Transductive Model Selection under Prior Probability Shift

Lorenzo Volpi[1,*], Alejandro Moreo[1] and Fabrizio Sebastiani[1]

[1] Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Via Giuseppe Moruzzi 1, 56124, Pisa, Italy

Abstract

Transductive learning is a supervised machine learning task in which, unlike in traditional inductive learning, the unlabelled data that require labelling are a finite set and are available at training time. Similarly to inductive learning contexts, transductive learning contexts may be affected by dataset shift, i.e., may be such that the assumption according to which the training data and the unlabelled data are independently and identically distributed (IID), does not hold. We here propose a method, tailored to transductive classification contexts, for performing model selection (i.e., hyperparameter optimisation) when the data exhibit prior probability shift, an important type of dataset shift typical of anti-causal learning problems. In our proposed method the hyperparameters can be optimised directly on the unlabelled data to which the trained classifier must be applied; this is unlike traditional model selection methods, that are based on performing cross-validation on the labelled training data. By tailoring model selection to the actual test distribution, our approach contributes to the trustworthiness of AI systems, as it enables more reliable and robust classifier deployment under changed conditions. We provide experimental results that show the benefits brought about by our method.

Keywords

Model selection, Hyperparameter optimisation, Classifier accuracy prediction, Dataset shift, Prior probability shift, Transductive learning

1. Introduction

A key requirement for trustworthy AI is *robustness under dataset shift* (i.e., robustness to scenarios in which the assumption that data are *identically and independently distributed* (IID) is not verified), as models trained and validated under the IID assumption often face reliability issues when deployed in real-world, dynamic scenarios. In applications where the distribution of the priors may vary unexpectedly, relying on traditional cross-validation for accuracy evaluation and, as a consequence, hyperparameter selection, can result in biased and misleading estimates of model performance.

Consider the outbreak of an epidemic, in which the prevalence of individuals affected by an infectious disease rapidly increases while the distribution of the symptoms (i.e., the effects) across the affected individuals remains unchanged. In such a scenario, we may want to train a classifier that, from the symptoms an individual displays, infers whether she is affected or not from the disease. We consider the situation in which (i) the data to be classified arrives in successive batches $U_1, U_2, \ldots$, and, (ii) given the epidemic, we may expect the prevalence of the affected individuals to evolve rapidly across batches. Assume that, for training the classifier, and in particular for selecting the combination of hyperparameters expected to yield the best accuracy under the new (i.e., epidemic) conditions, we have access to training data collected in the old (i.e., pre-epidemic) conditions. This scenario is problematic, as the training data and the unlabelled data to be classified are not IID, due to the fact that the prevalence of the individuals affected by the disease has changed from the training data to the unlabelled data. The classifier, and the chosen hyperparameter combination, may thus reveal suboptimal once used on the unlabelled data.

In this paper we present a method for optimising the hyperparameters (i.e., for performing *model selection* – MS) *directly on the batch of unlabelled data that need to be classified.* For reasons that will be explained in Section 2, we call this task *transductive model selection* (TMS). A TMS method has obvious

TRUST-AI: The European Workshop on Trustworthy AI. Organized as part of the European Conference of Artificial Intelligence - ECAI 2025. October 2025, Bologna, Italy.
*Corresponding author.

lorenzo.volpi@isti.cnr.it (L. Volpi); alejandro.moreo@isti.cnr.it (A. Moreo); fabrizio.sebastiani@isti.cnr.it (F. Sebastiani)
0009-0006-0851-8041 (L. Volpi); 0000-0002-0377-1025 (A. Moreo); 0000-0003-4221-6427 (F. Sebastiani)

advantages over the standard inductive model selection (IMS) method (that relies on cross-validation on the training data), since the chosen hyperparameters are tailored to the batch U_i of unlabelled data that need to be classified, and can thus deliver better performance on U_i than hyperparameters chosen via standard IMS. By moving from a "one-size-fits-all" approach to a context-aware solution, our method enhances the reliability of model selection under dataset shift, thereby contributing to more robust and trustworthy decision-making processes.

Our proposed method is based on techniques for *classifier accuracy prediction* (CAP) *under dataset shift* [1, 2, 3], and essentially consists of (i) predicting the accuracy that different classifiers, instantiated with different choices of hyperparameters, would obtain on our "out-of-distribution" batch U_i, and (ii) classifying the data in batch U_i using the classifier whose predicted accuracy is highest. In other words, our TMS method replaces traditional accuracy *computation* on *labelled* data with accuracy *estimation* on *unlabelled* data. While the method is generic, we here restrict our attention to the case in which the data are affected by *prior probability shift* (PPS), an important type of violation of the IID assumption.

For high-stakes applications such as the healthcare-related classification task discussed above, this suggests a policy of (a) training, on the labelled data, multiple classifiers, each characterised by a different combination of hyperparameters,[1] (b) storing these classifiers for later use, and, every time a new batch U_i of unlabelled data becomes available, (c) estimating (via CAP techniques) the accuracy that the different classifiers would have on U_i and (d) classifying the data in U_i via the classifier whose estimated accuracy is highest. In this way, assuming that Step (b) can be carried out efficiently, the classification of newly arrived data can be performed immediately and, as we will show, with a much higher accuracy than can be obtained via the traditional model selection method. Note that Step (a) is carried out only once, since we do not assume new labelled data to become available during the process.

The rest of the paper is organised as follows. Section 2 introduces the notation and provides a detailed description of the proposed method. Section 3 presents the experiments we have carried out and discusses the results we have obtained. Section 4 concludes the paper with a summary of our findings and a discussion of potential applications of this method.

2. Transductive Model Selection under Prior Probability Shift

In supervised machine learning we use training data to learn the internal parameters of the model (e.g., the weights of a neural network, or the coefficients of a hyperplane in a support vector machine). Many models trained in this way also rely on a set of hyperparameters (e.g., the learning rate in neural training, or the trade-off between margin and training error in support vector machines) that impose higher-level constraints on the learning process. Unlike the internal parameters of the model, hyperparameters are not learned during training, but must be set in advance. Finding good values for the hyperparameters is crucial for achieving good performance. Model selection, the task of choosing the values of the hyperparameters, is typically carried out by (a) testing the accuracy of the model under different combinations of hyperparameter values using cross-validation on labelled data, and (b) choosing the combination of hyperparameter values that maximizes model accuracy.

Relying on labelled data to evaluate differently configured models requires the labelled data to be representative of the unlabelled data the trained model will be applied to, a distributional assumption typically referred to as the IID assumption. Unfortunately, in real-world problems this assumption is often violated; in this case, the training data are *not* representative of the unlabelled data (which are thus said to be "out-of-distribution" data), and we say that the problem is affected by *dataset shift* [4]. In problems characterised by dataset shift, cross-validation on training data is thus a biased estimator of model accuracy [5], and this often leads to suboptimal choices of the hyperparameter values.

In classification, one type of dataset shift of particular relevance is *prior probability shift* (PPS) [4], also known as *label shift* [6]. This type of shift (sometimes considered the "paradigmatic" case of dataset shift in classification [7]) is characteristic of *anti-causal learning* problems [8] (also known as $Y \rightarrow X$ *problems* [9], where Y is a random variable ranging on the class labels and X is a random variable

[1]Note that this step is performed anyway during traditional hyperparameter optimisation.

ranging on vectors of covariates), i.e., problems where the goal is to predict the causes of a phenomenon from its observed effects.

PPS is characterised by two distributional assumptions, often called the *PPS assumptions*, i.e., (i) the class priors of the training distribution differ from those of the distribution of the unlabelled data (in symbols: $P(Y) \neq Q(Y)$, where P and Q are the distributions from which the training data and the unlabelled data are sampled, respectively); and (ii) the class-conditional distribution of the covariates in P is the same as that in Q (in symbols: $P(X|Y) = Q(X|Y)$).

The healthcare-related problem discussed in Section 1 is indeed an anti-causal learning problem. Indeed, if we take random variable Y to range over $\mathcal{Y} = \{\mathsf{Disease}, \mathsf{NoDisease}\}$, and random variable X to range over the vectors of covariates representing symptoms exhibited by individuals, the anti-causal nature of the problem is evident. If we take P and Q to be the data distributions characterizing the pre-epidemic and the epidemic scenarios, respectively, we are in the presence of PPS, since $P(Y) \neq Q(Y)$ (the prevalence values of Disease and NoDisease have changed when switching from P to Q) and $P(X|Y) = Q(X|Y)$ (the distributions of the symptoms exhibited by affected individuals are the same in P and Q).

In the presence of PPS (as in the presence of any other type of shift, for that matter), a classifier whose hyperparameters have been optimised on data from P may behave suboptimally when applied to data from Q (see Section 2.1 for a formal proof). For it to behave optimally on data from Q, hyperparameter optimisation should have been carried out on data from Q, but this is not possible if using standard cross-validation techniques, since the labels of data from Q are not known.

To address this problem, we introduce *transductive model selection* (TMS), a new strategy aimed at selecting the hyperparameter configuration for a given classifier (or the model from a pool of already trained candidates) that is predicted to be the best for a specific batch U_i of unlabelled data characterised by dataset shift. This strategy leverages recent advances in classifier accuracy prediction (CAP) [1, 2, 3] (a family of techniques specifically designed to estimate classifier accuracy under dataset shift), and focuses in particular on CAP methods tailored to PPS [3].

2.1. 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) estimator $\widehat{\mathrm{Acc}}$ of the classifier's accuracy obtained by means of cross-validation on training data. Assume our unlabelled data is drawn from a distribution Q related to P via PPS: can we trust our estimate? This amounts to asking whether our estimator is unbiased under PPS, i.e., whether $\mathrm{Bias}(\widehat{\mathrm{Acc}}) \equiv \mathbb{E}[\widehat{\mathrm{Acc}}] - Q(\hat{Y} = Y) = 0$, where $\hat{Y}$ is a random variable ranging on the predicted class labels. As our training data is drawn from distribution P, and since our estimate is arbitrarily good, asymptotically it holds that $\mathbb{E}[\widehat{\mathrm{Acc}}] = P(\hat{Y} = Y)$. For simplicity, let us focus on a generic binary problem (with $\mathcal{Y} = \{0, 1\}$). Note that

$$\begin{aligned} P(\hat{Y} = Y) &= P(\hat{Y} = 1, Y = 1) + P(\hat{Y} = 0, Y = 0) \\ &= P(\hat{Y} = 1|Y = 1)P(Y = 1) + P(\hat{Y} = 0|Y = 0)P(Y = 0) \end{aligned} \tag{1}$$

and that, similarly,

$$Q(\hat{Y} = Y) = Q(\hat{Y} = 1|Y = 1)Q(Y = 1) + Q(\hat{Y} = 0|Y = 0)Q(Y = 0) \tag{2}$$

We first observe that, as shown in [6, Lemma 1], the PPS assumption $P(X|Y) = Q(X|Y)$ (see Section 2) implies that $P(f(X)|Y) = Q(f(X)|Y)$ for any deterministic and measurable function f. In particular, if we take f to be our classifier h, it holds that $P(\hat{Y}|Y) = Q(\hat{Y}|Y)$. This means that $P(\hat{Y} = 1|Y = 1)$ and $Q(\hat{Y} = 1|Y = 1)$ are equal; we indicate both by the symbol "tpr", since they both represent the *true positive rate* of the classifier. Similarly, $P(\hat{Y} = 0|Y = 0)$ and $Q(\hat{Y} = 0|Y = 0)$ are equal, and we indicate both as "tnr", which stands for the *true negative rate* of the classifier. We can further simplify our equations via the shorthands $p = P(Y = 1)$ and $q = Q(Y = 1)$. It then follows that

$$\begin{aligned} \mathrm{Bias}(\widehat{\mathrm{Acc}}) &= \mathrm{tpr} \cdot p + \mathrm{tnr} \cdot (1 - p) - (\mathrm{tpr} \cdot q + \mathrm{tnr} \cdot (1 - q)) \\ &= (p - q) \cdot (\mathrm{tpr} - \mathrm{tnr}) \end{aligned} \tag{3}$$

PPS means that $p \neq q$; Equation 3 thus implies that $\mathrm{Bias}(\widehat{\mathrm{Acc}}) = 0$ holds only if $(\mathrm{tpr} - \mathrm{tnr}) = 0$. However, $\mathrm{tpr} = \mathrm{tnr}$ is not true in general (and is unlikely to be true in practice), thus implying that *the cross-validation estimator is biased under PPS.* A similar reasoning holds for the multiclass case.

2.2. Model Selection: From Induction to Transduction

Let us assume the following problem setting. Let Θ be the set of all assignments of values to hyperparameters that we want to explore as part of our model selection process; in this paper we will concentrate on a standard *grid search* exploration, although other strategies (e.g., Gaussian processes, randomized search) might be used instead. Let L be our (labelled) training set and U_i our batch of (unlabelled) data. Consider the class $\mathcal{H}$ of hypotheses, and let $h^\theta \in \mathcal{H}$ be the classifier with hyperparameters θ trained via some learning algorithm A using labelled data L. Let $M : \mathcal{H} \times \mathcal{U} \rightarrow \mathbb{R}$ be the measure of accuracy for a classifier $h \in \mathcal{H}$ on batch $U_i \in \mathcal{U}$ of unlabelled data we want to optimise h for. The model selection problem can thus be formalized as

$$\theta^* = \arg\max_{\theta \in \Theta} M(h^\theta, U_i) \tag{4}$$

Since we do not have access to the labels in U_i, the problem cannot be solved directly, and we must instead resort to approximations. The most common way for solving it corresponds to the traditional *inductive model selection* method (IMS – Section 2.2.1).

2.2.1. Inductive Model Selection

The IMS approach comes down to using part of the training data for evaluating each configuration of hyperparameters, based on the assumption that such an estimate of classifier accuracy generalizes for future data. In this paper we carry this out via standard cross-validation (although everything we say applies to k-fold cross-validation as well), splitting L (with stratification) into a proper training set L_{tr} and a validation set L_{va}. IMS is described in Algorithm 1.

However, since IMS is unreliable under PPS for the reasons discussed in Section 2.1, we propose an alternative model selection method called *transductive model selection* (TMS – Section 2.2.2).

2.2.2. Transductive Model Selection

The main difference between IMS and TMS lies in the accuracy estimation step. Unlike IMS, which estimates the accuracy on unlabelled data by computing accuracy on labelled (validation) data, TMS estimates the accuracy on unlabelled data directly on the available set of unlabelled data. To this aim, TMS employs a classifier accuracy prediction (CAP) method, i.e., a predictor $\psi_h : \mathcal{U} \rightarrow \mathbb{R}$ of the accuracy that h will exhibit on a batch $U_i \in \mathcal{U}$ of unlabelled data. However, this does not mean that TMS can avoid using part of the labelled data, since it still requires a portion of it to train the CAP method. Since the procedure is transductive, its outcome is not a generic classifier that can be applied to any future data, but the set of labels assigned to the unlabelled instances in U_i. TMS is described in Algorithm 2.

3. Experiments

In this section we present an experimental comparison between IMS and TMS under PPS.[2]

Experimental Protocol. The effectiveness measure we use in order to assess the quality of the model selection strategies is the ("vanilla") accuracy of the selected model, i.e., the fraction of classification decisions that are correct.

The experimental protocol we adopt is as follows. Given a dataset D, we split it into a training set L (70%) and a test set U (30%) via stratified sampling; we further split the training set into a "proper"

[2]The code to reproduce all our experiments is available on GitHub at https://github.com/lorenzovolpi/tms

Algorithm 1: Inductive Model Selection

```
for $\theta \in \Theta$ do
    $h_\theta \leftarrow A(\mathcal{H}, \theta, L_{\mathrm{tr}})$
    // Trains the classifier via algorithm $A$
    $\mathrm{Acc} \leftarrow M(h_\theta, L_{\mathrm{va}})$
    // Computes accuracy on validation data
    if $\mathrm{Acc} > \mathrm{BestAcc}$ then
        $\mathrm{BestAcc} \leftarrow \mathrm{Acc}$
        $h_\theta^* \leftarrow h_\theta$
    end if
end for
return $h_\theta^*$
// Returns an inductive classifier that can be
// applied to any set of unlabelled data
```

Algorithm 2: Transductive Model Selection

```
for $\theta \in \Theta$ do
    $h_\theta \leftarrow A(\mathcal{H}, \theta, L_{\mathrm{tr}})$
    // Trains classifier $h_\theta$ via algorithm $A$
    $\psi_{h_\theta} \leftarrow \mathrm{CAP}(h_\theta, L_{\mathrm{va}})$
    // Trains a CAP method for classifier $h_\theta$
    $\widehat{\mathrm{Acc}} \leftarrow \psi_{h_\theta}(U_i)$
    // Estimates accuracy on the unlabelled data
    if $\widehat{\mathrm{Acc}} > \mathrm{BestAcc}$ then
        $\mathrm{BestAcc} \leftarrow \widehat{\mathrm{Acc}}$
        $h_\theta^* \leftarrow h_\theta$
    end if
end for
return $\{(x_j, h_\theta^*(x_j)) : x_j \in U_i\}$
// Returns the inferred labels for the specific
// unlabelled data
```

training set L_{tr} and a validation set L_{va}, with $|L_{\mathrm{tr}}| = |L_{\mathrm{va}}|$, via stratification. In order to simulate PPS we apply the Artificial Prevalence Protocol (APP) [10] on U. This consists of drawing r vectors $\mathbf{v}_1, ..., \mathbf{v}_r$ (we here take $r = 1000$) of n prevalence values (with n the number of classes) from the unit simplex Δ^{n-1} (using the Kraemer sampling algorithm [11]), and extracting from U, for each $\mathbf{v}_i$, a bag U_i of $|U_i| = s$ elements (we here take $s = 100$) such that U_i satisfies the prevalence distribution of $\mathbf{v}_i$.[3] The advantage of the APP is that it allows us to test the robustness of our models to the entire range of PPS values, while embodying a bag extraction method that implements exactly the two distributional assumptions, mentioned in Section 2, that lie at the heart of PPS.

We then run our experiments by first training all the differently configured classifiers on L_{tr} and:

1. for applying IMS: computing the accuracy of each trained classifier on L_{va} and classifying the datapoints in all the U_i's via the classifier that has shown the best accuracy;
2. for applying TMS: for each U_i, estimating the accuracy on U_i of each trained classifier via a CAP method trained on L_{va}, and applying to U_i the classifier that has shown the best accuracy.

Classifiers and Hyperparameters. We test both model selection approaches on four classifier types, namely, classifiers trained via Logistic Regression (LR), k-Nearest Neighbours (k-NN), Support Vector Machines (SVM), and Multi-Layered Perceptron (MLP). Each classifier type is instantiated with multiple combinations of hyperparameters, with the total number of combinations depending on the number of classes in the dataset and on the classifier type.

Under PPS, one of the most interesting hyperparameters is probably the `class_weight` hyperparameter of LR and SVM, which allows rebalancing the relative importance of the classes to compensate for class imbalance. In the presence of PPS, exploring different class-balancing configurations increases the probability of instantiating a classifier trained according to a class importance scheme that fits well the unlabelled data. For LR and SVM, we consider different values for the `class_weight` hyperparameter depending on the number n of classes. In all cases, we include the configurations `balanced` (which assigns different weights to instances of different classes to compensate for class imbalance in the training data) and `None` (all instances count the same, which results in more popular classes dominating the learning process). Aside from these, we explore alternative `class_weight` values that try to compensate potentially high values of a single class, one class at a time. The reason why we limit ourselves to this kind of exploration is to prevent combinatorial explosion; focusing on more than one class at a time, or on more weight values, would result in a potentially un-manageable number of hyperparameter combinations, especially for high values of n. These alternative values must be specified as points in the probability simplex (i.e., the per-class balancing weights must add up to one).

[3]We use the term "bag" (i.e., multiset) since we sample with replacement, which might lead to U_i containing duplicates.

In multiclass problems with $n > 2$, we add n such configurations to the pool of values, which we obtain as all different permutations composed of one "high" weight and $(n - 1)$ "low" weights. We set the high value to $2/n$ (i.e., twice the mass of a uniform assignment) and distribute the remainder among the low values, thus setting each to $\left(\frac{1-2/n}{n-1}\right)$.[4] For example, when $n = 3$ we explore the `class_weight` assignments $(0.66, 0.165, 0.165)$, $(0.165, 0.66, 0.165)$, and $(0.165, 0.165, 0.66)$. In the binary setting, where a finer-grained set of combinations is manageable, we instead use a grid of class weights G and explore all combinations $(g, 1 - g)$ with $g \in G$. In particular, we use the grid $G = (0.2, 0.4, 0.6, 0.8)$, thus considering the `class_weight` assignments $(0.2, 0.8)$, $(0.4, 0.6)$, $(0.6, 0.4)$, and $(0.8, 0.2)$.

Concerning the other hyperparameters, we consider five different values (10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}) for hyperparameter `C` (the regularization strength for both LR and SVM), as well as two additional values of `gamma` (`scale` and `auto`) for SVM only. For k-NN we explore five values of k, i.e., of `n_neighbors` (5, 7, 9, 11, 13) and two values of `weights` (`uniform` and `distance`). For MLP, we test five values of `alpha` (10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}) and two values of `learning_rate` (`constant` and `adaptive`).[5]

Datasets. We use the 25 datasets from the UCI machine learning repository[6] that can be directly imported through UCI's Python API and that have at least 6 features and at least 800 instances. The number of instances per dataset varies from 830 (mammographc) to 1,024,985 (poker-hand), while the number of features varies from 6 (mhr) to 617 (isolet). The number of classes varies from 2 (german, mammographic, semeion, spambase, tictactoe) to 26 (isolet, letter). Class balance is highly variable, from datasets in which some of the classes represent less than 1% of the instances (e.g., one of the classes in poker-hand), to perfectly balanced datasets (e.g., image-seg).

Implementation Details for TMS. As our choice of the CAP method we adopt O-LEAP$_{\text{KDEy}}$, a member of the *Linear Equations for Accuracy Prediction* (LEAP) family [3] specifically devised for PPS. LEAP methods work by estimating the values of the cells of the contingency table deriving from the application of the classifier to the set of unlabelled data, where the estimation is obtained by solving a system of linear equations that represent the problem constraints (including the PPS assumptions); once the contingency table is estimated, any classifier accuracy measure can be computed from it. LEAP internally relies on a *quantifier* (i.e., a predictor of class prevalence values) [10]; following [3], for this purpose we employ the KDEy-ML quantification method [12]. From now on, we will refer to the TMS method that uses O-LEAP$_{\text{KDEy}}$ simply as TMS-All, since this method selects the best model across all classifier types (LR, SVM, k-NN, MLP) and all their hyperparameter combinations.

Baselines. We compare TMS-All against standard (inductive) approaches for model selection. In particular, we consider two variants of IMS: one in which model selection is performed independently for each classifier type (IMS-LR, IMS-SVM, etc.), and another in which model selection chooses among all classifier types and all hyperparameter combinations for each type (IMS-All); in other words, TMS-All stands to TMS as IMS-All stands to IMS. We also compare these model selection strategies against configurations of each classifier ($\emptyset$-LR, $\emptyset$-SVM, etc.) in which default hyperparameters are used.

We also consider $\emptyset$-TSVM, an instance of the transductive support vector machine (TSVM) algorithm [13], which directly infers the label of each unlabelled datapoint without generating a classifier.[7] While TSVM relies on the IID assumption and is not a proper model selection approach, we include it as a reference baseline because it captures the essence of transductive learning, and thus offers a meaningful point of comparison. For TSVM, we only consider instantiations with default hyperparameters since, to the best of our knowledge, there is no established way to tune hyperparameters for non-IID settings.

[4]Notice that the choice of $2/n$ as the high value is arbitrary, but we consider it a good choice to compensate for class imbalance.
[5] Hyperparameter names and default configurations follow those provided by the *scikit-learn* library (https://scikit-learn.org/); we have left all hyperparameters not explicitly discussed here at their default values.
[6]https://archive.ics.uci.edu/
[7]For TSVM we use the implementation proposed by [14].

Table 1
Classification accuracy results, obtained via classifiers whose hyperparameters have been optimised either via IMS or via TMS. **Boldface** represents the best result obtained for the given dataset. Superscript † denotes the methods (if any) whose scores are not statistically significantly different from the best one according to a Wilcoxon signed-rank test at 0.01 confidence level. Cells are colour-coded in order to facilitate readability, with a bright green (resp., red) cell indicating the best (resp., worst) system on the given dataset, and paler shades indicating intermediate performance values. Values after ± in each cell represent standard deviation.

	$\emptyset$					IMS					TMS
	$\emptyset$-LR	$\emptyset$-kNN	$\emptyset$-SVM	$\emptyset$-MLP	$\emptyset$-TSVM	IMS-LR	IMS-kNN	IMS-SVM	IMS-MLP	IMS-All	TMS-All
poker-hand	.121 ± .104	.133 ± .080	.139 ± .090	**.169 ± .086**	.137 ± .089	.122 ± .109	.133 ± .083	.153 ± .082	.150 ± .089	.150 ± .089	.134 ± .110
connect-4	.515 ± .198	.544 ± .146	.539 ± .201	.629 ± .136	.334 ± .241	.516 ± .198	.512 ± .184	.649 ± .114	.632 ± .140	.649 ± .114	**.653 ± .142**
shuttle	.716 ± .188	.855 ± .114	.790 ± .162	.892 ± .087	.409 ± .198	.743 ± .190	.866 ± .106	**.900 ± .079**	.879 ± .096	.879 ± .096	.880 ± .098
chess	.340 ± .065	.382 ± .061	.409 ± .068	.547 ± .068	.074 ± .062	.346 ± .066	.449 ± .058	**.564 ± .059**	.547 ± .068	**.564 ± .059**	.561† ± .067
letter	.763 ± .047	.902 ± .031	.911 ± .030	.928 ± .027	.055 ± .041	.765 ± .047	.915 ± .029	**.948 ± .023**	.928 ± .027	**.948 ± .023**	.947† ± .024
dry-bean	.936 ± .027	.934 ± .028	.938 ± .027	**.940 ± .027**	.250 ± .141	.936 ± .027	.932 ± .028	.940† ± .027	.938 ± .027	.938 ± .027	.938 ± .073
nursery	.850 ± .079	.715 ± .139	.868 ± .105	.997 ± .006	.249 ± .194	.876 ± .065	.743 ± .156	**.997 ± .006**	.993 ± .010	**.997 ± .006**	.996 ± .007
hand-digits	.940 ± .026	.988 ± .011	.992 ± .009	.992 ± .009	.182 ± .112	.944 ± .025	.989 ± .011	**.995 ± .007**	.992 ± .009	**.995 ± .007**	.994 ± .008
isolet	.952 ± .022	.856 ± .042	.954 ± .022	.949 ± .024	.036 ± .034	.951 ± .022	.882 ± .038	**.960 ± .020**	.951 ± .023	**.960 ± .020**	.960† ± .021
wine-quality	.306 ± .124	.341 ± .102	.306 ± .130	.329 ± .121	.200 ± .166	.313 ± .121	**.423 ± .102**	.354 ± .127	.319 ± .122	**.423 ± .102**	.349 ± .154
satellite	.807 ± .079	.870 ± .048	.863 ± .057	.882 ± .050	.257 ± .143	.806 ± .077	.874 ± .047	.882 ± .052	.877 ± .053	.882 ± .052	**.888 ± .048**
digits	.963 ± .020	.961 ± .021	.973 ± .017	.972 ± .018	.100 ± .089	.962 ± .020	.962 ± .021	**.977 ± .016**	.974 ± .017	**.977 ± .016**	.976† ± .016
page-block	.713 ± .135	.703 ± .158	.666 ± .168	.808 ± .100	.537 ± .214	.758 ± .126	.716 ± .152	.799 ± .100	.850 ± .075	.850 ± .075	**.903 ± .038**
waveform-v1	.867 ± .038	.813 ± .048	.858 ± .039	.844 ± .037	.334 ± .241	.866 ± .040	.843 ± .049	.861 ± .056	.844 ± .038	.866 ± .040	**.871 ± .101**
spambase	.922 ± .030	.877 ± .042	.924 ± .034	.937 ± .025	.580 ± .247	.925 ± .028	.893 ± .038	.915 ± .039	.935 ± .028	.935 ± .028	**.944 ± .027**
academic-success	.652 ± .150	.555 ± .147	.631 ± .151	.668 ± .116	.334 ± .241	.652 ± .150	.561 ± .180	.669 ± .055	.656 ± .118	.669 ± .055	**.736 ± .096**
abalone	.255 ± .069	.231 ± .059	.249 ± .071	.255 ± .066	.108 ± .079	.255 ± .069	.248 ± .067	.090 ± .083	**.260 ± .066**	.090 ± .083	.233 ± .104
molecular	.903 ± .030	.817 ± .060	.934 ± .027	.921 ± .028	.340 ± .239	.934 ± .025	.872 ± .042	.938 ± .029	.933 ± .027	.938 ± .029	**.946 ± .039**
image-seg	.931 ± .038	.905 ± .050	.918 ± .048	.928 ± .044	.277 ± .154	.947 ± .032	.914 ± .045	.945 ± .030	.934 ± .041	.945 ± .030	**.951 ± .029**
obesity	.821 ± .064	.718 ± .080	.811 ± .057	.846 ± .052	.163 ± .119	**.931 ± .032**	.753 ± .079	.894 ± .041	.846 ± .052	**.931 ± .032**	.924 ± .034
semeion	.848 ± .090	.872 ± .080	.830 ± .106	.914 ± .056	.504 ± .291	.848 ± .090	.829 ± .105	.884 ± .074	.891 ± .068	.891 ± .068	**.922 ± .106**
yeast	.613 ± .090	.635 ± .083	.648 ± .083	.646 ± .077	.324 ± .185	.641 ± .129	.654 ± .097	.581 ± .227	.679 ± .095	.641 ± .129	**.756 ± .178**
cmc	.483 ± .076	.441 ± .068	.494 ± .069	.469 ± .076	.344 ± .222	.526 ± .077	.441 ± .068	.354 ± .229	.475 ± .074	.475 ± .074	**.547 ± .198**
hcv	.251 ± .049	.251 ± .048	.239 ± .055	.246 ± .051	.249 ± .194	.229 ± .045	**.261 ± .052**	.248 ± .189	.236 ± .053	.248 ± .189	.238 ± .188
phishing	.635 ± .196	.682 ± .146	.623 ± .215	.704 ± .153	.525 ± .199	.635 ± .196	.725 ± .109	**.806 ± .072**	.715 ± .146	**.806 ± .072**	.743 ± .173
mhr	.629 ± .113	.714 ± .087	.721 ± .117	.701 ± .099	.574 ± .217	.651 ± .153	**.810 ± .044**	.729 ± .070	.731 ± .095	**.810 ± .044**	.795† ± .110
german	.657 ± .128	.604 ± .150	.610 ± .196	.668 ± .106	.496 ± .291	.669 ± .100	.574 ± .216	.496 ± .291	.642 ± .128	.496 ± .291	**.769 ± .137**
tictactoe	.975 ± .021	.900 ± .054	.960 ± .030	.970 ± .024	.504 ± .291	.980 ± .018	.900 ± .054	**.985 ± .015**	.970 ± .024	**.985 ± .015**	.981 ± .044
mammographic	.788 ± .044	.760 ± .043	.760 ± .051	.792 ± .049	.739 ± .046	.800 ± .044	.769 ± .049	.787 ± .046	.784 ± .045	.784 ± .045	**.837 ± .059**
Average	.695 ± .261	.688 ± .250	.709 ± .265	.743 ± .252	.318 ± .259	.706 ± .262	.703 ± .247	.734 ± .285	.741 ± .251	.745 ± .270	**.771 ± .265**

Results. Table 1 reports the accuracy scores obtained by the classifiers resulting from each of the model selection strategies considered; each accuracy value is the average across the 1000 tests we have carried out for that dataset. Overall, TMS-All tends to obtain the (per-dataset) best results, and obtains the greatest number of best results. In cases when TMS-All does not obtain the best result, it still tends to obtain, for most datasets, results that are not statistically significantly different from the best-performing baseline. Our experiments clearly show that adopting MS strategies tailored to PPS yields a substantial performance advantage with respect to MS techniques that assume IID data. Applying TMS appears also preferable to training TSVMs; although TSVMs are tailored to transductive contexts, in our experiments they underperform when facing scenarios affected by PPS.

Figure 1 displays the (true) accuracy of the classifiers as a function of the amount of PPS, measured as the L1 distance between the vectors of class prevalence values of the training set and test bag; the results are obtained as within-bin averages, where a bin groups all the bags U_i affected by a similar amount of PPS, across all datasets. A clear pattern emerging from the plot is that most methods perform similarly for low levels of PPS, but there is a clear advantage in adopting TMS at higher levels of shift.

The same figure also shows (as a black dashed line) the performance of an oracle, i.e., an idealized method that always picks the best classifier for each bag U_i. The gap between the oracle and classifiers optimized via traditional MS is small at low shift levels (indicating that IMS works well in near-IID scenarios). However, as shift increases, traditional approaches degrade substantially, while the gap between the oracle and TMS remains narrow, showing that TMS performs well under such conditions.

4. Conclusions

The ability to perform robustly under distribution shift is a key requirement for trustworthy AI [15]. We have discussed transductive model selection (TMS), a new way of performing context-aware model

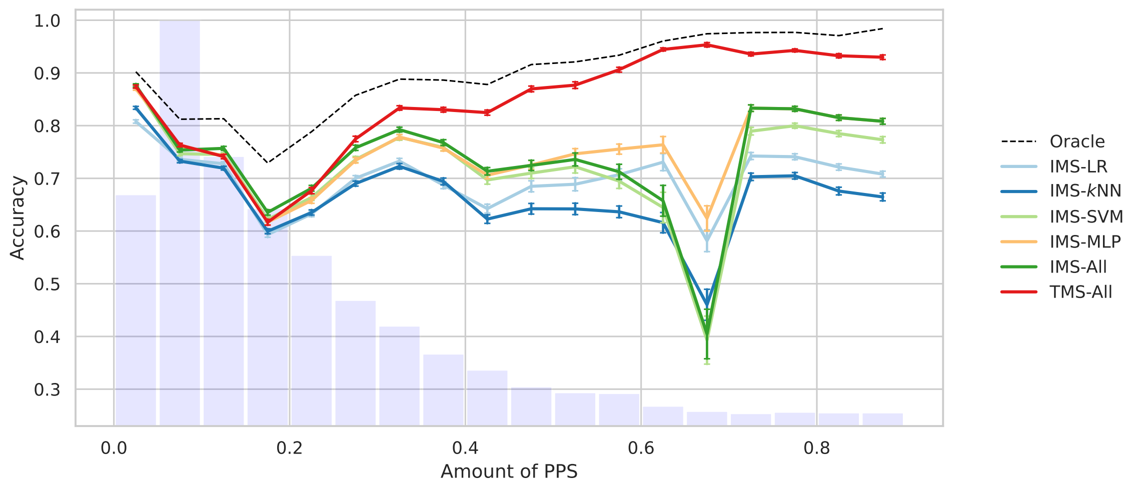

Figure 1: Accuracy of classifiers using different model selection strategies, as a function of the amount of PPS (measured in terms of the L1 distance between the training and test class prevalence distributions). Results are averaged across all test samples and datasets. The black dashed line represents the selection oracle. The curves are jagged because different datasets contribute at different PPS levels, as the maximum attainable PPS depends on the degree of the training set imbalance; the histogram in the background represents the density of experiments contributing at each level.

selection (i.e., hyperparameter optimisation) for classification applications. Essentially, TMS replaces traditional classifier accuracy *computation* on *training* data with classifier accuracy *estimation* on the finite set of *unlabelled* data that need to be classified at a certain point in time.

We have presented TMS experiments in a restricted setting, i.e., when the data are affected by prior probability shift, an important type of dataset shift that often affects anti-causal learning problems. Here, our experiments have shown that TMS boosts classification accuracy, i.e., bring about classifiers that outperform the classifiers whose hyperparameters have been optimised, as usual, by cross-validation on the labelled data. Note that TMS is not restricted to dealing with prior probability shift, and can also deal with other types of shift too (e.g., covariate shift). For this, one only needs to use, in place of the O-LEAP$_{\mathrm{KDEy}}$ method used in this paper (which is tailored to prior probability shift), a CAP method explicitly devised for the type of shift that the data suffer from.

TMS also holds promise for all applications that have a strictly transductive nature (i.e., in which *all* the unlabelled data to which the classifier needs to be applied are already known at training time), such as technology-assisted review (TAR – see e.g., [16]) for supporting e-discovery [17], online content moderation [18], or the production of systematic reviews [19]. Indeed, the next steps in our TMS research will include its application to these domains.

Transductive model selection not only improves robustness to distributional changes, but also strengthens the trustworthiness of AI systems by enabling more reliable classifier deployment than conventional “one-size-fits-all” inductive approaches.

Acknowledgments

LV’s work was supported by project “Italian Strengthening of ESFRI RI RESILIENCE” (ITSERR), funded by the European Union under the NextGenerationEU funding scheme (CUP B53C22001770006). AM’s and FS’s work was partially supported by project “Future Artificial Intelligence Research” (FAIR—CUP B53D22000980006), project “Quantification under Dataset Shift” (QuaDaSh—CUP B53D23026250001), and project “Strengthening the Italian RI for Social Mining and Big Data Analytics” (SoBigData.it—CUP B53C22001760006), all funded by the European Union under the NextGenerationEU funding scheme.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] S. Garg, S. Balakrishnan, Z. C. Lipton, B. Neyshabur, H. Sedghi, Leveraging unlabeled data to predict out-of-distribution performance, in: Proceedings of the 10th International Conference on Learning Representations (ICLR 2022), Virtual Event, 2022.

[2] D. Guillory, V. Shankar, S. Ebrahimi, T. Darrell, L. Schmidt, Predicting with confidence on unseen distributions, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2021), Montreal, CA, 2021, pp. 1134–1144.

[3] L. Volpi, A. Moreo, F. Sebastiani, LEAP: Linear equations for classifier accuracy prediction under prior probability shift, Machine Learning (2025). Forthcoming.

[4] A. Storkey, When training and test sets are different: Characterizing learning transfer, in: J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, N. D. Lawrence (Eds.), Dataset shift in machine learning, The MIT Press, Cambridge, US, 2009, pp. 3–28.

[5] M. Sugiyama, K. Müller, Model selection under covariate shift, in: Proceedings of the 15th International Conference on Artificial Neural Networks (ICANN 2005), Warsaw, PL, 2005, pp. 235–240. doi:10.1007/11550907_37.

[6] Z. C. Lipton, Y. Wang, A. J. Smola, Detecting and correcting for label shift with black box predictors, in: Proceedings of the 35th International Conference on Machine Learning (ICML 2018), Stockholm, SE, 2018, pp. 3128–3136.

[7] A. Ziegler, P. Czyż, Bayesian quantification with black-box estimators, Transactions on Machine Learning Research 2024 (2024).

[8] B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, J. M. Mooij, On causal and anticausal learning, in: Proceedings of the 29th International Conference on Machine Learning (ICML 2012), Edinburgh, UK, 2012.

[9] T. Fawcett, P. Flach, A response to Webb and Ting's 'On the application of ROC analysis to predict classification performance under varying class distributions', Machine Learning 58 (2005) 33–38.

[10] A. Esuli, A. Fabris, A. Moreo, F. Sebastiani, Learning to quantify, Springer Nature, Cham, CH, 2023.

[11] N. A. Smith, R. W. Tromble, Sampling uniformly from the unit simplex, Technical Report, Johns Hopkins University, 2004. https://www.cs.cmu.edu/~nasmith/papers/smith+tromble.tr04.pdf.

[12] A. Moreo, P. González, J. J. del Coz, Kernel density estimation for multiclass quantification, Machine Learning 114 (2025). doi:10.1007/s10994-024-06726-5.

[13] A. Gammerman, V. G. Vovk, V. Vapnik, Learning by transduction, in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998), Madison, US, 1998, pp. 148–155.

[14] T. Joachims, Transductive inference for text classification using support vector machines, in: Proceedings of the 16th International Conference on Machine Learning (ICML 1999), Bled, SL, 1999, pp. 200–209.

[15] R. Calegari, F. Giannotti, F. Pratesi, M. Milano, Introduction to the Special Issue on Trustworthy Artificial Intelligence, ACM Computing Surveys 56 (2024) 1–3. doi:10.1145/3649452.

[16] L. Gray, D. D. Lewis, J. Pickens, E. Yang, High-recall retrieval via technology-assisted review, in: Proceedings of the 47th ACM Conference on Research and Development in Information Retrieval (SIGIR 2024), Washington, US, 2024, pp. 2987–2988. doi:10.1145/3626772.3661376.

[17] D. W. Oard, W. Webber, Information retrieval for e-discovery, Foundations and Trends in Information Retrieval 7 (2013) 99–237. doi:10.1561/1500000025.

[18] E. Yang, D. D. Lewis, O. Frieder, TAR on social media: A framework for online content moderation, in: Proceedings of the 2nd International Conference on Design of Experimental Search & Information REtrieval Systems (DESIRES 2021), Padova, IT, 2021, pp. 147–155.

[19] G. Ferdinands, R. Schram, J. de Bruin, A. Bagheri, D. L. Oberski, L. Tummers, J. J. Teijema, R. van de

Schoot, Performance of active learning models for screening prioritization in systematic reviews: A simulation study into the average time to discover relevant records, Systematic Reviews 10 (2023). doi:10.1186/s13643-023-02257-7.