A method to enrich experimental datasets by means of numerical simulations in view of classification tasks

Damiano Lombardi, Fabien Raphel

A method to enrich experimental datasets by means of numerical simulations in view of classification tasks.

Damiano Lombardi*, Fabien Raphel†

March 31, 2021

Abstract

Classification tasks are frequent in many applications in science and engineering. A wide variety of statistical learning methods exists to deal with these problems. However, in many industrial applications, the number of available samples to train and construct a classifier is scarce and this has an impact on the classifications performances. In this work, we consider the case in which some a priori information on the system is available in form of a mathematical model. In particular, a set of numerical simulations of the system can be integrated to the experimental dataset. The main question we address is how to integrate them systematically in order to improve the classification performances. The method proposed is based on Nearest Neighbours and on the notion of Hausdorff distance between sets. Some theoretical results and several numerical studies are proposed.

Introduction

Classification tasks are frequent in many applications in science and engineering. The statistical learning methods which are proposed to deal with them rely on the fact that many examples (where the number of samples depends on the application under consideration) are available and can be exploited to uncover the underlying structure of the data and their separation in several classes. After the learning phase has been performed, a classifier is set up and can be used to infer to which class a new observed sample belongs to.

In many industrial applications the number of available samples is scarce, impacting the performances of the classification. A way to circumvent this limitation is to integrate to the available a posteriori information (provided by the available data) some a priori information (coming from experimental insight or theoretical knowledge) as proposed for instance in [1, 2, 3, 4].

*COMMEDIA, Inria Paris and LJLL, Sorbonne Université.
†COMMEDIA, Inria Paris and Notocord, part of Instem.

In this work we consider the case in which some a priori information is available in form of a mathematical model. Numerical simulations of several instances of the model can be computed and integrated to an available dataset in order to improve the classification performances. The main questions to be answered are: how many numerical simulation should we include, and which ones? Which information are needed in order to devise a systematical strategy? This work is devoted to the investigation of possible answers to these questions. This topic is closely related to two research fields in machine learning: domain adaptation and instance (or prototype) selection. The main goal of domain adaptation is to account for the discrepancies between target and test sets and propose ways to correct for them. An abundant literature on this subject is available, [5, 6, 7, 8]. The main difference with respect to the method proposed in the present work consists in the fact that in domain adaptation we often try to minimise a discrepancy between the datasets, whereas in the present work we focus on trying to improve a classification score. This is more similar, in the spirit, to the methods proposed in the field of instance selection. Different kind of algorithms have been proposed in this research field and can be divided into 4 different classes (commented and compared in the recent work [9]):

1. Incremental, such as Condensed Nearest Neighbors [10] and its variants [11, 12] or Instance-based learning [13]. These methods consist in building the training set by adding samples, chosen according different criteria.

2. Decremental such as Decremental Reduction Optimization Procedure [14, 15] or Hit Miss Network [16] consist in defining the training set by pruning samples from an available reservoir of potentially redundant (and corrupted) samples.

3. Batching such as Edited Nearest Neighbors [17], consists in testing whether each sample of the training set follows a removable criterion. All of the samples verifying this criteration are removed at once.

4. Fixed size such as Learning Vector Quantization [18] which consists in fixing a priori the size of the training set and selecting the samples to be used.

Recent studies have proposed in-between methods such as in [19]. These algorithms might have several drawbacks: in the methods in which we test one sample at a time and we decide if it has to be included or not into the training set, we might obtain a result which is sensitive to the order with which we test the samples. In some methods, the fitness function introduced to performed the selection is based on similarity criteria applied to the input features rather than the classification success rate, which might be suboptimal in some cases or it might depend upon hyperparameters which need to be tuned.

The main contributions of the present investigation are the following:

1. A systematic strategy can be set up, that enrich available training sets and improves the classification performance in a substantial way. The

only information which is exploited is a representative validation set, given even in form of samples or in form of a set of data and parameters of a reliable mathematical model describing the phenomenon.

2. The method which is proposed can be decomposed in two phases: an incremental one, in which we add to the training set samples taken from a reservoir of numerical simulations; a decremental one in which we prune samples to reduce redundancy and noise oversensitivity. We tried to reduce as much as possible the number of hyperparameters.

3. The obtained approach is not a generative one: it is not strictly needed to have an exhaustive training set distributed as the validation set; it is sufficient to add the most informative samples, in a sense that will be made more precise in the following, and that will be encoded in the fit functions used in the incremental and decremental phases.

The structure of the work is as follows. In Section 1 the method is proposed, and some properties are investigated from a theoretical standpoint. In Section 2 the discretisation is discussed, and in Section 3 some numerical test cases are presented to illustrate the approach.

1 The method

In this section, we detail the method proposed in the present work. The problem under investigation is a classification task, and, for sake of simplicity, we restrict to a binary classification. Four different sets of samples are introduced:

1. A *training set*, for which we know both the inputs (observations) and the output (labels), whose elements will be denoted by the superscript "tr". The training set is the main unknown of the problem, we wish to devise a way to construct it, starting from an available scarce (in the number of samples) training set.

2. A *validation* set, for which we know both the inputs (observations) and the output (labels), whose elements will be denoted by the superscript "v". This is the only source of information to construct the training set.

3. A *test* set, for which we know just the observation, whose elements will be denoted by the superscript "te".

4. A *reservoir* of numerical simulations of the systems, for which we know the observation and the label, to be used in order to construct or enrich the training set.

Several possible cases are met in realistic applications. First, we can be in a case in which we have an available experimental dataset covering all the possible meaningful instances of the problem under scrutiny, having however not so many samples (or not enough to have the wished performance on the

test set). We will call this a *complete validation* case. Second, we could be in a *incomplete validation* case, meaning that the experimental dataset to be used as training and validation covers only a subset of the possible instances (occurring in the test set). In both these situations, we would like to enrich the dataset by integrating elements of the reservoir in the training set. This is the simplest way to integrate some *a priori* information coming from mathematical modelling to the existing *a posteriori* information of the experimental data. We will consider here the cases of a perfect model (useful to validate certain aspect of the method) and the more realistic case in which the model is biased.

1.1 Context and notations

Let X be a random variable, representing the state of a system, for a population of individuals. A system configuration, identified by the realisation x, can belong to two classes, labelled $y = \{0, 1\}$. In an application, the system is observed through a measurement process and for a given observation $g \in \mathbb{R}^{n_g}$ (which in general results from the application of a non-linear function to x), we need to uncover whether the state belongs to the class $y = 0$ or $y = 1$.

The system observable for the population can be modelled by a random variable G defined on the probability space $(\Omega, \mathcal{A}, \mathbb{P})$, with $\Omega \subseteq \mathbb{R}^{n_g}$, $\mathcal{A}$ the σ-algebra of all the possible observables and $\mathbb{P}$ the probability measure. We denote $g^{(i)} \in \Omega$ a realisation of G and we assume that its probability density distribution, denoted $\rho(g)$, is a mixture of two densities. Let $\pi_0, \pi_1 \in (0, 1)$, such that $\pi_0 + \pi_1 = 1$. The probability density distribution reads:

$$\rho(g) = \pi_0 \rho_0(g) + \pi_1 \rho_1(g),$$

where $\rho_0(g), \rho_1(g)$ are the conditional probability density distributions for the classes 0 and 1 respectively, namely $\rho_{0,1}(g) = \rho(g|y = (0, 1))$.

In the following, the Lebesgue measure of a generic set A is denoted by $\mu_L(A)$. The classification success rate is based on a score function μ_s, which is a measure, introduced and described in [20], and that we recall for sake of completeness. The set of all the subsets in Ω is denoted by 2^Ω.

Definition: We define the score function μ_s as follows:

$$\mu_s : \begin{cases} 2^\Omega \times 2^\Omega \ \rightarrow \ \mathbb{R}^+ \\ (S_0, S_1) \ \mapsto \mu_s(S_0, S_1) \end{cases} \tag{1}$$

where we take:

$$\mu_s(S_0, S_1) = \pi_0 \int_{S_0} \rho_0^s dg + \pi_1 \int_{S_1} \rho_1^s dg, \tag{2}$$

with the given densities ρ_0^s, ρ_1^s, and the superscript "s" denote either the validation or the test set.

This score can be evaluated for all pairs of subsets S_0, S_1. It is related to the classification outcome when we compute it for the following pair:

$$\begin{cases} S_0 = \{g \in \mathbb{R}^{n_g}, \pi_0 \rho_0^{tr}(g) > \pi_1 \rho_1^{tr}(g)\} \\ S_1 = \{g \in \mathbb{R}^{n_g}, \pi_1 \rho_1^{tr}(g) > \pi_0 \rho_0^{tr}(g)\} \end{cases}, \tag{3}$$

where "tr" stands for the training set. As in [20], we make the following assumption:

$$\mu_L(S_2) = \mu_L\left(\{g \in \mathbb{R}^{n_g}, \pi_1 \rho_1^{tr}(g) = \pi_0 \rho_0^{tr}(g)\}\right) = 0.$$

Under the hypothesis that the set S_2 is a zero measure set, it follows that:

$$\rho_i^s = \rho_i^s \mathbf{1}_{S_i}, \forall i \implies \mu_s = 1.$$

Remark: The main goal is to enrich the training set aiming at improving the classification performance, which is quantified by the above introduced score. To this end, it is not needed to have the following strong outcome:

$$\pi_i \rho_i^{tr} = \pi_i \rho_i^v, \ i \in \{0, 1\}.$$

The propose approach is not a generative one seeking at generating samples distributed as the validation set, but samples which help improving the score. Henceforth, we could hopefully come up with a method which is less costly from a computational point of view.

1.2 Training set enrichment based on the Hausdorff distance: TSE-HD

We assume that Ω (defined in Section 1.1) is a measurable non-empty compact set of $\mathbb{R}^{n_g}$, and an observation of a system is $g \in \Omega \subset \mathbb{R}^{n_g}$.

At the beginning, the training set is given by the union of two known sets: $S_0^{(0)}$ and $S_1^{(0)}$: a sample of the training set is henceforth $g^{(tr)} \in S_0^{(0)} \cup S_1^{(0)}$. The goal is to progressively enrich the training set by making use of the samples in the reservoir of simulations. For the sake of simplicity, in this section, we make the hypothesis that the reservoir samples can cover Ω.

The information to be exploited comes from the knowledge of the validation set, either in form of samples or as a set of data and parameters of a mathematical model. This can be translated into two sets: $S_{0,1}^*$, with $S_1^* = \Omega \setminus S_0^*$, such that $S_0^* = \{g^{(v)} \in \Omega | y = 0\}$. These sets are optimal in the sense of the score function μ_v:

$$[S_0^*, S_1^*] = \arg \sup_{S_0, S_1 \subset \Omega} \mu_v. \tag{4}$$

In the following, we denote μ_* the score corresponding to these sets.

Let $n \in \mathbb{N}$ denotes the $n-$th step of the enrichment, we define $S_i^{(n)} \subseteq \Omega$ (for $i = 0$ or 1), the samples of the training set being $g^{(tr)} \in S_0^{(n)} \cup S_1^{(n)}$, as follows:

$$S_1^{(n)} = \Omega \setminus S_0^{(n)} \tag{5}$$

The score of the classification corresponding to these sets reads:

Definition:

$$\mu_v^{(n)} = \pi_0 \int_{S_0^{(n)}} \rho_0^v dg + \pi_1 \int_{S_1^{(n)}} \rho_1^v dg.$$

with:

$$\begin{cases} S_0^{(n)} = \{g \in \Omega, \pi_0 \rho_0^{(n)} > \pi_1 \rho_1^{(n)}\} \\ S_1^{(n)} = \{g \in \Omega, \pi_1 \rho_1^{(n)} > \pi_0 \rho_0^{(n)}\} \end{cases},$$

where $\rho_i^{(n)}$ is the pdf of the training set of class i and ρ_i^v is the pdf of the validation set of class i.

Starting from known sets $S_i^{(0)},\ i = 0, 1$, the goal is to transform them in order to converge to $S_i^*,\ i = 0, 1$, which maximizes the classification success rate. We construct a sequence which aims at increasing the cost function $\mu_v^{(n)}$, by observing that it is possible to make the sets $S_i^{(n)}$ to converge towards the optimal sets S_i^* by diminishing a suitable distance between these sets.

Let $\mathcal{B}(g, \varepsilon) \subset \Omega$ denote a ball of center g and radius $\varepsilon \geq 0$. The enrichment method is performed as follows. Let $S_{0,1}^{(n)}$ be the available set estimations.

1. Define $M^{(n)} = (S_0^* \cap S_1^{(n)}) \cup (S_1^* \cap S_0^{(n)})$
2. Solve the following problem[1]:

$$[g_{n+1}, \varepsilon_*] = \arg \sup_{g, \varepsilon \in \Omega} \left\{ \varepsilon \ | \mathcal{B}(g, \varepsilon) \subseteq M^{(n)} \right\}.$$

3. Let $\mathcal{B}_* = \mathcal{B}(g_{n+1}, \varepsilon)$. The update of the union of the intersections reads:

$$M^{(n+1)} = M^{(n)} \setminus \mathcal{B}_*,$$
$$S_0^{(n+1)} = \begin{cases} S_0^{(n)} \cup \mathcal{B}_* & \text{if } \mathcal{B}_* \subseteq S_0^* \cap S_1^{(n)}, \\ S_0^{(n)} \setminus \mathcal{B}_* & \text{if } \mathcal{B}_* \subseteq S_1^* \cap S_0^{(n)} \end{cases} \tag{6}$$

1.2.1 Analysis of the TSE-HD algorithm.

The convergence of the sets $S_{0,1}^{(n)}$ to the sets $S_{0,1}^*$ is studied. First, a Lemma is introduced, clarifying the meaning of the set $M^{(n)}$. Let $A \Delta B$ be the symmetric difference [21] between the sets A and B.

1. *For the set $M^{(n)}$, $\forall n \in \mathbb{N}$ it holds:*

$$M^{(n)} = S_0^* \Delta S_0^{(n)} = S_1^* \Delta S_1^{(n)},$$

The result of this Lemma, makes it possible to prove the following result (the proofs are presented in Supplementary material).

[1] On centrally symmetric sets, this would correspond to quantify the Bernstein widths of the set.

1. *Using the sequence of operations introduced above, almost surely, we have:*

$$\lim_{n\to+\infty} \mu_v^{(n)} = \mu_*.$$

Moreover, the gain on the score between two consecutive steps can easily be estimated. Its expression is given in the following result.

1. *Let $\mu_v^{(n)}$ be the score on the validation set at iteration $n \geq 0$. Then, $\forall n \in \mathbb{N}$, we have:*

$$\mu_v^{(n+1)} - \mu_v^{(n)} = \int_{\mathcal{B}_*} |\pi_1 \rho_1^v - \pi_0 \rho_0^v| dg \geq 0,$$

with $\mathcal{B}_* = \mathcal{B}(g_{n+1}, \epsilon_*)$ defined in the previous section. Moreover, the equality holds if and only if $\mu_L(\mathcal{B}_*) = 0$, where μ_L denotes the Lebesgue measure.

It follows that the gain is proportional to the total variation between ρ_0^v and ρ_1^v restricted to $\mathcal{B}_*$.

The result of the proposition states simply that, under the hypothesis that the system observable belongs to a compact set, and the set $S_{0,1}^*$ are known, the proposed iteration enrich the training set in such a way that the optimal classification score is retrieved. This algorithm shows some common properties with the algorithm detailed in [22]. In particular, the set sequence depends on the symmetric difference between the expected and the current set.

1.3 Reducing noise oversensitivity and bias induced errors: pruning.

At each stage of the TSE-HD algorithm, the samples of the training set contained in a selected ball $\mathcal{B}_*$ are added to the training set (either to $S_0^{(n+1)}$ or to $S_1^{(n+1)}$). As remarked in [15], a large number of noisy samples could lead to noise oversensitivity. Moreover, as the training set is enriched through numerical simulations, a bias could potentially pollute the classification results in regions where the samples of the validation set are scarce. To avoid these phenomena and to make the classification less prone to overfitting, a pruning phase is introduced, which consists in removing the samples which are not useful in improving the score.

Once TSE-HD is performed, the obtained training set consists in the pair $S^{(n,0)} = (S_0^{(n)}, S_1^{(n)})$. Since, in practice, we have a finite number of samples, these sets consist in a finite set of balls centred around a finite number of samples.

A stochastic algorithm is introduced. At the $k-$th iteration, a sample $g_k \in S_0^{(n)} \cup S_1^{(n)}$ of the training set is randomly selected. It can be considered as the center of a small ball $\mathcal{B}_k(g_k, \varepsilon_k)$ whose radius ε_k is such that the other samples do not belong to $\mathcal{B}_k$. The score is computed and the following action is taken:

$$S^{(n,k+1)} = \begin{cases} S^{(n,k)} \setminus \mathcal{B}_k \text{ if } \mu_v(S^{(n,k)} \setminus \mathcal{B}_k) \geq \mu_v(S^{(n,k)}) \\ S^{(n,k)} \text{ otherwise} \end{cases} \tag{7}$$

Remark that, by construction, at the end of the pruning step the score is at least as good as the beginning of the pruning step, and in some cases an improvement is obtained.

1.4 On realistic scenarios

In many applications different concerns may arise, such as the possible bias on the mathematical model (and then the database) [23, 24] and the incomplete validation case. We recall that in the present work we consider incomplete a validation set which does not cover the whole observable space Ω. In this section, a set of results are proposed to deal with these two cases.

1.4.1 Biased database

In general, the database obtained through a collection of experiments and/or simulations may have a bias. Let S_i^{te}, ($i = 0$ or 1) denote the test set which is supposed to cover Ω, *i.e.* $S_0^{te} \cup S_1^{te} = \Omega$:

$$\begin{cases} S_0^{(te)} = \{g \in \mathbb{R}^{n_g} | \pi_0 \rho_0^* > \pi_1 \rho_1^*\} \\ S_1^{(te)} = \{g \in \mathbb{R}^{n_g} | \pi_1 \rho_1^* > \pi_0 \rho_0^*\} \end{cases} . \tag{8}$$

The samples from these sets are samples drawn from the true underlying densities. The sets identified by using the densities of the model are:

$$\begin{cases} S_0^{(m)} = \{g \in \mathbb{R}^{n_g} | \pi_0 \rho_0^m > \pi_1 \rho_1^m\} \\ S_1^{(m)} = \{g \in \mathbb{R}^{n_g} | \pi_1 \rho_1^m > \pi_0 \rho_0^m\} \end{cases} . \tag{9}$$

The densities $\rho_{0,1}^m$ are in general different from the true ones. This is due to the model bias, which is such that the difference in the model state is propagated in the model observable g and hence in the density ρ^m. This, in turn, affects the sets $S_{0,1}^{(m)}$.

We recall that the sets satisfy:

$$\begin{cases} S_0^{te,m} \cup S_1^{te,m} = \Omega \\ S_0^{te,m} \cap S_1^{te,m} = \emptyset \end{cases} ,$$

We define the biased sets as follows:

$$\begin{cases} b_0 = S_0^m \cap S_1^{te} \\ b_1 = S_1^m \cap S_0^{te} \end{cases} .$$

The bias sets $b_{0,1}$ are quantifying, in a sense which is pertinent for the binary classification, the effect of the model bias.

2. *Let the sets $S_{0,1}^{te,m}$ be defined as in Eq.*(8)-(9)*. The following equalities hold:*

$$\begin{cases} S_0^m = (S_0^{te} \cup b_0) \setminus b_1 \\ S_1^m = (S_1^{te} \cup b_1) \setminus b_0 \end{cases}$$

.

The result of the Lemma makes it possible to prove the following result on the classification score of the test set:

2. *Let the hypothesis of Lemma 2 hold. Let*

$$\mu_b = \mu_{te}(S_0^m, S_1^m) = \int_{S_0^m} \pi_0 \rho_0^{te} dg + \int_{S_1^m} \pi_1 \rho_1^{te} dg, \tag{10}$$

be the score of the classification of the test set when the training set is defined by the model. The maximal score is represented by:

$$\mu_* = \mu_{te}(S_0^{te}, S_1^{te}). \tag{11}$$

It holds:

$$0 \leq \mu_b \leq \mu_*,$$

and, moreover:

$$\begin{cases} \mu_b = \mu_* \iff \mu_L(b_i) = 0, \text{ for } i \in \{0,1\} \\ \mu_b = 0 \iff S_i^m = S_j^{te} \text{ and } \rho_j^{te} = \rho_j^{te} \mathbf{1}_{\{S_j^{te}\}}, \text{ for } i,j \in \{0,1\},\ i \neq j \end{cases}.$$

Remark: In the case where $S_i^m = \emptyset$, we have $\mu_b = \int_\Omega \pi_j \rho_j^{te} dg$, $i \neq j$. It is straightforward to observe that in the case where there is no bias, we have the equality. In practice, we do not know S_j^{te}. It means that, if we only train with the model (database) we will compute the score over S_j^m.

1.4.2 The Validation set partially covers the set of possible outcomes.

In several situations it is possible to assess whether the validation set covers all the possible scenarios that could occur in the test set (even prior of receiving the test set). This is possible in particular when there is an underlying parametrisation of the system at hand, namely when the scenarios of interest are associated to values of data and parameters that characterise the solution of the models describing the phenomenon.

When the validation set partially covers Ω (incomplete validation set) we can show that the score on the test set (which is supposed to cover Ω) is lower than the score obtained with a validation set covering Ω (see Proposition 3).

3. *Let, $S_0^s \cup S_1^s = \Omega$ such that $S_0^s \cap S_1^s = \emptyset$ (for $s = te$ or v). Then,*

$$S_1^{te} \setminus S_1^v = S_0^v \setminus S_0^{te}.$$

3. *We denote $S_j^s = \{g | \pi_j \rho_j^s > \pi_k \rho_k^s\}$ $(k \neq j)$, where $s = te$ (test set) or v (validation set). We denote μ_{te}^c (resp. μ_{te}^p) the test set score obtained with a complete (resp. incomplete) validation set. By complete, we assume that the distribution of ρ_j^{te} and ρ_j^v are the same. Then,*

$$\mu_{te}^p \leq \mu_{te}^c.$$

In this scenario, we cannot use generative adversarial networks (GANs) [25] to enrich the training set in regions which are not covered by the validation set. This is due to the fact that the discriminator has no information on the region where there are no validation samples.

To enrich the training set, we propose first to enrich the validation set by adding to it samples extracted from the reservoir such that the enriched validation set covers all the possible meaningful scenarios.

If some information on the model bias is available (a statistics on the model bias), we proceed as follows. Let the bias in the observation be a random variable G_b, whose realisations are denoted by $g_b \in \mathbb{R}^{n_g}$. A sample of the reservoir is randomly picked in the region which is not covered by the validation set, whose observation is an element $g^{(r)} \in \mathbb{R}^{n_g}$. Then, a sample to be added to the validation set is:

$$g^{(v)} = g^{(r)} - g_b, \tag{12}$$

and the associated label is $y^{(v)} = y^{(r)}$.

2 Discretisation of the method.

When the enrichment method proposed in the previous section has to be applied to realistic cases, we need to account for the fact that the only available quantity is a set of labeled samples, which can be divided into training and validation sets. The method needs to be discretised in order to be practically implemented. Several elements need to be detailed. The first one is the estimation of the score function. Its computation requires a density estimation.

2.1 Density estimation in high-dimension.

To estimate the score by using a Monte Carlo method, we need to estimate a density in correspondence to a sample, namely the value $\rho(g) \in \mathbb{R}^+$. This task may be cumbersome due to the high-dimensionality of the space. Several methods of non-parametric density estimation are proposed in the literature. For the present work we consider as a starting point the k–nearest neighbors (KNN) estimation. In the KNN method, a tree-based algorithm subdivides the samples set into overlapping balls, each containing a fix number of samples, say $k \in \mathbb{N}^*$ on a total number of $N \in \mathbb{N}^*$ samples. The density is usually estimated by making the assumption that the density is roughly constant in a ball, leading to:

$$\rho(g^{(i)}) \approx \frac{k/N}{\mathrm{vol}(\mathcal{B}_i)}, \tag{13}$$

where $\mathcal{B}_i = \mathcal{B}(g^i, \varepsilon_i)$ and $\mathrm{vol}(\mathcal{B}_i)$ is its volume, computed according to the metric chosen to select the neighbors. We will denote the ℓ^p distance between two elements (g_1, g_2) as $\|g_1 - g_2\|_{\ell^{p,n_g}}$.

Remark: Following [26], if we want to classify a given sample g_* by using the Bayes rules, assuming $\mathbb{P}(y=0) = \mathbb{P}(y=1)$ and $N_0 = N_1 = N$, we will obtain the following result.

Let:

$$g_0^{(tr)} = \arg \inf_{g \in S_0^{(tr)}} \|g_* - g\|_{\ell^{p,ng}},$$
$$g_1^{(tr)} = \arg \inf_{g \in S_1^{(tr)}} \|g_* - g\|_{\ell^{p,ng}}.$$

Furthermore, let $\varepsilon_0, \varepsilon_1$ be the radius of the balls centred around $g_0^{(tr)}, g_1^{(tr)}$ respectively. The *a posteriori* probability reads:

$$\mathbb{P}(y=0|g_*) = \frac{\varepsilon_1^{n_g}}{\varepsilon_1^{n_g} + \varepsilon_0^{n_g}}.$$

This means that the classification outcome only depends on the distance between the closest points in each class in the training set and their respective k^{th} nearest neighbor. Figure 1 shows an example in which, by making use of this approach we wrongly classify a validation point. As the computed radius is lower for class 1 the validation point is labeled 1 instead of 0.

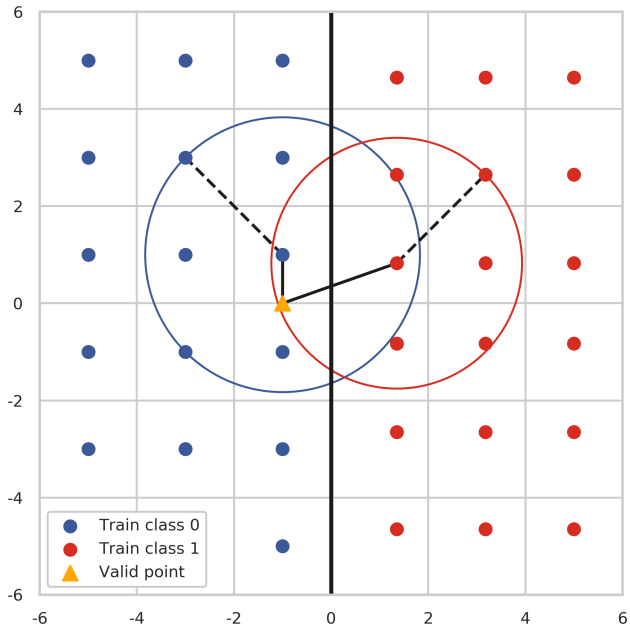

Figure 1: Section 2.1: Example of a wrongly classified point query point.

The issue shown in Fig.1 is mainly due to the assumption that the density is constant in the ball. We propose of replacing it by an approximation based on

Gaussian radial basis functions (RBFs). Let us introduce $\omega_i \in \mathbb{R},\ i = 1, \dots, k$; moreover, let the elements in a ball be $g^{(i)} \in \mathbb{R}^{n_g},\ i = 1, \dots, k$ and $\varepsilon_i > 0$ be the radius of the balls the samples $g^{(i)}$ are the center of. The density in a ball is expressed as:

$$\rho(g) \approx \sum_{i=0}^{k} \omega_i e^{-\frac{\|g-g^{(i)}\|_{\ell 2}^2}{2\varepsilon_i^2}},$$

Let $\overline{\rho}_i$ denote the density at the sample $g^{(i)}$ obtained by the classical KNN approximation. The weights ω_i are computed as the result of the following optimisation problem:

$$\rho_{app}(g) = \sum_{i=0}^{k} \omega_i e^{-\frac{\|g-g^{(i)}\|_{\ell 2}^2}{2\varepsilon_i^2}},$$

$$\mathcal{L}(\omega, \lambda) = \frac{1}{2} \sum_{i=1}^{k} |\omega_i - \overline{\rho}_i|^2 + \lambda \left(\frac{k}{N} - \int_{\mathcal{B}} \rho_{app}\ dg \right),$$

$$(\omega_*, \lambda_*) = \arg \inf_{\omega} \sup_{\lambda} \mathcal{L}(\omega, \lambda).$$

The interpretation is simple: the weights are close to the classical KNN estimated density (the Gaussian kernel being equal to one when evaluated at the sample), and when integrated on the ball, the approximation of the density retrieves the expected value of the mass in the ball. Let:

$$I_i = \int_{\mathcal{B}} e^{-\frac{\|g-g^{(i)}\|_{\ell 2}^2}{2\varepsilon_i^2}} dg.$$

The solution reads:

$$\omega_i^* = \overline{\rho}(g^{(i)}) + I_i \frac{k/N - \sum_{j=0}^{k} \overline{\rho}(g^{(j)}) I_j}{\sum_{j=0}^{k} I_j^2}.$$

The following Example aims at illustrating the effect of the above introduced approximation on a classification task.

Let $\Omega = [-5, 5]^2$ be the domain, and $g = (g_0, g_1) \in \Omega$. We define the two classes as follows:

$$y = \begin{cases} 0, & g_0 > 0 \\ 1, & g_0 \leq 0 \end{cases} \tag{14}$$

The sample size for the training set is $N_{0,1} = 18$. For each class the training set is uniformly distributed but with a different density (the density is higher for the class 1 as shown in Figure 1). The validation set is generated using a regular square mesh of Ω (with steps $\Delta g_0 = \Delta g_1 = 0.1$) where each node is a sample (it results in a validation sample size of $N_{0,1}^{te} = 5000$ for each class).

Figure 2 shows the result when the density is estimated via the classical KNN method and with the proposed Gaussian kernel correction. In this test,

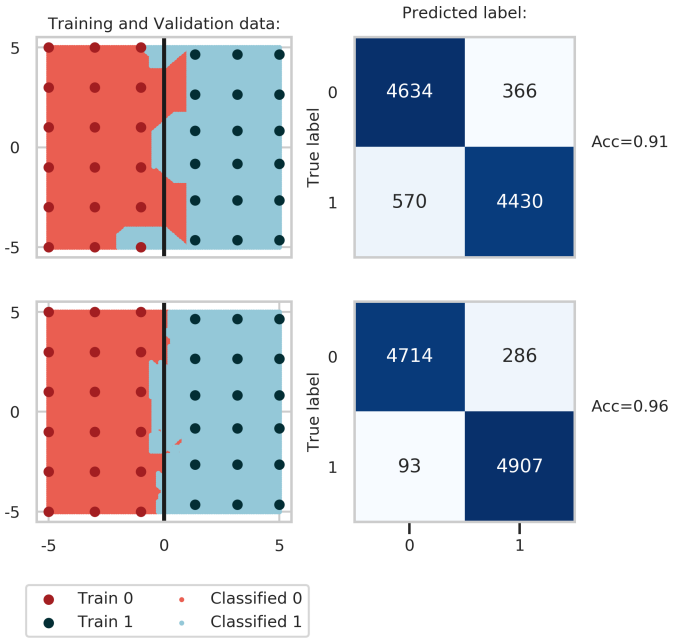

Figure 2: Section 2.1: Comparison of the two methods in a binary classification example. Number of neighbors: 5. Upper: usual KNN method. Lower: RBF based approximation. Left: training and validation sets. Right: corresponding confusion matrices.

the accuracy is significantly increased using the proposed technique (we pass from 0.86 to 0.96).

2.2 Computing the Hausdorff distance of sets.

One of the key steps of the proposed method is the approximation of the Hausdorff distance and the largest ball contained in the set $M^{(n)}$. Given the sets $S^n_{0,1}$, we can identify the $N_M \in \mathbb{N}^*$ samples, belonging to the validation set, which are in $M^{(n)} = \left(S_0^{(n)} \cap S_1^*\right) \cup \left(S_1^{(n)} \cap S_0^*\right)$. We denote $I_M^{(n)} \in \mathbb{N}$ the indices of these samples: $I_M^{(n)} = \left\{i \in 1, \ldots, N_v \text{ such that } g^{(i)} \in M^{(n)}\right\}$. The pairwise distance between every element of $M^{(n)}$ is computed, and the pair of elements maximising the distance is chosen:

$$i_*, j_* = \arg \max_{i,j \in I_M^{(n)}} \|g^{(i)} - g^{(j)}\|_{\ell^{p,n_g}}.$$

We then consider the segment relying the samples $g^{(i_*)}$ and $g^{(j_*)}$. The elements of this are characterise by the following expression. Let $\alpha \in [0, 1]$ and the

points: $g(\alpha) = (1-\alpha)g^{(i_*)} + \alpha g^{(j_*)}$. If the centre of the balls is chosen among the points of the segment, the problem reduces to finding α such that the radius of the ball inscribed in $M^{(n)}$ is the largest:

$$\alpha_* = \arg \sup_{\alpha \in [0,1]} \varepsilon,$$
$$\mathcal{B}(g(\alpha), \varepsilon) \subseteq M^{(n)}.$$

This problem is solved numerically by extensive search: the segment is discretised by considering a number of points on it, where the evaluation of the ball radius is performed.

Remark: During the enrichment process, it might happen that there are no elements in the reservoir belonging to the ball chosen to reduce the Hausdorff distance between the sets. We propose to add to the training set the center of the ball, labeled as the closest sample belonging to the validation set.

2.3 Summary of the method.

The overall method is summarised hereafter. Two validation sets are given, namely $S^*_{0,1} \subset \Omega$, in the form of sets of validation samples $g^{(v)}$. At the beginning of the procedure, we have two training sets $S^{(0)}_{0,1} \subset \Omega$, given in form of sets of samples $g^{(0)}$. At the beginning of a generic iteration of the method, say n, we have two training sets $S^{(n)}_{0,1}$.

1. *Evaluate the intersections between the validation sets and the current training sets:* $M^{(n)} = (S^*_0 \cap S^{(n)}_1) \cup (S^*_1 \cap S^{(n)}_0)$. To do so:
   (a) Evaluate the densities $\rho^{(n)}_{0,1}$ in the validation sample points $g^{(v)}$ by using the method described in Section 2.1.
   (b) Perform a Bayesian classification providing the labels y.
   (c) Compare the labels with the true validation labels y^*.
   (d) If $y \neq y^*$ then $g^{(v)} \in M^{(n)}$.
2. We compute an approximation of the Hausdorff distance, by evaluating the maximum of the distance between the well classified validation samples and the wrongly classified ones, that belong to $M^{(n)}$.
3. We compute the largest ball that is contained in $M^{(n)}$, by following the steps presented in Section 2.2.
4. We compute $S^{(n+1)}_{0,1}$ by adding to them the elements of the reservoir which are contained in the largest ball computed at the previous step, following Eq.(6).

3 Numerical experiments.

In this section, several numerical experiments are proposed to illustrate the enrichment method.

3.1 Two dimensional cases

A two dimensional application is performed on three study cases for which we consider $\Omega = [0, 1]^2$. For each study case, we randomly generated 2000 samples following a uniform law over Ω. The first half is gathered into the validation set whereas the second half is gathered into the test set. Figure 3 shows the validation set for each study case. The color corresponds to the label and the black line corresponds to the true delimitation of the two classes.

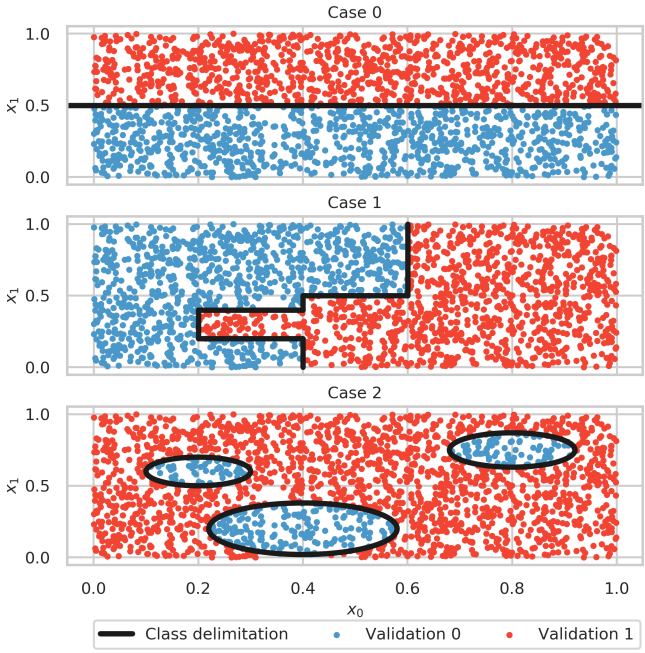

Figure 3: Section 3.1: Study cases.

The same random uniform process was performed to construct the initial training set (of size 20) and the reservoir of simulation (of size 1000). In this study we assume that the database is unbiased. The number of nearest neighbors is set to $k = 5$.

Figure 4 shows the constructed training set samples for each study case.

Two main points are highlighted by this figure:

- The whole initial database is not a must-have, only a small fraction of it is actually useful in view of improving the classification score.

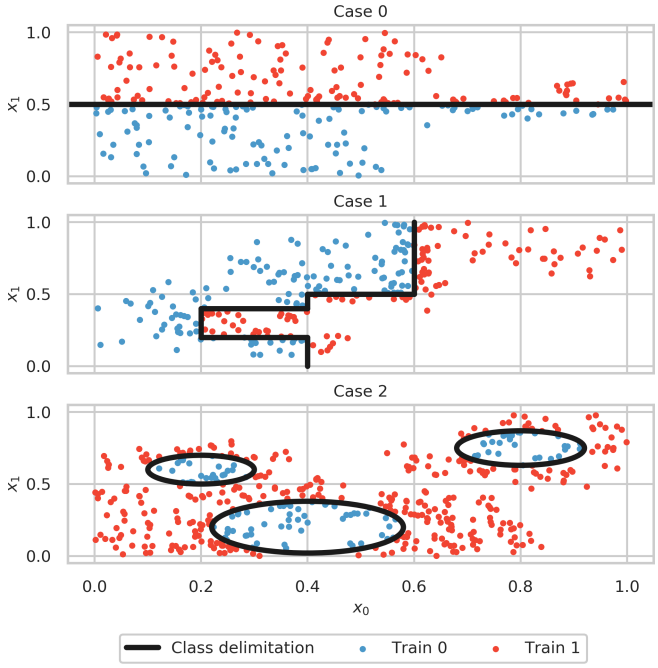

Figure 4: Section 3.1: Constructed training sets.

- The selected samples to construct the training set are mainly closed to the class delimitation.

Figure 5 shows the scores for the validation and test sets for each study case. As the algorithm is performed on the validation set, the score on the validation set is higher than the one on the test set (and its standard deviation smaller). Despite this slight overfitting, the constructed training set ensures a score higher than 0.96 on the test set for these three study cases.

3.2 A model in electro-physiology of cells.

This part is devoted to an example in electro-physiology. The observed model output, called action potential (AP) is the potential difference across the cell membrane. This is influenced by the value of several parameters which represent the conductances of some of the ion channels of the cell. The model we consider is called Minimal Ventricular (MV), presented in [27]; it is a system of parametric ordinary differential equations. We focus on three classification problems: given the model output determine if the conductances of sodium, calcium and potassium are above or below a certain threshold.

The dataset is synthetic and the numerical method used to approximate the model solution is a third order Backward Differentiation Formula (BDF3) with a time-step $\Delta t = 0.1$ms. A periodic source term in the equation is repeated

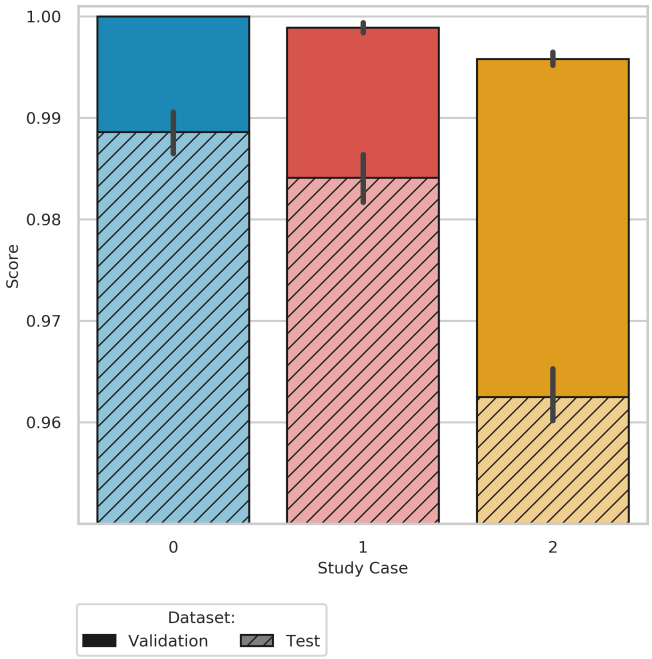

Figure 5: Section 3.1: Score obtained for the training and validation set for each study case.

every 1200ms and its parametrisation is given in Table 1.

Duration (ms)	Amplitude (pA/pF)
4.0	0.1

Table 1: Section 3.2: Stimuli parameters.

By starting from the third stimulation the system reaches periodicity (the ℓ^2 norm of the difference between two consecutive periods varies by less than 10^{-3}) we decided to only store the third period for this study.

A total of $n_s = 2420$ signal were generated with random triplets conductances (for sodium, calcium and potassium) following a uniform law over $[0.6, 1]^3$. It follows that for a realization $x = [x_{sodium}, x_{calcium}, x_{potassium}]$, the component x_i means that channel i is blocked at $100 * (1 - x_c)\%$. We consider the control case (as a reference) for the realization $x = [1, 1, 1]$ which leads to 100% of activity for each channel.

For each component c of a realization x, the labels y_c are given by:

$$y_c = \begin{cases} 0 \text{ if } x_c < 0.8 \text{ (”blocked”)} \\ 1 \text{ otherwise (”not blocked”)} \end{cases} . \tag{15}$$

The value 0.8 corresponds to the conductance threshold for the classification task described at the beginning of this section.

As we have three parameters, we divided the problem into three classification tasks: sodium, calcium and potassium conductances classification. An example of AP signals at control case ($x = [1, 1, 1]$) and in random case is shown in Figure 11 of the Appendix.

3.2.1 Biased data

Different biased datasets were generated from these $n_s = 2420$ simulated APs. These biased signals were obtained by computing the Fourier transform and putting to zero the entries corresponding to the higher frequencies. We considered three different levels of bias (expressed in terms of energy) as presented in Table 2

Bias level	Relative ℓ^2 error norm
Low	0.020
Medium	0.035
High	0.065

Table 2: Section 3.2.1: Biased datasets.

An example of an AP signal with its different levels of bias is shown in Figure 6.

3.2.2 Dictionary entry computation

For each sample (AP signal), we consider $n_g = 24$ observable quantities. These correspond to pairs times and amplitudes in different phases of the AP signal. They are computed in the same way for each sample and are shown in Fig. 7.

We denote $g_i^{(j)}$ the i^{th} dictionary entry of the j^{th} AP signal. Considering the control case as a reference, we propose to consider the following translated dictionary entries:

$$g_i^{(j)} = g_i^{(j)} - g_i^{(ctrl)}, \forall i, j.$$

It follows that, in the control case, we have $g_i^{(ctrl)} = 0, \forall i = 1, \ldots, n_g$. All the samples were then transformed in such a way that the compact domain Ω is the hypercube of dimension $n_g = 24$, side 1 and centered at $c = (\frac{1}{2}, \ldots, \frac{1}{2}) \in \mathbb{R}^{n_g}$.

3.2.3 Datasets preprocessing

Two study cases are performed: in the first one, we assume that the validation set covers Ω whereas in the second one we consider an incomplete validation (the validation set covers only a subset of Ω). To do so, from the unbiased dataset, we randomly extract $n_v = 89$ from the $n_s = 2420$ signals in such a way that 84 of them have a sodium and calcium activity higher than 0.85. The 5

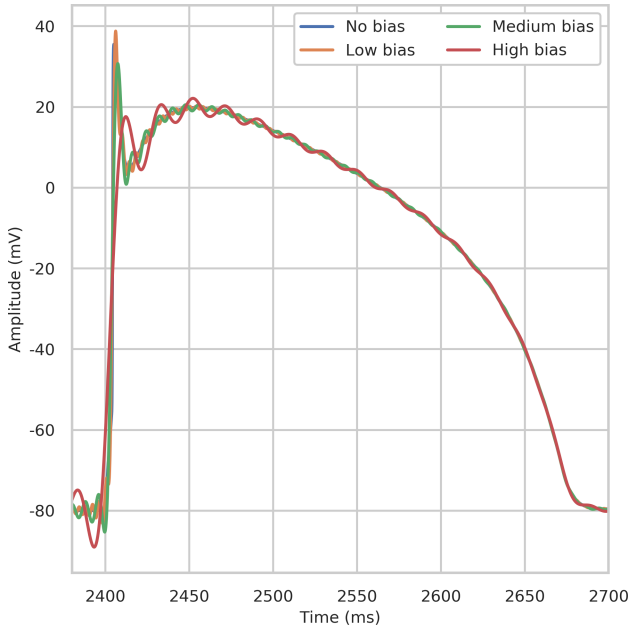

Figure 6: Section 3.2.1: Sample of an action potential signal generated by the MV model with its different levels of bias.

others are randomly chosen in such a way that at least one sample belongs to the other class (sodium and/or calcium conductance is lower than the threshold). Dataset's sizes are summarized in Table 3.

Validation set	n_t (test)	n_v (validation)	n_{tr} (initial training)	n_d (database)
Complete: Covers Ω	1000	400	20	1000
Incomplete: Partially covers Ω	1000	89	20	1000

Table 3: Section 3.2.3: Datasets sizes.

Test, validation and initial training sets are randomly extracted from the whole unbiased dataset ($n_s = 2420$). The database can be biased or unbiased depending on the study (chosen samples are the same, but with different biases). The random process is performed in such a way that a selected sample belongs to only one set and cannot be selected more than once. Figure 8 shows the densities of the variable x for the validation and test sets (for each class), in the sodium classification task.

As we can see, when the complete validation case is considered, the density of x is almost uniform over the whole domain of x (meaning that we have samples for almost all possible values of x). On the contrary, for the incomplete validation case (in the center) we clearly see that there are regions of the domain of x in which we do not have samples.

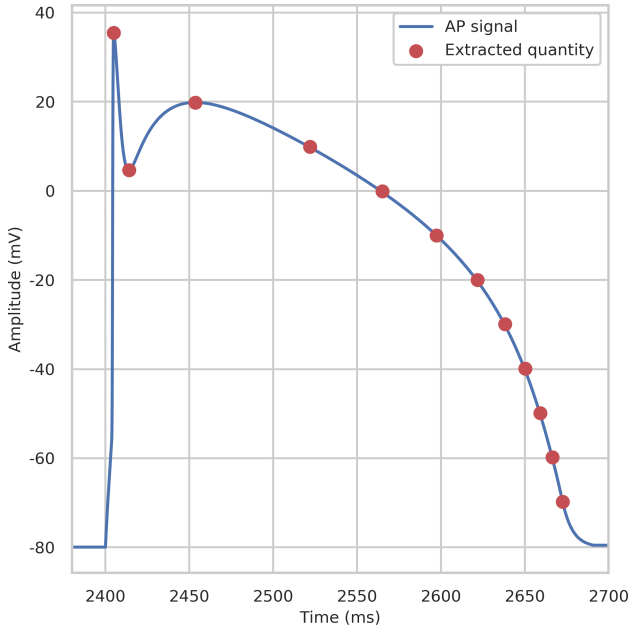

Figure 7: Section 3.2.2: Sample of an action potential signal generated by the MV model (control case: $x = [1, 1, 1]$) with the extracted quantities to generate the dictionary entries.

3.2.4 Computational results

All the following results were obtained using $k = 5$ nearest neighbors.

Comparison between complete and incomplete validation set

Figure 9 shows the scores obtained with a complete and incomplete validation set.

1. Complete validation set:
   (a) The validation score is higher than the test score because the optimization process is performed on the validation set.
   (b) The sodium conductance is easy to classify, whereas calcium conductance is the most difficult to infer. The fact that potassium and calcium conductances are more difficult to classify is due to the compensation effect between these two channels (see Figure 11), which is a known phenomenon in electrophysiology.
   (c) The scores are not significantly impacted by the bias as the proposed method naturally rejects it.

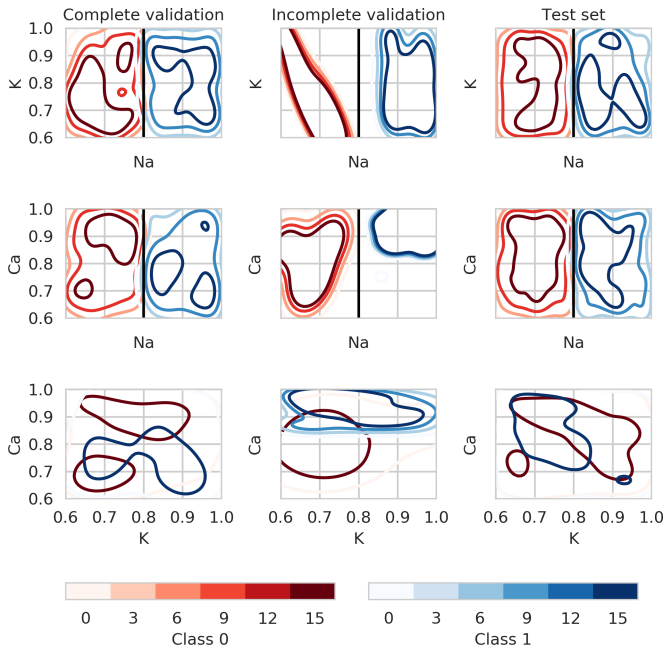

Figure 8: Section 3.2.3: Densities of validation and test sets for sodium classification. Black lines correspond to the class delimitation.

2. Incomplete validation set:

   (a) The validation score is higher than the test score because the optimization process is performed on the validation set.

   (b) The calcium conductance classification shows the lowest success rate whereas the potassium conductance classification shows the highest score. The fact that the potassium has the highest score is expected as no data were removed for this case. The scores obtained in the unbiased case are close to the expected scores: around 69% for the sodium, 75% for the potassium and 60% for the calcium (see Section D of the Appendix for more details). The bias does not highly affect the score except for the sodium in the highest bias case).

   (c) The bias is larger in the first part of the signal, as it can be seen in Fig. 6. This phase of the solution is known to be influenced by the sodium conductance. This explains why the score for the sodium classification is more impacted than the ones for calcium and potassium which show a more stable trend.

3. Complete vs Incomplete validation set:

   (a) The validation score is more stable and higher for the incomplete validation set. This is explained by the fact that we have less data

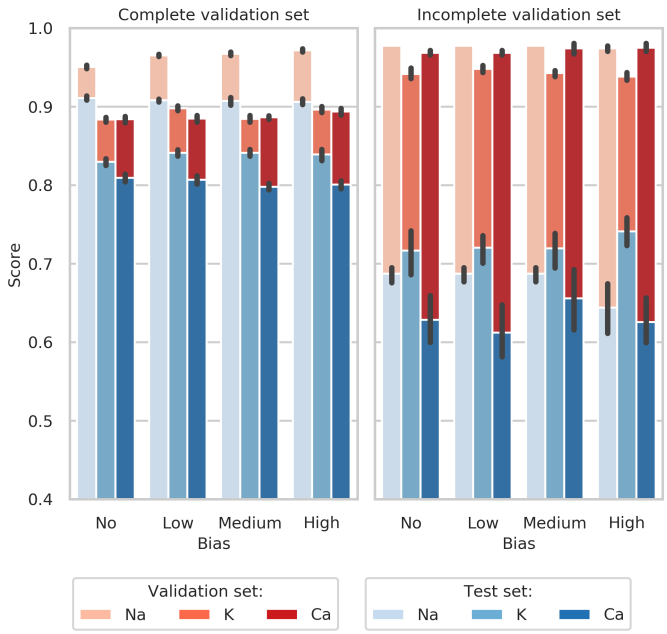

Figure 9: Section 3.2.4: Scores obtained with a complete and incomplete validation set.

in the validation set and aggregated in a smaller region, which eases the process.

(b) The test score is lower in the incomplete validation set case. This is because there are regions of Ω in which we do not have samples of the dataset. As we do not have information in these empty regions, the score is lower.

(c) For the same reasons as above, the variability on the test score is higher when the validation set is incomplete.

Database and validation set enrichment

As described in Section 1.4.2, once the training set enrichment process is performed on the incomplete validation set, we enrich the validation set with data from the database. In the case where we have a bias, we may exploit some statistical information on the bias to generate more pertinent labeled samples. We recall that we have 4 different study cases based on the database (see Section 3.2.1): without bias and with a low, medium and high level of bias. We assume that we know the a priori for the two classes: $\pi_0 = \pi_1 = \frac{1}{2}$. Then, we enriched the validation set in such a way the number of samples is each class is the same, with $n_v = 400$ (we added 311 samples). See Table 3.

Unbiased case In the unbiased case, we compute the dictionary entry mean and standard deviation for each class of the incomplete validation set. We denote $\widehat{\pi}_i$ the estimated *a priori*. Then, we randomly brows each sample of the database (for each class). While $n_v < 400$, if one of the entries is outside the corresponding (i.e same class) mean plus/minus the standard deviation, we add it to the validation set (and remove it from the database) if the following equation holds:

$$\min_i \widehat{\pi}_i^{(n+1)} > \min_i \widehat{\pi}_i^{(n)},$$

with $\widehat{\pi}_i^{(n+1)}$ the *a priori* computed considering the sample into the validation set and $\widehat{\pi}_i^{(n)}$ the *a priori* computed before considering the current sample into the validation set. In other words, it aims to consider the assumptions on the true *a priori* π_i described above.

Biased case For the biased case, we compute the average and standard deviation difference (in the dictionary entry space) between the incomplete validation set and the simulated data with the same parameters:

$$\begin{cases} b_m = \mathbb{E}\Big(D_{\theta_v} - V_{\theta_v}\Big) \\ b_s = \sqrt{\mathbb{E}\Big((D_{\theta_v} - V_{\theta_v})^2\Big)} \end{cases},$$

with $b_j \in \mathbb{R}^{n_g}$ the mean ($j = m$) or the standard deviation ($j = s$) and where V_{θ_v} is the incomplete validation set and D_{θ_v} is the simulated dataset obtained with θ_v as parameter entries of the simulated model. Then, from these statistics, for each sample of the database, we generate 4 ghosts samples following the approach described in Section 1.4.2. Here, we assume that the bias computed on the validation set is preserved on the empty region.

Results The results are shown in Figure 10.

1. The validation set (red and orange) vs test set (blue and green): we always obtain a higher score on the validation set.
2. The enrichment case (orange and green):
   (a) In the incomplete validation set case, the score on the test set is lower than the validation set.
   (b) The enrichment strategy implies a significantly higher score on the test set for sodium and calcium conductance classification.
   (c) This is not the case on the validation set, because we introduce some variability with the ghost and the correction.
   (d) The enrichment in the case where there is no bias (and no ghosts) induces scores closed to the complete validation set.

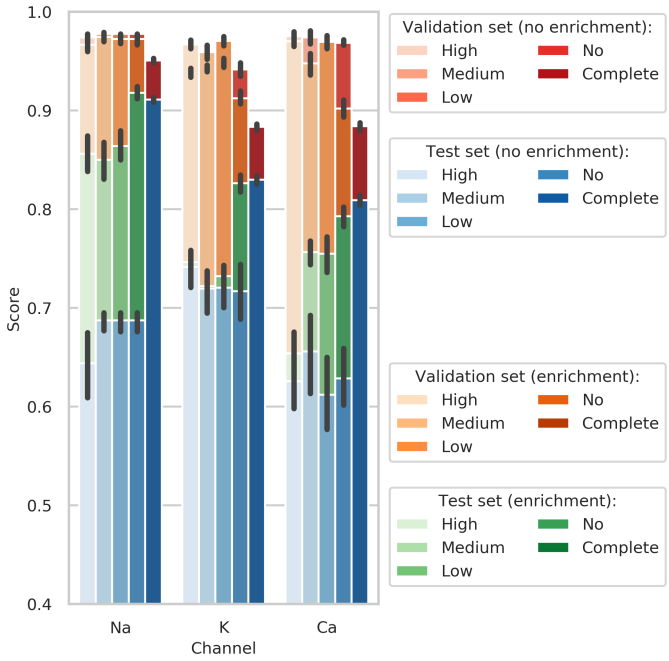

Figure 10: Section 3.2.4: Scores with incomplete/complete validation set and enriched validation set and database.

3. Conductance classification on the test set (blue and green):

   (a) The main score benefit is for the sodium conductance classification (from around 0.64 to around 0.85 depending on the bias). This is due to the fact that there is not a compensation effect between the sodium channel and the other channels (see Figure 11).

   (b) We also have a significant increase of the score for the calcium conductance classification (from about 0.62 to about 0.78).

4 Conclusions and perspectives

In the present work a method is proposed to enrich available experimental datasets by using numerical simulations in view of improving classification tasks performances. This is an example of potential interaction between statistical learning and mathematical modelling. The method is based on the probabilistic description of the observations of a phenomenon and a characterisation of the classification performances based on set distances. The main properties of the method have been investigated from a theoretical point of view and illustrated through some numerical experiments. The systematic construction and enrichment of the training set can have a significant impact on the classification score.

The proposed method performs a bias rejection to some extent, and, if statistical information on a model bias are available, these can be naturally integrated in the algorithm.

References

[1] Mendizabal, A., Fountoukidou, T., Hermann, J., Sznitman, R., & Cotin, S. (2018, September). A combined simulation and machine learning approach for image-based force classification during robotized intravitreal injections. In International conference on medical image computing and computer-assisted intervention (pp. 12-20). Springer, Cham.

[2] Müller, B., Hasman, A., & Blom, J. A. (1996). Building intelligent alarm systems by combining mathematical models and inductive machine learning techniques Part 2—Sensitivity analysis. International journal of bio-medical computing, 42(3), 165-179.

[3] Johnson, M. (2003). Classification of AE transients based on numerical simulations of composite laminates. Ndt & e International, 36(5), 319-329.

[4] Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., & Ryne, R. D. (2004). Combining field data and computer simulations for calibration and prediction. SIAM Journal on Scientific Computing, 26(2), 448-466.

[5] Wang, M., & Deng, W. (2018). Deep visual domain adaptation: A survey. Neurocomputing, 312, 135-153.

[6] Sun, S., Shi, H., & Wu, Y. (2015). A survey of multi-source domain adaptation. Information Fusion, 24, 84-92.

[7] Patel, V. M., Gopalan, R., Li, R., & Chellappa, R. (2015). Visual domain adaptation: A survey of recent advances. IEEE signal processing magazine, 32(3), 53-69.

[8] Zhang, K., Schölkopf, B., Muandet, K., & Wang, Z. (2013, May). Domain adaptation under target and conditional shift. In International Conference on Machine Learning (pp. 819-827). PMLR.

[9] Blachnik, M., & Kordos, M. (2020). Comparison of Instance Selection and Construction Methods with Various Classifiers. Applied Sciences, 10(11), 3933.

[10] Hart, P. (1968). The condensed nearest neighbor rule (corresp.). IEEE transactions on information theory, 14(3), 515-516.

[11] Ritter, G., Woodruff, H., Lowry, S., & Isenhour, T. (1975). An algorithm for a selective nearest neighbor decision rule (corresp.). IEEE Transactions on Information Theory, 21(6), 665-669.

[12] Vázquez, F., Sánchez, J. S., & Pla, F. (2005, June). A stochastic approach to Wilson's editing algorithm. In Iberian conference on pattern recognition and image analysis (pp. 35-42). Springer, Berlin, Heidelberg.

[13] Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. Machine learning, 6(1), 37-66.

[14] Wilson, D. R., & Martinez, T. R. (1997, July). Instance pruning techniques. In ICML (Vol. 97, No. 1997, pp. 400-411).

[15] Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. Machine learning, 38(3), 257-286.

[16] Marchiori, E. (2008). Hit miss networks with applications to instance selection.

[17] Tomek, I. (1976). AN EXPERIMENT WITH THE EDITED NEAREST-NIEGHBOR RULE.

[18] Nova, D., & Estévez, P. A. (2014). A review of learning vector quantization classifiers. Neural Computing and Applications, 25(3), 511-524.

[19] Cano, J. R., Herrera, F., & Lozano, M. (2005). Stratification for scaling up evolutionary prototype selection. Pattern Recognition Letters, 26(7), 953-963.

[20] Lombardi, D., & Raphel, F. (2019). A greedy dimension reduction method for classification problems.

[21] Givant, S., & Halmos, P. (2008). Introduction to Boolean algebras. Springer Science & Business Media.

[22] Binev, P., Cohen, A., Dahmen, W., & DeVore, R. (2014). Classification algorithms using adaptive partitioning. Annals of Statistics, 42(6), 2141-2163.

[23] Gu, M., & Anderson, K. (2018). Calibration of imperfect mathematical models by multiple sources of data with measurement bias. arXiv preprint arXiv:1810.11664.

[24] Tedeschi, L. O. (2006). Assessment of the adequacy of mathematical models. Agricultural systems, 89(2-3), 225-247.

[25] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint arXiv:1406.2661.

[26] Gweon, H., Schonlau, M., & Steiner, S. H. (2019). The k conditional nearest neighbor algorithm for classification and class probability estimation. PeerJ Computer Science, 5, e194.

[27] Bueno-Orovio, A., Cherry, E. M., & Fenton, F. H. (2008). Minimal model for human ventricular action potentials in tissue. Journal of theoretical biology, 253(3), 544-560.

A Proofs in Section 1.2.1

Lemma 1. For the set $M^{(n)}$, $\forall n \in \mathbb{N}$ it holds:

$$M^{(n)} = S_0^* \Delta S_0^{(n)} = S_1^* \Delta S_1^{(n)},$$

Proof of Lemma 1. By definition of the symmetric difference, we have:

$$S_0^* \Delta S_0^{(n)} = (S_0^* \setminus S_0^{(n)}) \cup (S_0^{(n)} \setminus S_0^*).$$

$$\iff$$

$$S_0^* \Delta S_0^{(n)} = (S_0^* \cap {S_0^{(n)}}^C) \cup (S_0^{(n)} \cap {S_0^*}^C),$$

where ${S_0^{(n)}}^C = \Omega \setminus S_0^{(n)}$ and ${S_0^*}^C = \Omega \setminus S_0^*$ are the complementary sets of $S_0^{(n)}$ and S_0^* respectively. It follows that:

$$S_0^* \Delta S_0^{(n)} = (S_0^* \cap S_1^{(n)}) \cup (S_0^{(n)} \cap S_1^*) = M^{(n)}.$$

The proof for $S_1^* \Delta S_1^{(n)}$ is similar. □

Proposition 1. Using the sequence of operations introduced in Section 1.2, almost surely, we have:

$$\lim_{n \to +\infty} \mu_v^{(n)} = \mu_*.$$

Proof of Proposition 1. By definition of S_j^* and $S_j^{(n)}$ (see Equation 5), we have:

$$(S_0^* \cap S_1^{(n)}) \cap (S_1^* \cap S_0^{(n)}) = \emptyset.$$

Then, $M^{(n)}$ is a disjoint union of two sets. This implies that:

$$\mu_L(M^{(n)}) = \mu_L(S_0^* \cap S_1^{(n)}) + \mu_L(S_1^* \cap S_0^{(n)}).$$

Remark that, by definition of the Lebesgue measure on a set and due to the compactness of the sets, we have the following inequalities:

$$0 \leq \mu_L(M^{(n)}) < +\infty.$$

It is straightforward to show that:

$$\mu_L(M^{(n)}) = 0 \iff \mu_v^{(n)} = \mu_* \text{ almost surely,}$$

Let assume that $\mu_L(M^{(n)}) > 0$. It follows that at least one of the following inequalities is satisfied:

$$\begin{cases} \mu_L(S_0^* \cap S_1^{(n)}) > 0 \\ \mu_L(S_1^* \cap S_0^{(n)}) > 0 \end{cases}.$$

Let S' be the set such that:

$$S' = \arg\max \Big(\mu_L(S_0^* \cap S_1^{(n)}), \mu_L(S_1^* \cap S_0^{(n)}) \Big).$$

We then have $\mu_L(S') > 0$. Therefore, $\exists g_{n+1} \in S'$ and $\varepsilon > 0$such that the ball $\mathcal{B}(g_{n+1}, \varepsilon) \subseteq S'$. By definition of $M^{(n)}$ (see Section 1.2), we have:

$$M^{(n+1)} = M^{(n)} \setminus \mathcal{B}.$$

As $\mathcal{B} \in S' \subseteq M^{(n)}$ and $\mu_L(\mathcal{B}) > 0$, we have:

$$0 \leq \mu_L(M^{(n+1)}) < \mu_L(M^{(n)}).$$

We have a sequence of measures which is strictly decreasing and bounded. Thus, this sequence converges to its minimum. Let assume that this minimum is $\delta > 0$. Then, it exists a non-empty ball such that the measure will decrease, which is impossible. It follows that:

$$\lim_{n \to +\infty} \mu_L(M^{(n)}) = 0.$$

Therefore,

$$\lim_{n \to +\infty} S_i^{(n)} = S_i^*,$$

almost everywhere for $i = 0$ or 1. Hence, almost surely, we have:

$$\lim_{n \to +\infty} \mu_v^{(n)} = \mu_*.$$

□

Corollary 1. Let $\mu_v^{(n)}$ be the score on the validation set at iteration $n \geq 0$. Then, $\forall n \in \mathbb{N}$, we have:

$$\mu_v^{(n+1)} - \mu_v^{(n)} = \int_{\mathcal{B}_*} |\pi_1 \rho_1^v - \pi_0 \rho_0^v| dg \geq 0,$$

with $\mathcal{B}_* = \mathcal{B}(g_{n+1}, \epsilon_*)$ defined in the previous section. Moreover, the equality holds if and only if $\mu_L(\mathcal{B}_*) = 0$, where μ_L denotes the Lebesgue measure.

Proof of Corollary 1. By definition, $\forall n \in \mathbb{N}$, we have:

$$\mu_v^{(n)} = \int_{S_0^{(n)}} \pi_0 \rho_0^v dg + \int_{S_1^{(n)}} \pi_1 \rho_1^v dg.$$

Then, at iteration $n+1$, we have:

$$\mu_v^{(n+1)} = \int_{S_0^{(n+1)}} \pi_0 \rho_0^v dg + \int_{S_1^{(n+1)}} \pi_1 \rho_1^v dg,$$

with:

$$S_0^{(n+1)} = \begin{cases} S_0^{(n)} \cup \mathcal{B}_* \text{ if } \mathcal{B}_* \subseteq S_0^* \cap S_1^{(n)} \\ S_0^{(n)} \setminus \mathcal{B}_* \text{ if } \mathcal{B}_* \subseteq S_1^* \cap S_0^{(n)} \end{cases}.$$

Let us consider the first scenario: $S_0^{(n+1)} = S_0^{(n)} \cup \mathcal{B}_*$. Then using the fact that the sets are disjoint, we have:

$$\mu_v^{(n+1)} = \int_{S_0^{(n)}} \pi_0 \rho_0^v dg + \int_{\mathcal{B}_*} \pi_0 \rho_0^v dg + \int_{S_1^{(n)}} \pi_1 \rho_1^v dg - \int_{\mathcal{B}_*} \pi_1 \rho_1^v dg,$$

which immediately yields to:

$$\mu_v^{(n+1)} - \mu_v^{(n)} = \int_{\mathcal{B}_*} (\pi_0 \rho_0^v - \pi_1 \rho_1^v) dg \geq 0.$$

Here, we assumed that $\mathcal{B}_* \subseteq S_0^* \cap S_1^{(n)}$. The inequality is given by the definition of S_0^*. On this set, we have: $\pi_0 \rho_0^v - \pi_1 \rho_1^v > 0$. The equality is then obtained if and only if $\mu_L(\mathcal{B}_*) = 0$. Considering the second scenario, we finally obtain:

$$\mu_v^{(n+1)} - \mu_v^{(n)} = \int_{\mathcal{B}_*} |\pi_0 \rho_0^v - \pi_1 \rho_1^v| dg \geq 0.$$

□

B Proofs in Section 1.4.1

Lemma 2. Let the sets $S_{0,1}^{te,m}$ be defined as in Eq.(8)-(9).The following equalities hold:

$$\begin{cases} S_0^m = (S_0^{te} \cup b_0) \setminus b_1 \\ S_1^m = (S_1^{te} \cup b_1) \setminus b_0 \end{cases}$$

.

Proof of Lemma 2. Let us focus on the first equality of the lemma (the proof for the second equality is similar). We have:

$$(S_0^{te} \cup b_0) \setminus b_1 = \left(S_0^{te} \setminus b_1\right) \cup \left(b_0 \setminus b_1\right)$$

As $b_1 \cap b_0 = \emptyset$ we have:

$$(S_0^{te} \cup b_0) \setminus b_1 = \left(S_0^{te} \setminus b_1\right) \cup b_0 = \left(S_0^{te} \setminus (S_1^m \cap S_0^{te})\right) \cup b_0.$$

$$\Longleftrightarrow$$

$$(S_0^{te} \cup b_0) \setminus b_1 = \Big(S_0^{te} \setminus S_1^m\Big) \cup b_0 = \Big(S_0^{te} \setminus S_1^m\Big) \cup \Big(S_0^m \cap S_1^{te}\Big)$$

$$\Longleftrightarrow$$

$$(S_0^{te} \cup b_0) \setminus b_1 = \Big(S_0^m \cap S_0^{te}\Big) \cup \Big(S_0^m \cap S_1^{te}\Big) = S_0^m \cap \Big(S_0^{te} \cup S_1^{te}\Big).$$

Since $S_0^{te} \cup S_1^{te} = \Omega$, we finally obtain:

$$(S_0^{te} \cup b_0) \setminus b_1 = S_0^m.$$

□

Proposition 2. Let the hypothesis of Lemma 2 hold. Let

$$\mu_b = \mu_{te}(S_0^m, S_1^m) = \int_{S_0^m} \pi_0 \rho_0^{te} dg + \int_{S_1^m} \pi_1 \rho_1^{te} dg, \tag{16}$$

be the score of the classification of the test set when the training set is defined by the model. The maximal score is represented by:

$$\mu_* = \mu_{te}(S_0^{te}, S_1^{te}). \tag{17}$$

It holds:

$$0 \leq \mu_b \leq \mu_*,$$

and, moreover:

$$\begin{cases} \mu_b = \mu_* \iff \mu_L(b_i) = 0, \text{ for } i \in \{0,1\} \\ \mu_b = 0 \iff S_i^m = S_j^{te} \text{ and } \rho_j^{te} = \rho_j^{te} \mathbf{1}_{\{S_j^{te}\}}, \text{ for } i,j \in \{0,1\},\ i \neq j \end{cases}.$$

Proof of Proposition 2. We have:

$$\mu_b = \int_{S_0^m} \pi_0 \rho_0^{te} dg + \int_{S_1^m} \pi_1 \rho_1^{te} dg.$$

Then from Lemma 2 and based on sets definition, we have:

$$\mu_b = \int_{S_0^{te}} \pi_0 \rho_0^{te} dg + \int_{b_0} \pi_0 \rho_0^{te} dg - \int_{b_1} \pi_0 \rho_0^{te} dg + \int_{S_1^{te}} \pi_1 \rho_1^{te} dg + \int_{b_1} \pi_1 \rho_1^{te} dg - \int_{b_0} \pi_1 \rho_1^{te} dg$$

$$\Longleftrightarrow$$

$$\mu_b = \mu_* + \int_{b_0} (\pi_0 \rho_0^{te} - \pi_1 \rho_1^{te}) dg + \int_{b_1} (\pi_1 \rho_1^{te} - \pi_0 \rho_0^{te}) dg.$$

By virtue of the definition of the sets b_0, b_1, it holds:

$$\begin{cases} g \in b_0 \implies \pi_1 \rho_1^{te} > \pi_0 \rho_0^{te} \\ g \in b_1 \implies \pi_0 \rho_0^{te} > \pi_1 \rho_1^{te} \end{cases}.$$

It immediately leads to $\mu_b \leq \mu_*$. Moreover,

$$\mu_b = \mu_* \implies \mu_L(b_i) = 0, \ (i \in \{0,1\}),$$

and,

$$\mu_L(b_i) = 0, \ (i \in \{0,1\}) \implies \mu_b = \mu_*.$$

Then,

$$\mu_b = \mu_* \iff \mu_L(b_i) = 0, \ (i \in \{0,1\}).$$

Concerning the left hand side of the inequality, we have:

$$\begin{cases} S_0^{te} = (S_0^{te} \cap S_1^m) \cup (S_0^{te} \setminus S_1^m) \\ S_1^{te} = (S_1^{te} \cap S_0^m) \cup (S_1^{te} \setminus S_0^m) \end{cases}.$$

In particular, the intersection of the two members for each equation is empty. Then, we can rewrite μ_b as follows:

$$\mu_b = \int_{S_0^{te} \cap S_1^m} \pi_0 \rho_0^{te} dg + \int_{S_0^{te} \setminus S_1^m} \pi_0 \rho_0^{te} dg + \int_{S_1^{te} \cap S_0^m} \pi_1 \rho_1^{te} dg + \int_{S_1^{te} \setminus S_0^m} \pi_1 \rho_1^{te} dg + \int_{S_0^m \cap S_1^{te}} (\pi_0 \rho_0^{te} - \pi_1 \rho_1^{te}) dg + \int_{S_1^m \cap S_0^{te}}$$

$$\iff$$

$$\mu_b = \int_{S_1^m \cap S_0^{te}} \pi_1 \rho_1^{te} dg + \int_{S_0^m \cap S_1^{te}} \pi_0 \rho_0^{te} dg + \int_{S_0^{te} \setminus S_1^m} \pi_0 \rho_0^{te} dg + \int_{S_1^{te} \setminus S_0^m} \pi_1 \rho_1^{te} dg.$$

As each integrand is positive or null, we have $\mu_b \geq 0$.

1. Let assume that $S_i^m = S_j^{te}$ and $\rho_j^{te} = \rho_j^{te} \mathbf{1}_{\{S_j^{te}\}}$, for $i, j \in \{0,1\}$, $i \neq j$. Then, we have $\mu_b = 0$.

2. Let assume that $\mu_b = 0$. By definition of the different sets, it is easy to show that μ_b is defined as a sum of integrals over disjoint sets. As each integrand is positive or null, it follows that each integral has to be equal to 0. Recalling that $\pi_0 \rho_0^{te} > \pi_1 \rho_1^{te} \geq 0$ over S_0^{te}, it is obvious that we necessary have $S_0^{te} \subseteq S_1^m$. For the same reason, we have $S_1^{te} \subseteq S_0^m$ (from the fourth integral). Let $x \in S_1^m \setminus S_0^{te}$. Then, $x \in S_1^m \cap S_1^{te}$ which is impossible because $S_1^{te} \subseteq S_0^m$. It follows that:

   $$S_i^{te} = S_j^m, i, j \in \{0,1\}, i \neq j.$$

   Then, two ensure that the two first integrals are equal to 0, we necessary have:

   $$\rho_i^{te} = \rho_i^{te} \mathbf{1}_{\{S_i^{te}\}}, i \in \{0,1\}.$$

Finally,

$$\mu_b = 0 \iff \begin{cases} S_0^m = S_1^{te} \iff S_1^m = S_0^{te} \\ \rho_i^{te} = \rho_i^{te} \mathbf{1}_{\{S_i^{te}\}} \end{cases}.$$

In other words, the worst case for μ_b is obtained when the model is as bad as possible. □

C Proof in Section 1.4.2

Lemma 3 Let, $S_0^s \cup S_1^s = \Omega$ such that $S_0^s \cap S_1^s = \emptyset$ (for $s = te$ or v). Then,

$$S_1^{te} \setminus S_1^v = S_0^v \setminus S_0^{te}.$$

Proof of Lemma 3.

$$S_1^{te} \setminus S_1^v = S_1^{te} \setminus (\Omega \setminus S_0^v).$$

Using some set theory properties,

$$S_1^{te} \setminus S_1^v = (S_0^v \cap S_1^{te}) \cup (S_1^{te} \setminus \Omega) = S_0^v \cap S_1^{te} = S_0^v \cap (\Omega \setminus S_0^{te}) = \Omega \cap (S_0^v \setminus S_0^{te}).$$

Then we finally obtain:

$$S_1^{te} \setminus S_1^v = S_0^v \setminus S_0^{te}.$$

□

Proposition 3. We denote $S_j^s = \{g | \pi_j \rho_j^s > \pi_k \rho_k^s\}$ $(k \neq j)$, where $s = te$ (test set) or v (validation set). We denote μ_{te}^c (resp. μ_{te}^p) the test set score obtained with a complete (resp. incomplete) validation set. By complete, we assume that the distribution of ρ_j^{te} and ρ_j^v are the same. Then,

$$\mu_{te}^p \leq \mu_{te}^c.$$

Proof of Proposition 3.

$$\mu_{te}^c = \int_{S_0^v} \pi_0 \rho_0^{te} dg + \int_{S_1^v} \pi_1 \rho_1^{te} dg.$$

As $\rho_j^{te} = \rho_j^v$, we have $S_j^{te} = S_j^v$. Then,

$$\mu_{te}^c = \int_{S_0^{te}} \pi_0 \rho_0^{te} dg + \int_{S_1^{te}} \pi_1 \rho_1^{te} dg.$$

In the incomplete validation case, we have either:

$$\begin{cases} S_1^v \subseteq S_1^{te} \text{ and } S_0^{te} \subseteq S_0^v \\ S_0^v \subseteq S_0^{te} \text{ and } S_1^{te} \subseteq S_1^v \end{cases}.$$

.

By symmetry of the problem, let assume that:

$$S_1^v \subseteq S_1^{te} \text{ and } S_0^{te} \subseteq S_0^v.$$

We then have:

$$\mu_{te}^p = \int_{S_0^v} \pi_0 \rho_0^{te} dg + \int_{S_1^v} \pi_1 \rho_1^{te} dg = \int_{S_0^{te}} \pi_0 \rho_0^{te} dg + \int_{S_1^{te}} \pi_1 \rho_1^{te} dg + \int_{S_0^v \setminus S_0^{te}} \pi_0 \rho_0^{te} dg - \int_{S_1^{te} \setminus S_1^v} \pi_1 \rho_1^{te} dg.$$

Using Lemme 3, we have:

$$\mu_{te}^{p} = \mu_{te}^{c} - \int_{S_1^{te} \setminus S_1^{v}} (\pi_1 \rho_1^{te} - \pi_0 \rho_0^{te}) dg.$$

Moreover, we know that $\pi_1 \rho_1^{te} \geq \pi_0 \rho_0^{te}$ over S_1^{te}. Hence, the second term of the previous equation is positive. Then,

$$\mu_{te}^{p} \leq \mu_{te}^{c}.$$

□

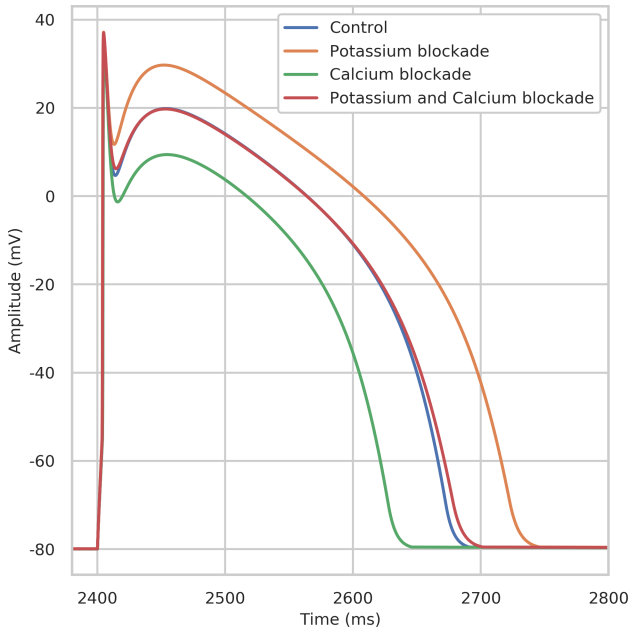

Figure 11: Section 3.2.4: Comparison of different channels blockade (20% of blockade). Sodium channel blockade is mainly known to reduce the depolarization peak, calcium channel blockade is mainly known to reduce the plateau phase and the duration whereas potassium channel blockade is mainly known to induce a signal prolongation.

D MV: scores in the incomplete validation set scenario

For this study we make the following assumptions:

- AP behavior under sodium blockade does not depend on potassium and calcium channel activities.

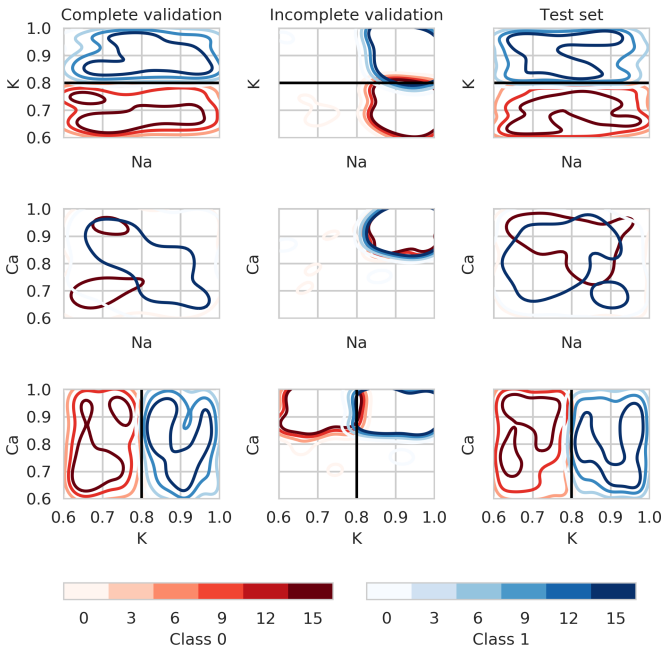

Figure 12: Section 3.2.3: Densities of validation and test sets for potassium channel blockade classification. Black lines correspond to the class delimitation.

- AP behavior under potassium and/or calcium channel blockade are dependent.

The following study is coarse, but presented to justify scores obtained in Section 3.2.4 of the manuscript.

D.1 Sodium channel blockade

In the incomplete validation case, sodium activities for the validation set belong to $(0.85, 1)$. We recall that each activity is a independent realization of a random variable following a uniform law over $(0.6, 1)$. Let assume that for the test set (for which sodium activities belong to $(0.6, 1)$) has n_t elements. Then, we expect to have $0.625 * n_t$ elements over $(0.6, 0.85)$ and $0.375 * n_t$ elements over $(0.85, 1)$. As the set is complete over $(0.85, 1)$ we assume that the training set enrichment is well performed which leads to a perfectly well classified test set over $(0.85, 1)$. Conversely, as we do not have information over $(0.6, 0.85)$ we assume that half of the test set is well classified over this region. It follows that the averaged score μ is:

$$\mu = \frac{\frac{1}{2} * 0.625 * n_t + 0.375 * n_t}{n_t} = 0.6875. \tag{18}$$

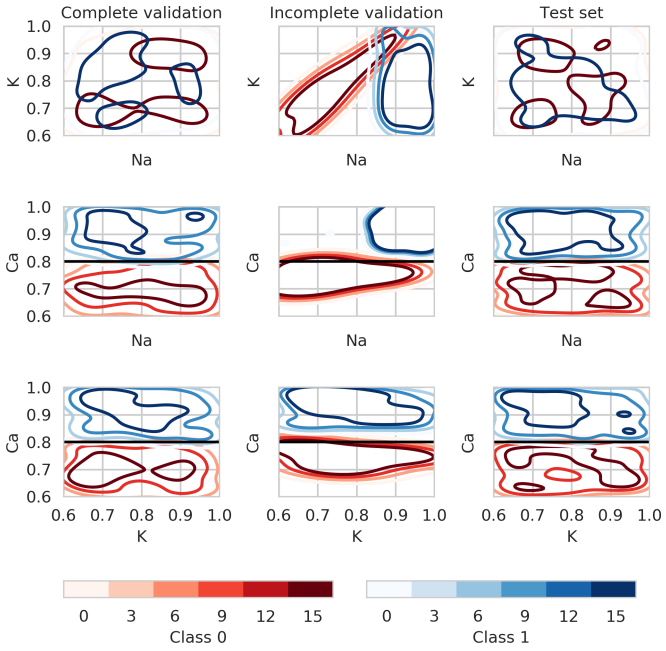

Figure 13: Section 3.2.3: Densities of validation and test sets for calcium channel blockade classification. Black lines correspond to the class delimitation.

Then, by simulation, we expect to have a score close to 0.69 for the sodium channel blockade study in the incomplete validation set case.

D.2 Potassium channel blockade

For this scenario, we use the same idea as the one described in the previous section. The upper panel of Figure 14 shows regions where we well (green), wrongly (red) and partly well (orange) classify the test set. The lower panel shows the ratio between the potassium and the calcium activity.

Over the incomplete validation region, the lowest ratio for the class 1 ($\theta_K > 0.8$ is 0.81 and the highest ratio for class 0 is 0.93. As the minimal ratio in the unknown region: $\{\theta_{Ca} < 0.85 \cup \theta_K > 0.8\}$ is 0.98 all this region will be well classified. The red area is obtained using the same argument. The orange area corresponds to the region where ratios can be from either side of the class delimitation in the incomplete validation set.

Finally, summing the green area and half of the orange area we obtain a score μ which is approximately 0.75.

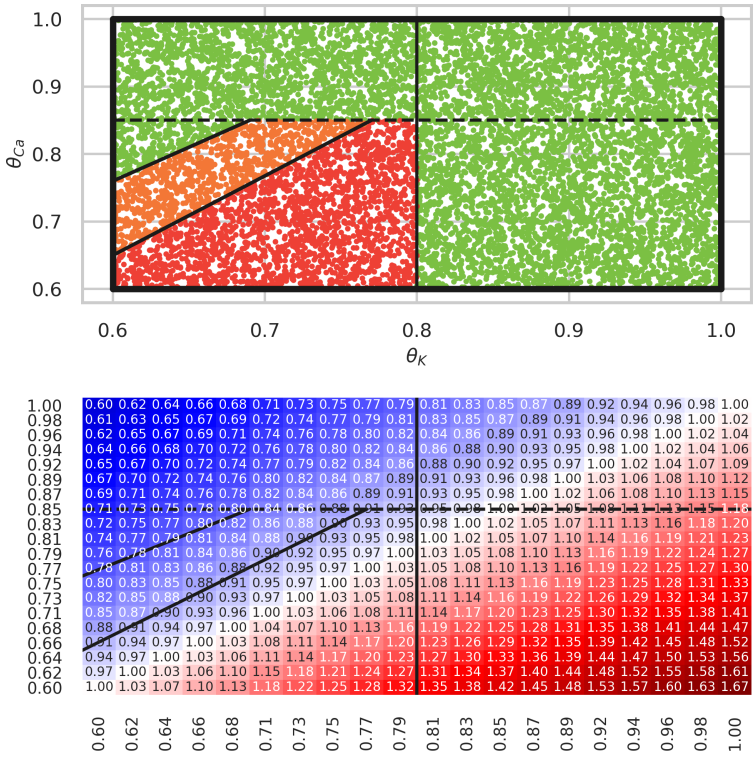

Figure 14: Section D.2: Expected test set classification. Upper panel: red region corresponds to the wrongly classified test set, green region corresponds to the well classified test set and the orange area corresponds to the region where half of the test set is well classified.

D.3 Calcium channel blockade

This scenario uses exactly the same arguments as the one exposed in the previous section. The corresponding figure is shown in Figure 15.

These strategy lead to a score approximately equal to 0.6.

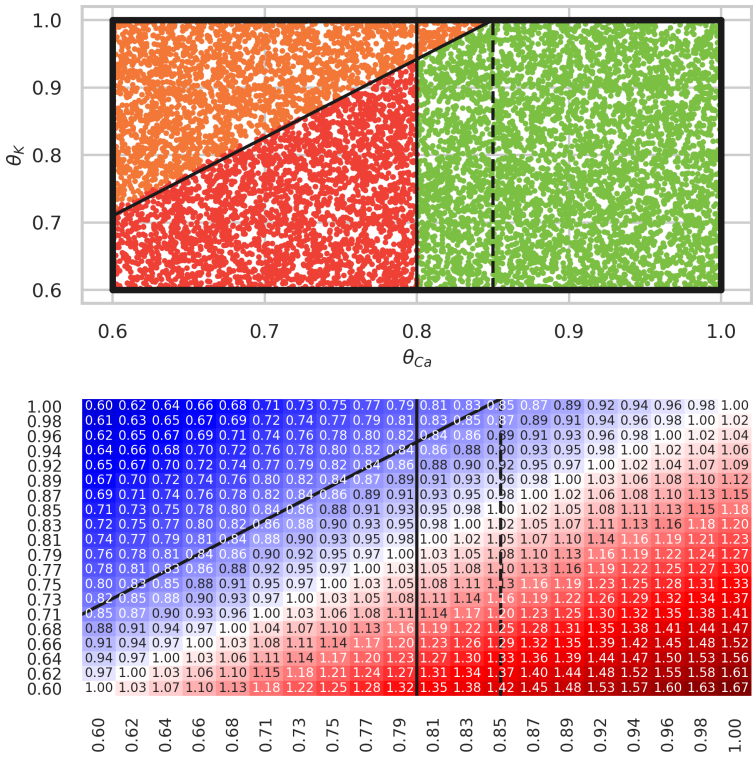

Figure 15: Section D.3: Expected test set classification. Upper panel: red region corresponds to the wrongly classified test set, green region corresponds to the well classified test set and the orange area corresponds to the region where half of the test set is well classified.