithms: In order to generate the classifiers whose accuracy we aim to predict, we consider five standard learning algorithms: Logistic Regression (LR), k-Nearest Neighbours (kNN), Support Vector Machines (SVM), MultiLayer Perceptrons (MLP), and Random Forests (RF). All classifiers are trained using the implementations provided by scikit-learn [23]. We leave all hyperparameters at their default values, except for SVM, for which we use the RBF kernel.

Baselines: As baselines against which we test the performance of our methods, we use DoC [11], QuAcc [29], LEAP [28], and L-CBPE [17].

For DoC, we use our own implementation, since no public implementation was released by the authors. For QuAcc, we use the QuAcc-$K\times K$ variant, which estimates the entries of the contingency table by reducing the problem to K quantification tasks over the K classes. For LEAP, we use O-LEAP, the overconstrained variant that takes into account all the equations governing the contingency-table entries. For QuAcc and LEAP, we use the original implementations available online.[4] Since CBPE [17] assumes posterior probabilities to be calibrated for the target distribution, we first recalibrate them with LasCal [24], a calibration method designed for PPS, and refer to the resulting pipeline as L-CBPE. We use the original implementations released by the respective authors; since the available CBPE implementation only supports binary problems, we evaluate this baseline only on binary datasets.

We exclude several previously proposed CAP methods from our comparison; some [3, 15] rely on assumptions that are too restrictive for our setting, while others [4, 9, 20, 31] were not competitive in preliminary experiments. We nevertheless include L-CBPE, despite its lower competitiveness, because CBPE is, to the best of our knowledge, the only existing uncertainty-aware CAP method.

The code that reproduces all our experiments is available on GitHub.[5]

Evaluation Measures: We evaluate the competing CAP methods along two dimensions. First, we assess the quality of their point estimates; second, we assess the quality of their interval estimates.

For evaluating point estimates, we use absolute error (AE). For a generic classifier accuracy measure A, AE is defined as $|A(h,U_i) - \widehat{A}(h,U_i)|$, where $A(h,U_i)$ is the true accuracy of classifier h on the i-th test set U_i, and $\widehat{A}(h,U_i)$ is the corresponding value predicted by the CAP method. In our experiments, the target classifier accuracy measure A is vanilla accuracy (Acc), which is one of the most widely used measures of classifier effectiveness.

For uncertainty evaluation, we rely on coverage and amplitude of the interval estimates generated by each method. Let $[\ell_i, u_i]$ be the interval estimate generated by a given method at $\alpha = 0.05$ significance level for the i-th test set, and let $a_i = A(h,U_i)$ be the corresponding true accuracy. Coverage ($\mathcal{C}$) is defined as the fraction of the N test sets for which the true accuracy falls within the predicted interval, i.e., $\mathcal{C} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\ell_i \leq a_i \leq u_i]$. Amplitude ($\mathcal{A}$) is defined as the average width of the predicted intervals, i.e., $\mathcal{A} = \frac{1}{N}\sum_{i=1}^{N}(u_i - \ell_i)$. A good interval estimate should thus exhibit high coverage and low amplitude. Since these two objectives are conflicting, we also report the mean Winkler score ($\mathcal{W}$), which combines them into a single measure. For a given interval $[\ell_i, u_i]$, the Winkler score [10] is defined as

$$\mathcal{W}_i = (u_i - \ell_i) + \frac{2}{\alpha}\left(\max\{0, \ell_i - a_i\} + \max\{0, a_i - u_i\}\right)$$

[4] `https://github.com/lorenzovolpi/cap-ml`.

[5] `https://github.com/lorenzovolpi/Uncertainty-Aware-CAP`.

Lower values of AE, amplitude, and Winkler score indicate better performance, while higher values of coverage are preferable. Throughout all experiments, we use nominal level $(1 - \alpha) = 0.95$.

Evaluation Protocol: The experimental protocol we adopt is as follows. Given a labelled collection D of pairs (x_i, y_i), we split it into a training set (70%) and a test set U (30%) via stratified sampling. We further split, with stratification, the training set into a set L (50%), used for training the classifier h, and a validation set V (50%), used for training the CAP methods. These partitions are consistent across all methods.

We then draw, uniformly at random, 100 prevalence vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{100}$ from the unit simplex Δ^{K-1}. For each prevalence vector $\mathbf{v}_i$, we randomly extract, with replacement, from U a test set U_i of size $|U_i| = 1000$ whose class prevalence values match $\mathbf{v}_i$. This protocol, known as the Artificial Prevalence Protocol (APP), is the standard protocol for simulating PPS in the literature [7, 8].

Since our evaluation is uncertainty-aware, all methods are required to return both a point estimate and an interval estimate for each test set. SQS-H, SQS-S, BayCAP, and L-CBPE provide such intervals natively: SQS-H and SQS-S return empirical percentile intervals, BayCAP returns posterior credible intervals, and L-CBPE derives confidence intervals from the Poisson-binomial distribution induced by the calibrated confidence scores. For methods that only return point estimates (namely LEAP, QuAcc, and DoC), we obtain confidence intervals by bootstrap resampling the test set and taking the empirical $\frac{\alpha}{2}$ and $(1 - \frac{\alpha}{2})$ quantiles of the resulting accuracy estimates; we return the mean of such samples as the point estimate.

In all experiments, we let each method generate 1,000 estimates. For surrogate-quantifier-based methods, we draw validation sets σ_j such that $|\sigma_j| = |U_i|$. For SQS-H, we generate $m = 1,000$ such sets. For SQS-S, we first draw $n = 10,000$ candidate sets and then retain the closest $m = \lceil \gamma\, n \rceil$ sets, with $\gamma = 0.1$, thus obtaining $m = 1,000$ estimates. Our implementation of BayCAP builds on the BayesianCC implementation of [33], available in QuaPy. This implementation uses NUTS [13] as its sampler; we run 500 warm-up iterations before collecting posterior samples. In all experiments, we use uniform Dirichlet priors, i.e., Dirichlet distributions with all concentration parameters equal to one. As the surrogate quantifier in our methods, we adopt PACC [1] as implemented on the QuaPy[6] package [22], since PACC is an efficient algorithm that requires no hyperparameter tuning.

6.1 Results

Table 1 summarizes the experimental results in terms of average ranks. For each classifier and dataset, methods are ranked according to their mean AE and Winkler score ($\mathcal{W}$) across the APP test sets; the table then reports the average rank across the 15 datasets, separately for each classifier. Since raw AE and

[6] `https://github.com/HLT-ISTI/QuaPy`

Winkler scores are not directly comparable across datasets, average ranks provide a more compact comparison of point-estimation performance and prediction-interval quality; lower ranks are better. Results for L-CBPE are not included in this summary because L-CBPE was evaluated only on the binary datasets (5 out of 15). The full table of results is reported in Appendix A.

Table 1. Evaluation of point estimation (AE) and predicted interval quality ($\mathcal{W}$) in terms of macro-averaged ranks; lower ranks are better. Boldface indicates the best result for a given row and measure; superscript † denotes results (if any) that are not statistically significantly different from the best one according to a Wilcoxon signed-rank test at significance level 0.05. Cells are colour coded to ease visual inspection, with darker green indicating better results.

	AE						$\mathcal{W}$					
	DoC	QuAcc	LEAP	SQS-H	SQS-S	BayCAP	DoC	QuAcc	LEAP	SQS-H	SQS-S	BayCAP
LR	3.27†	4.80	2.53†	**2.33**	3.47†	4.60	4.27	4.80	3.93	**2.33**	2.60†	3.07†
kNN	3.00†	5.00	**2.67**	**2.67**†	4.00†	3.67	4.20	5.00	4.00	**1.87**	3.13	2.80†
SVM	3.93	4.93	**2.07**	2.53†	3.80	3.73	4.87	4.53	3.53	**2.00**	2.80†	3.27†
MLP	2.80†	5.73	2.93†	**2.20**	3.93	3.40	3.80	5.07	4.00	**2.13**	3.00†	3.00†
RF	2.73†	4.73	2.73†	**2.60**	4.07	4.13	3.93	4.27	3.93	**2.47**	3.13†	3.27†

The results show that SQS-H is the most competitive of the proposed methods for point estimation. It obtains the best average rank, although the statistical significance tests indicate that DoC and, especially, LEAP also belong to the group of best-performing methods. The full results in Table 2 in Appendix A show that SQS-H is competitive (i.e., it either obtains the best result or is not statistically significantly different from the best) in 43 out of 75 classifier-dataset combinations. This places it ahead of LEAP and DoC, which obtain 38 and 29 such results, respectively.

SQS-H also obtains the strongest results in terms of interval quality, achieving the best average rank. The statistical significance tests indicate that SQS-S and, especially, BayCAP also belong to the group of best-performing methods. In terms of Winkler score, SQS-H is competitive (i.e., it either obtains the best result or is not statistically significantly different from the best) in 45 out of 80 classifier-dataset combinations, as shown in Table 2 in Appendix A. This substantially outperforms all competing methods. Although SQS-S and BayCAP are less competitive in terms of point estimation, their strong Winkler scores suggest that they can model uncertainty effectively, even when their central estimates are not always the most accurate. Overall, SQS-H achieves the best balance between point-estimation accuracy and interval quality.

It may seem surprising that SQS-S underperforms SQS-H, since SQS-S was designed to mitigate one of the limitations of SQS-H, namely that SQS-H compares true validation prevalence values with an estimated target prevalence. However, SQS-S does so by sacrificing part of the available labelled data. This creates a trade-off between a more sophisticated matching criterion and a less accurate quantifier, trained on less labelled information.

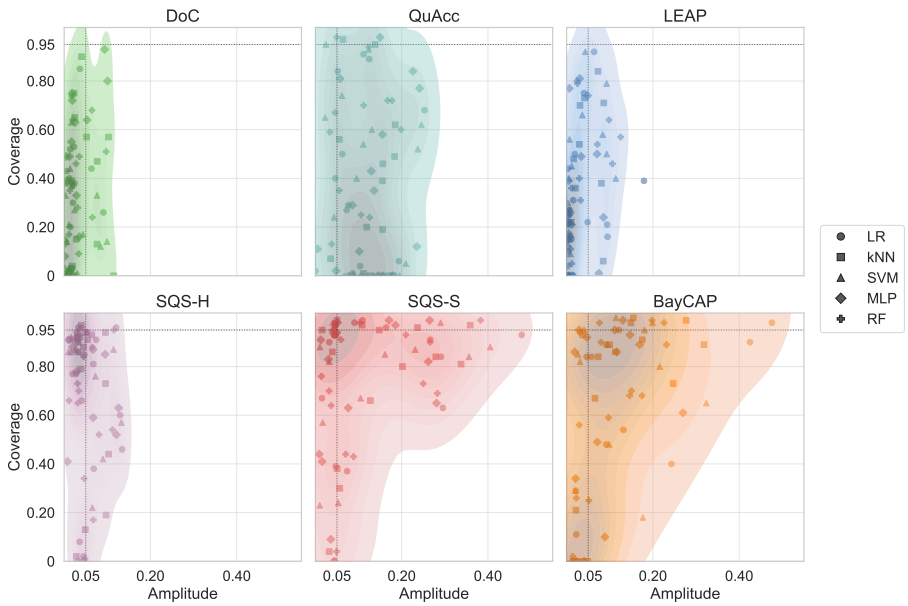

Fig. 1. Relationship between coverage-amplitude across different methods. Each point represents the average amplitude (x-axis) and coverage (y-axis) obtained by one method on one classifier-dataset combination across the 100 test sets generated via APP. Facets correspond to CAP methods, and markers distinguish the target classifier algorithms. The shaded regions show a two-dimensional kernel density estimate of the points in each facet. The horizontal dotted line marks the nominal coverage level 0.95, while the vertical dotted line marks a reference amplitude value established at 0.05. Points close to the intersection of both lines are desirable.

As explained in Section 6, L-CBPE is evaluated only on binary datasets. Although it obtains competitive point estimates in a few cases, its Winkler scores are poor. This behaviour is consistent with CBPE's dependence on calibrated posterior probabilities for the target distribution, a condition that is difficult to guarantee under PPS even when using LasCal.

Figure 1 provides a complementary view of the quality of the interval estimates by relating their amplitude to their coverage.[7] The plot clarifies an aspect not immediately visible from the Winkler scores alone. Among the baselines, DoC and LEAP tend to produce intervals with relatively small amplitude, often close to the reference value that we establish at 0.05, but their coverage is typically far below the nominal 0.95 level. QuAcc exhibits the same under-coverage issue,

[7] L-CBPE is not reported in Figure 1. Its results are less informative in this visualization, since they are available only for binary datasets and, in most cases, the method returns intervals with very small amplitude and, as a consequence, very low coverage, often close to zero.

while also producing intervals with larger amplitudes. This indicates that, although some bootstrap-based baselines may obtain competitive Winkler scores, their intervals are often not well calibrated as 95% interval estimates.

The proposed uncertainty-aware methods display a different behaviour. SQS-S and BayCAP generally achieve high coverage, but often at the cost of wide intervals. SQS-H instead shows a more balanced trade-off, with most results concentrated in regions of high coverage and moderate amplitude. This confirms the trend observed in Table 2: among the proposed methods, SQS-H is the one that most consistently balances accurate point estimates with reliable and informative interval estimates.

As a final remark, BayCAP has two features that are not exploited in our experimental setting. First, among the methods considered here, it is the only one that naturally allows prior knowledge to be incorporated. In our experiments, we use a uniform prior; however, in applications where prior knowledge about plausible target prevalence values is available, more informative priors could provide an advantage. Second, the Bayesian formulation allows the spread of the posterior distribution to be adjusted through a temperature parameter. The current implementation uses temperature equal to one, but this parameter could be tuned on validation sets to improve the calibration of the resulting uncertainty estimates. We leave the investigation of both directions to future work.

7 Conclusion

In this paper we have addressed classifier accuracy prediction under prior probability shift from an uncertainty-aware perspective. We have proposed three methods (SQS-H, SQS-S, BayCAP) that natively provide both point estimates of classifier performance and interval estimates around them. Our empirical comparison with existing CAP methods shows that SQS-H is the most effective among the evaluated methods, both in terms of point-estimation accuracy and uncertainty assessment. Among the proposed methods, BayCAP is, to the best of our knowledge, the first Bayesian approach to CAP. Although our experiments do not show BayCAP to outperform the other methods considered, its Bayesian formulation offers additional flexibility that remains unexplored in this work, i.e., the possibility of incorporating informative priors and calibrating posterior uncertainty through tempering. In future work, we plan to investigate these directions, as well as to extend the analysis to other types of dataset shift and to evaluation measures beyond vanilla accuracy.

Bibliography

[1] Bella, A., Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J.: Quantification via probability estimators. In: Proceedings of the 11th IEEE International Conference on Data Mining (ICDM 2010). pp. 737–742. Sydney, AU (2010).

[2] Bhaskaruni, D., Moss, F.P., Lan, C.: Estimating prediction qualities without ground truth: A revisit of the reverse testing framework. In: Proceedings of the

24th International Conference on Pattern Recognition (ICPR 2018). pp. 49–54. Beijing, CN (2018).
[3] Chen, J., Liu, F., Avci, B., Wu, X., Liang, Y., Jha, S.: Detecting errors and estimating accuracy on unlabeled data with self-training ensembles. In: Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021). pp. 14980–14992. Virtual Event (2021)
[4] Deng, W., Suh, Y., Gould, S., Zheng, L.: Confidence and dispersity speak: Characterizing prediction matrix for unsupervised accuracy estimation. In: Proceedings of the 40th International Conference on Machine Learning (ICML 2023). pp. 7658–7674. Honolulu, US (2023)
[5] Deng, W., Zheng, L.: AutoEval: Are labels always necessary for classifier accuracy evaluation? In: Proceedings of the 2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2021). pp. 15069–15078. Virtual Event (2021).
[6] Elsahar, H., Gallé, M.: To annotate or not? Predicting performance drop under domain shift. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019). pp. 2163–2173. Hong Kong, CN (2019).
[7] Esuli, A., Fabris, A., Moreo, A., Sebastiani, F.: Learning to quantify. Springer Nature, Cham, CH (2023).
[8] Forman, G.: Quantifying counts and costs via classification. Data Mining and Knowledge Discovery **17**(2), 164–206 (2008).
[9] Garg, S., Balakrishnan, S., Lipton, Z.C., Neyshabur, B., Sedghi, H.: Leveraging unlabeled data to predict out-of-distribution performance. In: Proceedings of the 10th International Conference on Learning Representations (ICLR 2022). Virtual Event (2022)
[10] Gneiting, T., Raftery, A.E.: Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association **102**(477), 359–378 (2007).
[11] Guillory, D., Shankar, V., Ebrahimi, S., Darrell, T., Schmidt, L.: Predicting with confidence on unseen distributions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021). pp. 1134–1144. Montreal, CA (2021).
[12] Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. In: Proceedings of the 5th International Conference on Learning Representations (ICLR 2017). Toulon, FR (2017)
[13] Hoffman, M.D., Gelman, A.: The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. Journal of Machine Learning Research **15**(1), 1593–1623 (2014).
[14] Hofman, P., Sale, Y., Hüllermeier, E.: Uncertainty quantification for machine learning: One size does not fit all. In: Proceedings of the 40th AAAI Conference on Artificial Intelligence (AAAI 2026). pp. 21726–21734. Singapore, SN (2026).
[15] Jiang, Y., Nagarajan, V., Baek, C., Kolter, J.Z.: Assessing generalization of SGD via disagreement. In: Proceedings of the International Conference on Learning Representations (ICLR 2022). Virtual Event (2022)
[16] Kelly, M., Longjohn, R., Nottingham, K.: The UCI machine learning repository, `https://archive.ics.uci.edu`
[17] Kivimäki, J., Bialek, J., Kuberski, W., Nurminen, J.K.: Performance estimation in binary classification using calibrated confidence. Machine Learning **115**(3), 67 (2026).

[18] Kivimäki, J., Nurminen, J.K., Białek, J., Kuberski, W.: Confidence-based estimators for predictive performance in model monitoring. Journal of Artificial Intelligence Research **82**, 209–240 (2025).
[19] Lipton, Z.C., Wang, Y., Smola, A.J.: Detecting and correcting for label shift with black box predictors. In: Proceedings of the 35th International Conference on Machine Learning (ICML 2018). pp. 3128–3136. Stockholm, SE (2018)
[20] Lu, Y., Qin, Y., Zhai, R., Shen, A., Chen, K., Wang, Z., Kolouri, S., Stepputtis, S., Campbell, J., Sycara, K.P.: Characterizing out-of-distribution error via optimal transport. In: Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2023). New Orleans, US (2023)
[21] Moreno-Torres, J.G., Raeder, T., Alaíz-Rodríguez, R., Chawla, N.V., Herrera, F.: A unifying view on dataset shift in classification. Pattern Recognition **45**(1), 521–530 (2012).
[22] Moreo, A., Esuli, A., Sebastiani, F.: QuaPy: A Python-based framework for quantification. In: Proceedings of the 30th ACM International Conference on Knowledge Management (CIKM 2021). pp. 4534–4543. Gold Coast, AU (2021).
[23] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
[24] Popordanoska, T., Radevski, G., Tuytelaars, T., Blaschko, M.B.: Lascal: Label-shift calibration without target labels. In: Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024). pp. 65386–65414. Vancouver, CA (2024)
[25] Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D. (eds.): Dataset shift in machine learning. The MIT Press, Cambridge, US (2009).
[26] Redyuk, S., Schelter, S., Rukat, T., Markl, V., Bießmann, F.: Learning to validate the predictions of black box machine learning models on unseen data. In: Proceedings of the Workshop on Human-In-the-Loop Data Analytics (HILDA@SIGMOD 2019). pp. 4:1–4:4. Amsterdam, NL (2019).
[27] Tasche, D.: Comments on Friedman's method for class distribution estimation. arXiv:2405.16666 [cs.LG] (2024)
[28] Volpi, L., Moreo, A., Sebastiani, F.: LEAP: Linear equations for classifier accuracy prediction under prior probability shift. Machine Learning **114**(12), Article 293 (2025).
[29] Volpi, L., Moreo, A., Sebastiani, F.: QuAcc: Using quantification to predict classifier accuracy under prior probability shift. Intelligenza Artificiale **19**(2), 141–157 (2025).
[30] Wang, L., Wang, X., Liao, K.P., Cai, T.: Semisupervised transfer learning for evaluation of model classification performance. Biometrics **80**(1) (2024).
[31] Xie, R., Wei, H., Feng, L., Cao, Y., An, B.: On the importance of feature separability in predicting out-of-distribution error. In: Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS 2023). pp. 27783–27800. New Orleans, US (2023)
[32] You, S.D., Liu, H., Liu, C.: Predicting classification accuracy of unlabeled datasets using multiple deep neural networks. IEEE Access **10**, 44627–44637 (2022).
[33] Ziegler, A., Czyż, P.: Bayesian quantification with black-box estimators. Transactions on Machine Learning Research **2024** (2024)

A Full table of results

Table 2 reports the full set of experiments in terms of AE and Winkler score ($\mathcal{W}$); lower is better for both measures. Boldface indicates the best result for a given row and measure; superscript † denotes results that are not statistically significantly different from the best one according to a Wilcoxon signed-rank test at significance level 0.01. The cell colouring is meant to ease visual inspection. For each evaluation measure and each classifier-dataset combination, colours are assigned locally within the corresponding row. Specifically, we consider the standard deviation for the distribution of scores obtained by all methods across all test sets. The method with the best mean score in the row is coloured green. Methods whose mean score differs from the best mean score by more than one standard deviation are coloured red, while intermediate cases are coloured according to a green-to-red gradient, based on their distance from the best mean.

Table 2. Results of the CAP experiments in terms of AE and Winkler score ($\mathcal{W}$).

		AE							$\mathcal{W}$						
		Baseline methods				Our methods			Baseline methods				Our methods		
		DoC	QuAcc	LEAP	L-CBPE	SQS-H	SQS-S	BayCAP	DoC	QuAcc	LEAP	L-CBPE	SQS-H	SQS-S	BayCAP
LR	spambase	0.014	0.031	0.006†	0.030	**0.006**	0.008†	0.007	0.299	0.438	0.206	36.623	**0.036**	0.054	0.038
	semeion	**0.009**	0.030	0.037	0.083	0.043	0.036	0.041	**0.063**	0.314	1.199	33.885	1.009	0.643	0.127
	german	**0.040**	0.176	0.055†	0.079	0.057	0.050†	0.078	0.674†	3.411	1.351	25.152	1.233	0.779	**0.579**
	tictactoe	**0.004**	0.024	0.006	0.013	0.007	0.007	0.011	0.086†	0.946	0.184	38.892	0.066	0.083†	**0.053**
	mammographic	0.086	0.112	0.059	0.064	0.060	0.070	**0.052**	3.435	2.734	2.350	30.723	1.529	1.966	**0.444**
	poker-hand	0.081	0.089	0.146	-	0.087†	**0.073**	0.246	3.211	2.153	3.185	-	1.445	**0.608**	0.864
	connect-4	0.079	0.028†	0.071	-	**0.026**	0.128	0.107	1.730	0.156	1.082	-	**0.142**	1.677	0.497
	shuttle	0.149	0.083	**0.012**	-	0.022	0.127	0.127	5.762	2.101	**0.078**	-	0.118	0.600	0.687
	chess	0.027	0.029	**0.015**	-	0.021	0.028	0.027	0.768	0.181	**0.137**	-	0.156	0.180	0.168
	letter	0.013†	0.131	0.013	-	**0.010**	0.017	0.065	0.228	3.570	0.273	-	**0.065**	0.102	1.965
	dry-bean	0.010	0.268	0.008	-	**0.007**	0.010	0.015	0.222	8.282	0.232	-	**0.041**	0.052	0.247
	nursery	0.021	0.016	**0.011**	-	0.018	0.019	0.021	0.465	0.120†	0.152†	-	0.197	0.110	**0.095**
	hand-digits	0.008†	0.313	0.007†	-	**0.007**	0.008†	0.021	0.132	10.431	0.170	-	**0.036**	0.053	0.440
	isolet	**0.006**	0.454	0.013	-	0.009†	0.007†	0.178	0.050	15.473	0.345	-	0.071†	**0.047**	6.387
	wine-quality	0.100	0.102	0.101	-	**0.059**	0.092	0.210	3.649	0.967†	2.314	-	0.797†	**0.497**	3.794
kNN	spambase	0.010	0.028	0.007†	0.021	0.007†	0.012	**0.007**	0.062	0.347	0.213	34.621	0.046	0.073	**0.043**
	semeion	0.040	0.310	**0.013**	0.037	0.013†	0.041	0.029	1.179	8.536	0.276	34.514	**0.095**	0.677	0.153
	german	0.209	0.153	**0.031**	0.076	0.034†	0.056	0.099	6.076	3.334	0.340†	22.927	**0.306**	0.585	0.422
	tictactoe	0.026	0.032	0.030	**0.020**	0.034	0.035	0.033	0.693	0.577	0.980	35.705	0.820	0.783	**0.173**
	mammographic	**0.030**	0.127	0.040	0.108	0.041	0.033	0.034	0.986	3.417	1.587	29.739	0.714	0.502	**0.118**
	poker-hand	0.071	0.131	0.078	-	0.074	**0.051**	0.110	2.513	2.563	1.975	-	1.502	**0.253**	1.490
	connect-4	0.046	0.026†	**0.022**	-	0.023†	0.061	0.031	0.717	0.159	0.162	-	**0.133**	0.478	0.155
	shuttle	0.112	0.143	0.014	-	**0.011**	0.118	0.034	3.092	3.245	0.123†	-	**0.055**	0.693	0.182
	chess	0.026	0.098	0.016†	-	**0.015**	0.030	0.022	0.383†	1.768	0.249†	-	**0.113**	0.174	0.165
	letter	**0.007**	0.521	0.013	-	0.009†	0.010	0.078	0.067†	17.657	0.345	-	**0.046**	0.051	2.684
	dry-bean	0.013	0.273	0.010	-	**0.008**	0.010	0.017	0.229	8.591	0.295	-	**0.055**	0.065	0.329
	nursery	0.056	**0.010**	0.010†	-	0.012	0.075	0.034	0.851	0.073	0.150†	-	**0.071**	0.474	0.183
	hand-digits	**0.003**	0.655	0.004†	-	0.003†	0.003†	0.022	0.068	23.002	0.101	-	**0.018**	0.019	0.631
	isolet	**0.010**	0.308	0.015	-	0.011†	0.020	0.159	0.112†	9.885	0.309	-	**0.061**	0.122	5.455
	wine-quality	0.071	0.088	0.121	-	0.108	0.095	**0.053**	2.337	1.171†	3.073	-	2.603	0.381	**0.281**
SVM	spambase	0.018	0.038	**0.008**	0.031	0.008†	0.009†	0.008	0.616	0.748	0.254	36.661	**0.044**	0.071†	0.058†
	semeion	0.068	0.022	0.008	0.066	0.007†	**0.007**	0.016	1.934	0.193	0.159	36.596	0.046	**0.043**	0.114
	german	0.091	0.254	0.042†	0.164	0.066	**0.039**	0.159	2.106	6.394	0.593†	22.955	1.262	**0.413**	2.456
	tictactoe	0.013	0.007	**0.004**	0.022	0.004†	0.016	0.005	0.310	0.116	0.073	39.050	**0.026**	0.442	0.042
	mammographic	0.056	0.054†	0.062	0.077	0.063	0.061†	**0.054**	1.661	0.674	2.387	29.609	1.609	1.527	**0.398**
	poker-hand	0.079	0.132	**0.052**	-	0.082	0.053†	0.079†	2.934	1.920	0.748†	-	2.070	**0.293**	1.126
	connect-4	0.055	0.056	0.033†	-	**0.031**	0.063	0.040†	1.005	0.527	0.236†	-	**0.184**	0.432	0.261
	shuttle	0.119	0.074	**0.012**	-	0.013	0.101	0.031	2.981	1.375	0.094	-	**0.086**	0.524	0.200
	chess	0.035	0.025	**0.016**	-	0.024	0.041	0.022	0.998	0.144	0.185†	-	0.223†	0.294	**0.143**
	letter	**0.008**	0.443	0.014	-	0.008†	0.009†	0.083	0.097†	15.119	0.406	-	**0.050**	0.052	2.880
	dry-bean	0.009	0.323	0.008	-	**0.007**	0.009	0.014	0.174	8.525	0.227	-	**0.038**	0.053	0.243
	nursery	0.008	0.005	**0.003**	-	0.005	0.010	0.005	0.256	**0.030**	0.039†	-	0.035	0.136	0.035†
	hand-digits	**0.003**	0.480	0.003†	-	0.003†	0.003†	0.021	0.058	18.018	0.067	-	**0.019**	0.023	0.614
	isolet	0.013	0.435	0.013	-	**0.008**	0.010	0.178	0.235	15.143	0.343	-	**0.055**	0.055	6.404
	wine-quality	0.109	0.111	0.077	-	**0.065**	0.104	0.149	4.265	1.313†	1.323	-	0.785†	**0.527**	1.059†
MLP	spambase	0.009	0.017	0.006†	0.043	0.006†	0.009	**0.006**	0.133	0.135	0.199	37.105	**0.034**	0.066†	0.036
	semeion	**0.013**	0.067	0.035	0.079	0.042	0.055	0.036	0.168†	1.656	1.141	34.740	1.072	1.503	**0.121**
	german	0.021†	0.067	**0.017**	0.056	0.029	0.035	0.042	**0.115**	0.416	0.160†	25.580	0.314†	0.342†	0.163
	tictactoe	0.005†	0.028	**0.004**	0.017	0.005†	0.012	0.007	0.116	0.968	0.104	38.707	**0.031**	0.195	0.049
	mammographic	0.085	0.085	0.076	0.075	0.078	0.080	**0.071**	3.009	1.467	2.929	30.551	2.278	2.337	**1.012**
	poker-hand	0.062	0.088	0.099	-	0.052†	0.050†	**0.041**	1.875	0.902	2.247	-	0.665†	**0.268**	0.270
	connect-4	0.033†	0.029†	0.035†	-	**0.027**	0.037	0.028†	0.247	**0.161**	0.359†	-	0.179	0.198	0.181
	shuttle	0.024	0.144	**0.009**	-	0.014	0.092	0.017	0.253	5.075	**0.074**	-	0.088	0.543	0.131
	chess	0.022†	0.082	0.020†	-	**0.018**	0.026†	0.034	0.471	1.072	0.351†	-	**0.128**	0.175	0.380
	letter	**0.006**	0.651	0.011	-	0.007	0.008	0.080	0.057	22.997	0.297	-	**0.046**	0.050	2.782
	dry-bean	**0.007**	0.137	0.008†	-	0.008†	0.011	0.014	0.121	3.739	0.242	-	**0.048**	0.061	0.231
	nursery	0.009	0.105	**0.002**	-	0.008	0.009	0.003	0.320	2.062	**0.016**	-	0.176	0.180	0.026
	hand-digits	0.003†	0.506	0.003	-	**0.002**	0.003†	0.022	0.046	17.027	0.081	-	0.015	**0.015**	0.651
	isolet	**0.007**	0.449	0.014	-	0.009†	0.009†	0.177	0.058	15.271	0.361	-	0.085	**0.052**	6.332
	wine-quality	0.112	0.200	0.169	-	**0.072**	0.125	0.139	4.323	3.706	5.246	-	1.226†	**1.018**	1.674
RF	spambase	**0.012**	0.018	0.012†	0.016†	0.012†	0.013†	0.014†	0.439	0.150	0.420	37.277	**0.087**	0.094	0.132
	semeion	0.020	0.031	**0.008**	0.031	0.011	0.054	0.020	0.420	0.293	**0.071**	31.303	0.077	1.074	0.198
	german	**0.028**	0.058	0.037	0.093	0.044	0.052	0.093	**0.373**	0.462	0.524	24.530	0.676	0.715	0.502
	tictactoe	**0.008**	0.018	0.009†	0.017	0.012	0.014	0.009†	0.102†	0.282†	0.176	36.179	0.136†	0.154†	**0.077**
	mammographic	0.049†	**0.046**	0.055	0.090	0.055	0.048†	0.048†	1.365	0.357†	2.182	29.691	1.286	0.986	**0.246**
	poker-hand	0.078†	0.099†	**0.063**	-	0.095	0.067†	0.084†	2.851	1.185†	0.817†	-	2.508	**0.360**	1.629†
	connect-4	0.073	0.028†	0.042	-	**0.023**	0.087	0.042	1.857	0.146	0.346†	-	**0.143**	0.815	0.247
	shuttle	**0.020**	0.048	0.031	-	0.037	0.125	0.058	0.286†	1.205	0.703	-	0.684	1.062	**0.228**
	chess	**0.016**	0.106	0.021†	-	0.017†	0.018†	0.035	0.347	2.076	0.402	-	**0.102**	0.115	0.557
	letter	0.010	0.574	0.014	-	**0.008**	0.008†	0.084	0.197	19.554	0.422	-	**0.039**	0.048	2.931
	dry-bean	0.013	0.127	0.009	-	**0.008**	0.011	0.016	0.259	2.946	0.260	-	**0.052**	0.066	0.289
	nursery	0.034	0.007†	**0.006**	-	0.009	0.047	0.016	0.846	0.050	0.057	-	**0.047**	0.260	0.095
	hand-digits	0.006	0.563	**0.003**	-	0.005	0.006	0.018	0.181	20.899	0.069†	-	0.067†	**0.052**	0.459
	isolet	0.016	0.524	0.021	-	**0.013**	0.014†	0.180	0.431	18.006	0.641	-	0.119†	**0.076**	6.432
	wine-quality	**0.050**	0.060†	0.067†	-	0.058†	0.067†	0.057†	0.722†	0.615†	1.268	-	0.887†	0.385	**0.349**