Partial data querying through racing algorithms

Vu-Linh Nguyen, Sébastien Destercke, Marie-Hélène Masson

Partial data querying through racing algorithms

Vu-Linh Nguyen[a], Sébastien Destercke[a,*], Marie-Hélène Masson[a,b]

[a] *UMR CNRS 7253 Heudiasyc, Sorbonne Universités, Université de Technologie de Compiègne CS 60319 - 60203 Compiègne cedex, France*
[b] *Université de Picardie Jules Verne, France*

Abstract

The paper studies the problem of actively learning from instances characterized by imprecise features or imprecise class labels, where by actively learning we understand the possibility to query the precise value of imprecisely specified data. We differ from classical active learning by the fact that in the later, data are either fully precise or completely missing, while in our case they can be partially specified. Such situations can appear when sensor errors are important to encode, or when experts have only specified a subset of possible labels when tagging data. We provide a general active learning technique that can be applied in principle to any model. It is inspired from racing algorithms, in which several models are competing against each others. The main idea of our method is to identify the query that will be the most helpful in identifying the winning model in the competition. After discussing and formalizing the general ideas of our approach, we illustrate it by studying the particular case of binary SVM in the case of interval valued features and set-valued labels. The experimental results indicate that, in comparison to other baselines, racing algorithms provide a faster reduction of the uncertainty in the learning process, especially in the case of imprecise features.

Keywords: partial data, interval-valued data, set-valued labels, data querying, active learning, racing algorithms

1. Introduction

Although classical learning schemes assume that every instance is fully specified, there are situations where such an assumption is unlikely to hold, and where the data can be qualified of *partial* or *imprecise*. By "partial data", we refer to the situation where either some features or the labels are imperfectly known, that is are specified by sets of possible values rather than a precise one. For example, when the label of some training instances is only known to belong to a set of labels, or when some features are imprecisely given in the form of intervals.

Classical statistical solutions to solve this problem include the use of different imputation techniques [5] or the use of likelihood-based techniques such as the

*Corresponding author
Email addresses: linh.nguyen@hds.utc.fr (Vu-Linh Nguyen), sebastien.destercke@hds.utc.fr (Sébastien Destercke), mmasson@hds.utc.fr (Marie-Hélène Masson)

EM algorithm [4] and its extensions. The use of such techniques however implies to satisfy specific statistical assumptions about the missingness process (e.g., missing-at-random assumption), that can be very hard or impossible to check in practice, especially since we do not have access to the original precise data. More recently, the problem of learning from partial data has gained an increasing interest within the machine learning community, and many methods [2, 3, 10] that have shown their efficiency for different problems have been developed. Yet, even if these methods can handle partial data, their performances usually degrade as data become more and more partial or imprecise, as more and more uncertainty is present in the learning process.

This work explores the following question about learning from partial data: if we have the possibility to gain more information on some of the partial instances, which instance and what feature of this instance should we query? In the case of a completely missing label (and to a lesser extent of missing features), this problem known as active learning has already been largely treated [16] and applied in different fields like natural language processing, text or image classification, recommender systems [6, 14, 23, 19]. However, we are not aware of similar works concerning the case of partial data. Note that for the case of features, there is even very few active learning methods addressing the problem of missing features. In this work, we provide a new general active learning technique that can be applied in principle to any model and partially missing input/features, and illustrate it on the case of SVM. It is inspired from the concept of racing algorithms [12], in which several models are competing against each others. They were initially introduced to select an optimal configuration of a given lazy learning model (e.g., K-nn methods), and since then have been applied to other settings such as multi-armed bandits [9]. The idea of such racing algorithms is to oppose a (finite) set of alternatives in a race, and to progressively discard losing ones as the race goes along. In our case, the set of alternatives will be different possible models, and the race will consist in iteratively querying the precise value of some partial features or labels. Indeed, as data are partial, the performance of each model is uncertain and several candidate models can be optimal. By iteratively making queries, i.e. asking to an oracle the precise value of a partial data, these performances will become less and less uncertain, and more models will be discarded from the race. The key question is then to identify those data that will be the most helpful in reducing the set of possible winners in the race, in order to converge as quickly as possible to the optimal model.

The rest of this paper is organized as follows: we present in Section 2 the basic notations used in this paper. Section 3 introduces the general principles of racing algorithms and formalizes the problem of quantifying the influence of a query on the race. We then study the application of our approach using the particular case of a binary SVM. Section 4 is focused on interval-valued features, while Section 5 explores the case of set-valued labels. Some experiments are then performed in Section 6 to demonstrate the effectiveness of our proposals. Before concluding the paper, Section 7 discusses some computational issues of the presented approaches, generalizing some of the results concerning SVM

method. Note that this paper is an extension of [13], with full proofs, larger experiments as well as the addition of the set-valued label case for binary SVM and a discussion about the complexity of the approach.

2. Preliminaries

In classical supervised setting, the goal of the learning approach is to find a model $m : \mathcal{X} \to \mathcal{Y}$ within a set $\mathcal{M}$ of models from a set $\mathbf{D} = \{(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y} | i = 1, \ldots, n\}$ of n input/output samples, where $\mathcal{X}$ and $\mathcal{Y}$ are respectively the input and the output spaces[1]. The empirical risk $R(m)$ associated to a model m is then evaluated as

$$R(m) = \sum_{i=1}^{n} \ell(y_i, m(\mathbf{x}_i)) \tag{1}$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is the loss function, and $\ell(y, m(\mathbf{x}))$ is the loss of predicting $m(\mathbf{x})$ when observing y. The selected model is then the one minimizing (1), that is

$$m^* = \arg\min_{m \in \mathcal{M}} R(m). \tag{2}$$

Another way to see the model selection problem that will be useful in this paper is to assume that a model m_l is said to be better than m_k (denoted $m_l \succ m_k$) if

$$R(m_k) - R(m_l) > 0, \tag{3}$$

or in other words if the risk of m_l is lower than the risk of m_k. Given the relation $\succ$ on $\mathcal{M}$, Equation (1) then simply amounts to take as best model the maximal element of $\succ$, or in case of equality due to indifference, one of the maximal model chosen arbitrarily.

In this work, we are however interested in the case where data are partial, that is where general samples are of the kind $(\mathbf{X}_i, Y_i) \subseteq \mathcal{X} \times \mathcal{Y}$. Here and in the rest of this paper, capital letters are used for partial data and small letters will denote precise one, and bold letters will represent vectors and Cartesian products of feature values. When the data is partial, Equations (1), (2) and (3) are no longer well-defined, and can be extended in multiple different ways. Two of the most common ways to extend them is either to use a minimin (optimistic) or a maximin (pessimistic) approach [20, 22]. That is, if we extend Equation (1) to a lower bound

$$\begin{aligned} \underline{R}(m) &= \inf_{(\mathbf{x}_i, y_i) \in (\mathbf{X}_i, Y_i)} \sum_{i=1}^{n} \ell(y_i, m(\mathbf{x}_i)) \\ &= \sum_{i=1}^{n} \inf_{(\mathbf{x}_i, y_i) \in (\mathbf{X}_i, Y_i)} \ell(y_i, m(\mathbf{x}_i)) := \sum_{i=1}^{n} \underline{\ell}(Y_i, m(\mathbf{X}_i)) \end{aligned} \tag{4}$$

[1] As $\mathcal{X}$ is often multi-dimensional, we will denote its elements and subsets by bold letters.

and an upper bound

$$\overline{R}(m) = \sup_{(\mathbf{x}_i,y_i)\in(\mathbf{X}_i,Y_i)} \sum_{i=1}^{n} \ell(y_i, m(\mathbf{x}_i)) \tag{5}$$
$$= \sum_{i=1}^{n} \sup_{(\mathbf{x}_i,y_i)\in(\mathbf{X}_i,Y_i)} \ell(y_i, m(\mathbf{x}_i)) := \sum_{i=1}^{n} \overline{\ell}(Y_i, m(\mathbf{X}_i))$$

then the optimal minimin m^*_{mm} and maximin m^*_{Mm} models are

$$m^*_{mm} = \arg\min_{m\in\mathcal{M}} \underline{R}(m) \quad \text{and} \quad m^*_{Mm} = \arg\min_{m\in\mathcal{M}} \overline{R}(m).$$

The minimin approach usually assumes that data are distributed according to the model, and tries to find the best data replacement (or disambiguation) combined with the best possible model [10]. Conversely, the maximin approach assumes that data are distributed in the worst possible way, and select the model performing the best in the worst situation, thus guaranteeing a minimal performance of the model [21]. However, such an approach, due to the overly conservative nature of its assumptions, will often lead to sub-optimal model, so we will prefer the first principle.

It should be noted that both the minimin and maximin approaches lead to choose a unique optimal model, despite the uncertainty present in the data. Our work focuses on a different approach, where we do not search for an optimal model right away, but rather consider sets of potentially optimal models to then try to identify the best one through querying. In this case, we consider that a model m_l is better than m_k (still denoted $m_l \succ m_k$) if

$$\underline{R}(m_{k-l}) = \inf_{(\mathbf{x}_i,y_i)\in(\mathbf{X}_i,Y_i)} R(m_k) - R(m_l) > 0, \tag{6}$$

which is a direct extension of Equation (3). That is, $m_l \succ m_k$ if and only if it is better under every possible precise replacement $(\mathbf{x}_i, y_i)$ consistent with the partial instances $(\mathbf{X}_i, Y_i)$. We can then denote by

$$\mathcal{M}^* = \{m \in \mathcal{M} : \nexists m' \in \mathcal{M}\, s.t.\, m' \succ m\} \tag{7}$$

the set of undominated models within $\mathcal{M}$, that is the set of models that are maximal with respect to the partial order $\succ$. The practical computation of (4)-(6) depends on the type of classifier considered in the race and will be explained in details in section 4 for the particular case of binary SVM.

Example 1. Figure 1 illustrates a situation where $\mathcal{Y}$ consists of two different classes (grey and white), and $\mathcal{X}$ of two dimensions. Only imprecise data are numbered: squares are assumed to have precise features, and unknown labels are represented by striped squares (i.e., a data with partial label and features would be a striped rectangle). Assuming that we only have two models $\mathcal{M} = \{m_1, m_2\}$ to compare (the models in Figure 1 could be decision stumps, or one-level decision trees), we would choose $m_2 = m^*_{Mm}$ as the maximin model and $m_1 = m^*_{mm}$ as the minimin one. The two models would however be incomparable according to (6), as both $\underline{R}(m_{1-2})$ and $\underline{R}(m_{2-1})$ are negative, hence $\mathcal{M}^* = \mathcal{M}$ in this case. The rest of the paper then deals with the data to query in order to reduce $\mathcal{M}^*$

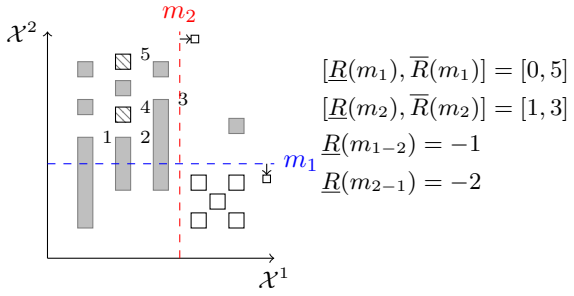


Figure 1: Illustration of partial data and competing models

3. Partial data querying: a racing approach

Both the minimin and maximin approaches pursue the same goal: obtaining a unique model from partially specified data. In this sense, they are quite close to approaches using imputation or EM algorithms. The idea we defend in this paper is different: we want to identify and query those data that will be the most helpful in reducing the set $\mathcal{M}^*$. In order to this, we will try to quantify how useful an information is to decide what is the best model among those in $\mathcal{M}^*$. We will now formalise this idea. We first assume that $\mathcal{X} = \mathcal{X}^1 \times \ldots \times \mathcal{X}^p$ is a Cartesian product of p spaces, and that a partial data $(\mathbf{X}_i, Y_i)$ can be expressed as $(\times_{j=1}^p X_i^j, Y_i)$, and furthermore that if $\mathcal{X}^j \subseteq \mathbb{R}$ is a subset of the real line, then X_i^j is an interval. The data who have imprecise features in Figure 1 could be of this kind.

A query on a partial data $(\times_{j=1}^p X_i^j, Y_i)$ consists in transforming one of its dimension X_i^j or Y_i into the true precise value (x_i^j or y_i) provided by an oracle (an expert, a precise measuring device). More precisely, Q_i^j denotes the query made on X_i^j or Y_i, with $j = p+1$ for Y_i. Given a model m_l and a data $(\times_{j=1}^p X_i^j, Y_i)$, the result of a query can have an effect on the interval $[\underline{R}(m_l), \overline{R}(m_l)]$, depending on whether it changes the interval $[\underline{\ell}(Y_i, m_l(\mathbf{X}_i)), \overline{\ell}(Y_i, m_l(\mathbf{X}_i))]$. Similarly, when assessing whether the model m_l is preferred to m_k, the query can have an influence on the value $\underline{R}(m_{k-l})$ or not. We formalise this by two functions, $E_{Q_i^j} : \mathcal{M} \to \{0, 1\}$ and $J_{Q_i^j} : \mathcal{M} \times \mathcal{M} \to \{0, 1\}$ such that:

$$E_{Q_i^j}(m_l) = \begin{cases} 1 & \text{if } \exists x_i^j \in X_i^j \text{ that reduces } [\underline{R}(m_l), \overline{R}(m_l)] \\ 0 & \text{else} \end{cases} \tag{8}$$

and

$$J_{Q_i^j}(m_k, m_l) = \begin{cases} 1 & \text{if } \exists x_i^j \in X_i^j \text{ that increases } \underline{R}(m_{k-l}) \\ 0 & \text{else.} \end{cases} \tag{9}$$

When $j = p+1$, X_i^j is to be replaced by Y_i. $E_{Q_i^j}(m_l)$ simply tells us whether or not the query can affect our evaluation of the model m_l, while $J_{Q_i^j}(m_k, m_l)$ informs us whether the query can help to tell apart m_l from m_k.

Example 2. In Figure 1, questions related to partial classes (points 4 and 5) and to partial features (points 1, 2 and 3) have respectively the same potential effect, so we can restrict our attention to Q_4^3 (the class of point 4) and to Q_1^2 (the second feature of point 1). For these two questions, we have

- $E_{Q_4^3}(m_1) = E_{Q_4^3}(m_2) = 1$ and $J_{Q_4^3}(m_1, m_2) = J_{Q_4^3}(m_2, m_1) = 0$.
- $E_{Q_1^2}(m_1) = 1$, $E_{Q_1^2}(m_2) = 0$ and $J_{Q_1^2}(m_1, m_2) = J_{Q_1^2}(m_2, m_1) = 1$.

This example shows that while some questions may reduce our uncertainty about many model risks (Q_4^3 reduce risk intervals for both models), they may be less useful than other questions to tell two models apart (Q_1^2 can actually lead to declare m_2 better than m_1), hence it is useful to consider both individual and pairwise effects of a unique query.

Following the idea of racing algorithms, which concentrate on the best potential model, Definitions (8) and (9) allow us to define the value of query as follows:

Definition 1. Given m_{k^*} the best current potential model, the value of a query Q_i^j is defined as

$$Value(Q_i^j) = E_{Q_i^j}(m_{k^*}) + \sum_{k \neq k^*} J_{Q_i^j}(m_k, m_{k^*}). \tag{10}$$

We can now finally propose our querying method inspired by racing algorithms, that consists in building an initial set $\{m_1, \ldots, m_R\}$ of models, and then make them race against each other. The initial set can be instantiated by sampling several precise data sets $(\mathbf{x}_i, y_i) \in (\mathbf{X}_i, Y_i)$, and then learning an optimal model from each of these precise selection. Algorithm 1 summarises the general procedure applied to find the best query and to update the race once this set is built. This algorithm simply searches the query that will have the biggest impact on the minimin model and its competitors, adopting the optimistic attitude of racing algorithms. Once a query has been made, the data set as well as the set of competitors are updated, so that only potentially optimal models remain.

Notice that the best model (learned from fully precise data) may not be in the set $\mathcal{M}$ of competitors. This is also true for some active learning techniques such as Query-by-committee. This means that, at the end of the querying process, two solutions arise: either retain the best model m_{k^*} within $\mathcal{M}$, or retrain a new model from the completed data. Note that since we will not query all partial data in practice (otherwise trying to find best queries is meaningless), we will have to use learning techniques able to cope with such data [11]

In the next sections, we illustrate our proposed setting and its potential interest with the popular SVM algorithm. We separate the two cases of interval-valued features from set-valued labels, for three reasons: (i) we can expect that imprecision in both aspects is less likely to happen in practice, (ii) this makes the exposure of the methods easier to follow, and (iii) considering both cases at

once would quickly induce a too important imprecision in the results. We leave the combination of the two approaches to the reader, especially since binary SVM are here used as an illustration of our general approach.

Algorithm 1: One iteration of the racing algorithm to query data.

Input: data (X_i, Y_i), set $\{m_1, \ldots, m_R\}$ of models
Output: updated data and set of models

$k^* = \arg\min_{k \in \{1,\ldots,R\}} \underline{R}(m_k)$;
foreach *query* Q_i^j **do**
  $Value(Q_i^j) = E_{Q_i^j}(m_{k^*}) + \sum_{k \neq k^*} J_{Q_i^j}(m_k, m_{k^*})$;
$Q_{i^*}^{j^*} = \arg\max_{Q_i^j} Value(Q_i^j)$;
Get value $x_{i^*}^{j^*}$ of $X_{i^*}^{j^*}$;
foreach $k, l \in \{1, \ldots, R\} \times \{1, \ldots, R\}$, $k \neq l$ **do**
  Compute $\underline{R}(m_{k-l})$;
  if $\underline{R}(m_{k-l}) > 0$ **then** remove m_k from $\{m_1, \ldots, m_R\}$;

4. Application to binary SVM: interval-valued features

In the binary SVM setting [1], the input space $\mathcal{X} = \mathbb{R}^p$ is the real space and the binary output space is $\mathcal{Y} = \{-1, 1\}$, where $-1, 1$ encode the two possible classes. The model $m_l = (\mathbf{w}_l, c_l)$ corresponds to the "maximum-margin" hyperplane $\mathbf{w}_l \mathbf{x} + c_l$ with $\mathbf{w}_l \in \mathbb{R}^p$ and $c_l \in \mathbb{R}$. For convenience sake, we will use $(\mathbf{w}_l, c_l)$ and m_l interchangeably from now on. We will also focus in this section on the case of imprecise features and precise labels, and will denote y_i the label of training instances. We will also focus on the classical $0-1$ loss function defined as follows for an instance $(\mathbf{x}_i, y_i)$:

$$\ell(y_i, m_l(\mathbf{x}_i)) = \begin{cases} 0 \text{ if } y_i \cdot m_l(\mathbf{x}_i) \geq 0 \\ 1 \text{ if } y_i \cdot m_l(\mathbf{x}_i) < 0, \end{cases} \quad := \ell_l(y_i, \mathbf{x}_i) \tag{11}$$

where $m_l(\mathbf{x}_i) = \mathbf{w}_l \mathbf{x}_i + c_l$, and $\ell_l(y_i, x_i)$ is used as a short notation for $\ell(y_i, m_l(\mathbf{x}_i))$.

4.1. Instances inducing imprecision in empirical risk

Before entering into the details of how single risk bounds $[\underline{R}(m_l), \overline{R}(m_l)]$ and pairwise risk bounds $\underline{R}(m_{k-l})$ given by Equations (4)-(6), and query effects $E_{Q_i^j}(m_l)$ and $J_{Q_i^j}(m_k, m_l)$ given by Equations (8)-(9) can be estimated in practice, we will first investigate under which conditions an instance $(\mathbf{X}_i, y_i)$ induces imprecision in the empirical risk. Such instances are the only ones of interest here, since if $\underline{\ell}_l(y_i, \mathbf{X}_i) = \overline{\ell}_l(y_i, \mathbf{X}_i) = \ell_l(y_i, \mathbf{X}_i)$, then $E_{Q_i^j}(m_l) = 0$ for all $j = 1, \ldots, p$. Furthermore, if an instance $(\mathbf{X}_i, y_i)$ is precise w.r.t both m_k and m_l, then $J_{Q_i^j}(m_k, m_l) = 0$ for all $j = 1, \ldots, p$. Thus, only instances which are imprecise w.r.t at least one model are interested when determining $J_{Q_i^j}(m_k, m_l)$.

Definition 2. Given a SVM model m_l, an instance $(\mathbf{X}_i, y_i)$ is called an imprecise instance w.r.t. m_l if and only if

$$\exists \mathbf{x}_i^{'}, \mathbf{x}_i^{''} \in \mathbf{X}_i \text{ s.t } m_l(\mathbf{x}_i^{'}) \geq 0 \text{ and } m_l(\mathbf{x}_i^{''}) < 0. \tag{12}$$

Instances that do not satisfy Definition 2 will be called precise instances (w.r.t. m_l). Being precise means that the sign of $m_l(\mathbf{x}_i)$ is the same for all $x_i \in \mathbf{X}_i$, which implies that the loss $\underline{\ell}_l(y_i, \mathbf{X}_i) = \overline{\ell}_l(y_i, \mathbf{X}_i)$ is precisely known. The next example illustrates the notion of (im)precise instances.

Example 3. Figure 2 illustrates a situation with two models and where the two different classes are represented by grey ($y = +1$) and white ($y = -1$) colours. From the figure, we can say that $(\mathbf{X}_1, y_1)$ is precise w.r.t both m_1 and m_2, $(\mathbf{X}_2, y_2)$ is precise w.r.t m_1 and imprecise w.r.t m_2, $(\mathbf{X}_3, y_3)$ is imprecise w.r.t both m_1 and m_2 and $(\mathbf{X}_4, y_4)$ is imprecise w.r.t m_1 and precise w.r.t m_2.

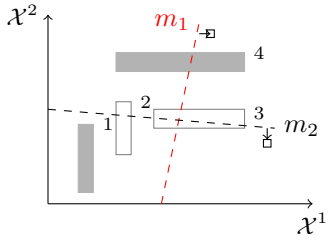


Figure 2: Illustration of interval-valued instances

Determining whether an instance is imprecise w.r.t. m_l is actually very easy in practice. Let us denote by

$$\underline{m}_l(\mathbf{X}_i) := \inf_{\mathbf{x}_i \in \mathbf{X}_i} m_l(\mathbf{x}_i) \text{ and } \overline{m}_l(\mathbf{X}_i) := \sup_{\mathbf{x}_i \in \mathbf{X}_i} m_l(\mathbf{x}_i) \tag{13}$$

the lower and upper bounds reached by model m_l over the space $\mathbf{X}_i$. The following result characterizing imprecise instances, as well as when a hyperplane $m_l(\mathbf{x}_i) = 0$ intersects with a region $\mathbf{X}_i$, follows from the fact that the image of a compact set by a continuous function is also compact.

Proposition 1. *Given $m_l(\mathbf{x}_i) = \mathbf{w}_l \mathbf{x}_i + c_l$ and the set $\mathbf{X}_i$, then $(\mathbf{X}_i, y_i)$ is imprecise w.r.t. m_l if and only if*

$$\underline{m}_l(\mathbf{X}_i) < 0 \text{ and } \overline{m}_l(\mathbf{X}_i) \geq 0. \tag{14}$$

Furthermore, we have that the hyperplane $m_l(\mathbf{x}_i) = 0$ intersects with the region $\mathbf{X}_i$ if and only if (14) *holds. In other words, $\exists \mathbf{x}_i \in \mathbf{X}_i$ s.t. $m_l(\mathbf{x}_i) = 0$.*

Proof. Since continuous functions preserve compactness and connectedness [7], then the image $f(\mathbf{X}) = Y$ of a compact and connected set $\mathbf{X}$ is compact and

connected. Furthermore, a set on $\mathbf{R}^p$ is compact if and only if it is closed and bounded (Heine–Borel Theorem [15]), then $\mathbf{X}$ is a closed, bounded and connected set which is exactly a closed interval. Or in other words, we have that

$$m_l(\mathbf{X}_i) = \left[\underline{m}_l(\mathbf{X}_i), \overline{m}_l(\mathbf{X}_i)\right],$$

is an interval consisting of every possible values that can take $m_l(\mathbf{x}_i)$ for $\mathbf{x}_i \in \mathbf{X}_i$. That (14) is equivalent to (12) then immediately follows. Also, we have that $\exists \mathbf{x}_i \in \mathbf{X}_i$ s.t. $m_l(\mathbf{x}_i) = 0$ if and only if $0 \in \left[\underline{m}_l(\mathbf{X}_i), \overline{m}_l(\mathbf{X}_i)\right]$. □

This proposition means that to determine whether an instance $(\mathbf{X}_i, y_i)$ is imprecise, we only need to compute values $\underline{m}_l(\mathbf{X}_i)$ and $\overline{m}_l(\mathbf{X}_i)$, which can be easily done using Proposition 2.

Proposition 2. *Given $(\mathbf{X}_i, y_i)$ with $X_i^j = \left[a_i^j, b_i^j\right]$ and SVM model $(\mathbf{w}_l, c_l)$, we have*

$$\overline{m}_l(\mathbf{X}_i) = \sum_{w_l^j \geq 0} w_l^j b_i^j + \sum_{w_l^j < 0} w_l^j a_i^j + c_l$$
$$\underline{m}_l(\mathbf{X}_i) = \sum_{w_l^j \geq 0} w_l^j a_i^j + \sum_{w_l^j < 0} w_l^j b_i^j + c_l.$$

Proof. Since $m_l(\mathbf{x}_i)$ is a linear function, it is monotonic in each dimension, hence the extreme values are obtained at points $\mathbf{x}_i \in \times_{j=1}^p \{a_i^j, b_i^j\}$. Furthermore, $m_l(\mathbf{x}_i)$ decreases (increases) w.r.t $\mathbf{x}_i^j$ if $w_l^j < 0$ ($w_l^j > 0$). Hence, Proposition 2 holds. □

Again, it should be noted that only imprecise instances are of interest here, as these are the only instances that, once queried, can result in an increase of the lower empirical risk bounds. We will therefore focus on those in the next sections.

Example 4. Consider the model m_l on a 3-dimensional space given by $\mathbf{w}_l = (2, -1, 1)$ and the partial instance $\mathbf{X}_i = [1, 3] \times [2, 5] \times [1, 2]$. In this case, we have

$$\underline{m}_l(\mathbf{X}_i) = 1 \times 2 + 5 \times -1 + 1 \times 1 = -2,$$
$$\overline{m}_l(\mathbf{X}_i) = 3 \times 2 + 2 \times -1 + 2 \times 1 = 6,$$

hence the instance $\mathbf{X}_i$ is imprecise with respect to m_l

4.2. Empirical risk bounds and single effect

We are now going to investigate the practical computation of $\underline{R}(m_l)$, $\overline{R}(m_l)$, as well as the value $E_{Q_i^j}(m_l)$ of a query on a model m_l. Equations (4) (resp. (5)) implies that the computation of $\underline{R}(m_l)$ (resp. $\overline{R}(m_l)$) can be done by first

computing $\underline{\ell}_l(y_i, \mathbf{X}_i)$ (resp. $\overline{\ell}_l(y_i, \mathbf{X}_i)$) for $i = 1, \ldots, n$ and then summing the obtained values. This means that we can focus our attention on computing $\underline{\ell}_l(y_i, \mathbf{x}_i)$ and $\overline{\ell}_l(y_i, \mathbf{x}_i)$ for a single instance, as obtaining $\underline{R}(m_l)$, $\overline{R}(m_l)$ from them is straightforward. Note that we have $\underline{\ell}_l(y_i, \mathbf{X}_i) = 0$ and $\overline{\ell}_l(y_i, \mathbf{X}_i) = 1$ if and only if $\mathbf{X}_i$ is imprecise w.r.t. m_l, a fact that can easily be checked using Proposition 1. The bounds of the loss interval for the model m_l and datum $(\mathbf{X}_i, y_i)$ is

$$[\underline{\ell}_l(y_i, \mathbf{X}_i), \overline{\ell}_l(y_i, \mathbf{X}_i)] = \begin{cases} [0,0] \text{ if } \min(y_i \cdot \overline{m}_l(\mathbf{X}_i), y_i \cdot \underline{m}_l(\mathbf{X}_i)) \geq 0 \\ [0,1] \text{ if } \overline{m}_l(\mathbf{X}_i) \cdot \underline{m}_l(\mathbf{X}_i) < 0 \\ [1,1] \text{ if } \max(y_i \cdot \overline{m}_l(\mathbf{X}_i), y_i \cdot \underline{m}_l(\mathbf{X}_i)) < 0 \end{cases} \tag{15}$$

Let us now focus on estimating the effect of a query. As with the loss bounds, the only situation where a query Q_i^j can affect the empirical risk bounds, and hence the only situation where $E_{Q_i^j}(m_l) = 1$, is when the interval $[\underline{\ell}_l(y_i, \mathbf{X}_i), \overline{\ell}_l(y_i, \mathbf{X}_i)]$ can be reduced by querying X_i^j. Therefore we can also focus on a single instance to evaluate it. In the case of 0-1 loss, the only case where $E_{Q_i^j}(m_l) = 1$ is the one where $[\underline{\ell}_l(y_i, \mathbf{X}_i), \overline{\ell}_l(y_i, \mathbf{x}_i)]$ goes from $[0,1]$ before the query to a precise value after it, or in other words if there is $x_i^j \in X_i^j$ such that $\mathbf{X}_i^{'} = \times_{j' \neq j} X_i^{j'} \times \{x_i^j\}$ is precise w.r.t. m_l. According to Proposition 1, this means that either $\underline{m}_l(\mathbf{X}_i^{'})$ should become positive, or $\overline{m}_l(\mathbf{X}_i^{'})$ should become negative after a query Q_i^j. The conditions to check whether this is possible are given in the next proposition.

Proposition 3. *Given $(\mathbf{X}_i, y_i)$ with $X_i^j = \left[a_i^j, b_i^j\right]$ and a model m_l s.t. $\mathbf{X}_i$ is imprecise, then $E_{Q_i^j}(m_l) = 1$ if and only if one of the following conditions holds*

$$\underline{m}_l(\mathbf{X}_i) \geq -|w_l^j|(b_i^j - a_i^j) \tag{16}$$

or

$$\overline{m}_l(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j). \tag{17}$$

Proof. Let us concentrate on the first condition (the second one can be proved similarly). If we denote by $\underline{m}_l^{Q_i^j}$ the lower bound reached by m_l on $\mathbf{X}_i'$ (the set resulting from the query answer), then we have the following inequality

$$\underline{m}_l^{Q_i^j}(\mathbf{X}_i^{'}) \leq \underline{m}_l(\mathbf{X}_i) + |w_l^j|(b_i^j - a_i^j)$$

giving us a tight upper bound for it. Indeed, if $w_l^j \geq 0$, then $\underline{m}_l$ is obtained for $x_i^j = a_i^j$ (by Proposition 2), and it can increase by at most $w_l^j(b_i^j - a_i^j)$ if the result of the query Q_i^j is $x_i^j = b_i^j$ (the case $w_l^j \leq 0$ is similar). Since $\underline{m}_l(\mathbf{X}_i)$ is known to be negative (from Proposition 1 and the fact that $\mathbf{X}_i$ is imprecise), it can only become positive after a query Q_i^j if $\underline{m}_l(\mathbf{X}_i) + |w_l^j|(b_i^j - a_i^j)$ is positive.

Finally, by investigating the change of $sign(w_l^j)$, we have:

B1: Q_i^j can change the sign of $\underline{m}_l(\mathbf{x}_i)$ iff

$$\begin{cases} \underline{m}_l(\mathbf{x}_i) + w_l^j(b_i^j - a_i^j) \geq 0 \text{ if } w_l^j \geq 0, \\ \underline{m}_l(\mathbf{x}_i) - w_l^j(b_i^j - a_i^j) \geq 0 \text{ if } w_l^j < 0. \end{cases}$$

B2: Q_i^j can change the sign of $\overline{m}_l(\mathbf{x}_i)$ iff

$$\begin{cases} \overline{m}_l(\mathbf{x}_i) - w_l^j(b_i^j - a_i^j) < 0 \text{ if } w_l^j \geq 0 \\ \overline{m}_l(\mathbf{x}_i) + w_l^j(b_i^j - a_i^j) < 0 \text{ if } w_l^j < 0. \end{cases}$$

□

$\underline{R}(m_l), \overline{R}(m_l)$, needed in the line 1 of Algorithm 1 to identify the most promising model k^*, are computed easily by summing over all training instances the intervals $[\underline{\ell}_l(y_i, \mathbf{X}_i), \overline{\ell}_l(y_i, \mathbf{X}_i)]$ given by Equation (15), while Equations (16)-(17) give easy ways to estimate the values of $E_{Q_i^j}(m_{k^*})$, needed in line 3 of Algorithm 1.

Example 5. Let us consider again Example 4, and check whether querying the last ($j = 3$) or second dimension may induce some effect on the emprical risk bounds. Using Proposition 3, we have for Q_i^3 that

$$\underline{m}_l(\mathbf{X}_i) = -2 < -1 \times (2 - 1) \text{ and } \overline{m}_l(\mathbf{X}_i) = 6 > 1 \times (2 - 1),$$

hence $E_{Q_i^3}(m_l) = 0$, as none of the conditions are satisfied. We do have, on the contrary, that

$$\underline{m}_l(\mathbf{X}_i) = -2 \geq -1 \times (5 - 2),$$

hence $E_{Q_i^2}(m_l) = 1$. Indeed, if $x_i^2 = 2$ (the query results in the lower bound), then the model becomes positive for any replacement of $\mathbf{X}_i' = [1, 3] \times 2 \times [1, 2]$.

4.3. Pairwise risk bounds and effect

Let us now focus on how to compute, for a pair of models m_k and m_l, whether a query Q_i^j will have an effect on the value $\underline{R}(m_{k-l})$. For this, we will have to compute $\underline{R}(m_{k-l})$, which is a necessary step to estimate the indicator $J_{Q_i^j}(m_k, m_l)$ of a possible effect of Q_i^j. To do that, note that $\underline{R}(m_{k-l})$ can be rewritten as

$$\underline{R}(m_{k-l}) = \inf_{\mathbf{x}_i \in \mathbf{X}_i, i=1,\ldots,n} (R(m_k) - R(m_l)) = \sum_{i=1}^{n} \underline{\ell}_{k-l}(y_i, \mathbf{X}_i) \tag{18}$$

with

$$\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = \inf_{\mathbf{x}_i \in \mathbf{X}_i} \Big(\ell_k(y_i, \mathbf{x}_i) - \ell_l(y_i, \mathbf{x}_i) \Big), \tag{19}$$

meaning that computing $\underline{R}(m_{k-l})$ can be done by summing up $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$ over all $\mathbf{X}_i$, similarly to $\underline{R}(m_l)$ and $\overline{R}(m_l)$. Also, $J_{Q_i^j}(m_k, m_l) = 1$ if and only if Q_i^j can increase $\underline{R}(m_{k-l})$. We can therefore focus on the computation of $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$ and its possible changes. First note that if $\mathbf{X}_i$ is precise w.r.t. both m_k and m_l, then $\ell_k(y_i, \mathbf{X}_i) - \ell_l(y_i, \mathbf{X}_i)$ is a well-defined value, as each loss is precise, and in this case $J_{Q_i^j}(m_k, m_l) = 0$. Therefore, the only cases of interest are those where $\mathbf{X}_i$ is imprecise w.r.t. to at least one model. We will first treat the case where it is imprecise for only one, and then we will proceed to the more complex one where it is imprecise w.r.t. both. Note that imprecision with respect to each model can be easily established using Proposition 1.

4.3.1. Imprecision with respect to one model

Let us consider the case where $\mathbf{X}_i$ is imprecise w.r.t. either m_k or m_l. In each of these two cases, the loss induced by $(\mathbf{X}_i, y_i)$ on the model for which it is precise is fixed. Hence, to estimate the lower loss $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$, as well as the effect of a possible query Q_i^j, we only have to look at the model for which $(\mathbf{X}_i, y_i)$ is imprecise. The next proposition establishes the lower bound $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$, necessary to compute $\underline{R}(m_{k-l})$.

Proposition 4. *Given $(\mathbf{X}_i, y_i)$ with $X_i^j = [a_i^j, b_i^j]$ and two models m_k and m_l s.t $(\mathbf{X}_i, y_i)$ is imprecise w.r.t. one and only one model, then we have*

$$\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = \ell_k(y_i, \mathbf{X}_i) - 1 \qquad \text{if } \mathbf{X}_i \text{ imprecise w.r.t. } m_l \tag{20}$$

$$\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = 0 - \ell_l(y_i, \mathbf{X}_i) \qquad \text{if } \mathbf{X}_i \text{ imprecise w.r.t. } m_k. \tag{21}$$

Proof. We will only prove Equation (20), the proof for Equation (21) being similar. First note that if $\mathbf{X}_i$ is precise with respect to m_k, then $\ell_k(y_i, \mathbf{X}_i)$ is precise. Second, the value of $\ell_l(y_i, \mathbf{X}_i) \in \{0, 1\}$, since $\mathbf{X}_i$ is imprecise with respect to m_l, hence the lower bound is obtained for $\mathbf{x}_i \in \mathbf{X}_i$ such that $\ell_l(y_i, \mathbf{x}_i) = 1$. □

We kept the 0 in Equation (21) to make clear that we take the lower bound of the loss w.r.t. m_k, and the precise value of $\ell_l(y_i, \mathbf{X}_i)$. Let us now study under which conditions a query Q_i^j can increase $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$, hence under which conditions $J_{Q_i^j}(m_k, m_l) = 1$. The two next propositions respectively address the case of imprecision w.r.t. m_k and m_l. Given a possible query Q_i^j on $\mathbf{X}_i$, the only possible way to increase $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$ is for the updated $\mathbf{X}_i'$ to become precise w.r.t. to the model for which $\mathbf{X}_i$ was imprecise, and moreover to be so that $\ell_l(y_i, \mathbf{X}_i') = 0$ ($\ell_k(y_i, \mathbf{X}_i') = 1$) if $\mathbf{X}_i$ is imprecise w.r.t. m_l (m_k).

Proposition 5. *Given $(\mathbf{X}_i, y_i)$ with $X_i^j = [a_i^j, b_i^j]$ and two models m_k and m_l s.t. $(\mathbf{X}_i, y_i)$ is imprecise w.r.t. m_l, the question Q_i^j is such that $J_{Q_i^j}(m_k, m_l) = 1$ if and only if one of the two following conditions holds*

$$y_i = 1 \text{ and } \underline{m}_l(\mathbf{X}_i) \geq -|w_l^j|(b_i^j - a_i^j) \tag{22}$$

or

$$y_i = -1 \text{ and } \overline{m}_l(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j). \tag{23}$$

Proof. First note that if $\mathbf{X}_i$ is imprecise w.r.t. m_l, then the only case where $\underline{\ell}_{k-l}(\mathbf{X}_i)$ increases is when the updated instance $\mathbf{X}_i^{'}$ is precise w.r.t. m_l after the query Q_i^j is performed and the precise loss becomes $\ell_l(y_i, \mathbf{X}_i^{'}) = 0$.

Let us consider the case $y_i = 1$ (the case $y_i = 0$ is similar). To have $\ell_l(y_i, \mathbf{X}_i^{'}) = 0$, we must have $\underline{m}_l(\mathbf{X}_i^{'}) \geq 0$. Using the same argument as in Proposition 3, we easily get the result. □

Proposition 6. *Given $(\mathbf{X}_i, y_i)$ with $X_i^j = [a_i^j, b_i^j]$ and two models m_k and m_l s.t. $(\mathbf{X}_i, y_i)$ is imprecise w.r.t. m_k, the query Q_i^j is such that $J_{Q_i^j}(m_k, m_l) = 1$ if and only if one of the two following condition holds*

$$y_i = 1 \text{ and } \overline{m}_k(\mathbf{X}_i) < |w_k^j|(b_i^j - a_i^j) \tag{24}$$

or

$$y_i = -1 \text{ and } \underline{m}_k(\mathbf{X}_i) \geq -|w_k^j|(b_i^j - a_i^j). \tag{25}$$

The proof is analogous to the one of Proposition 5. In summary, if $\mathbf{X}_i$ is imprecise w.r.t. only one model, estimating $J_{Q_i^j}(m_k, m_l)$ comes down to identify whether the $\mathbf{X}_i$ can become precise with respect to such a model, in such a way that the lower bound is possibly increased. Propositions 5 and 6 show that this can be checked easily using our previous results of Section 4.1 concerning the empirical risk. Actually, in this case, the problem essentially boils down to the problem of Section 4.2.

4.3.2. Imprecision with respect to both models

Given $\mathbf{X}_i$ and two models m_k, m_l, we define :

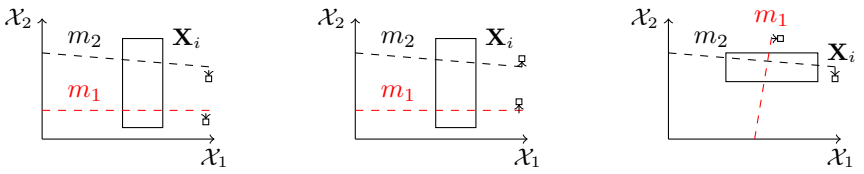
$$m_{k-l}(\mathbf{X_i}) = m_k(\mathbf{X}_i) - m_l(\mathbf{X}_i). \tag{26}$$

We thus have:

$$m_{k-l}(\mathbf{X_i}) > 0 \text{ if } m_k(\mathbf{x}_i) - m_l(\mathbf{x}_i) > 0 \quad \forall \mathbf{x}_i \in \mathbf{X}_i \tag{27}$$

$$m_{k-l}(\mathbf{X_i}) < 0 \text{ if } m_k(\mathbf{x}_i) - m_l(\mathbf{x}_i) < 0 \quad \forall \mathbf{x}_i \in \mathbf{X}_i. \tag{28}$$

In the other cases, this means that there are $\mathbf{x}_i^{'}, \mathbf{x}_i^{''} \in \mathbf{X}_i$ for which the model difference have different signs. The reason for introducing such differences is that, if $m_{k-l}(\mathbf{X_i}) > 0$ or $m_{k-l}(\mathbf{X_i}) < 0$, then not all combinations in $\{0,1\}^2$ are possible for the pair $(\ell_k(y_i, \mathbf{x}_i), \ell_l(y_i, \mathbf{x}_i))$, while they are in the other case. These various situations are depicted in Figure 3, where the white class is again the negative one ($y_i = -1$).



(a) $m_{1-2}(\mathbf{X_i}) > 0$ (b) $m_{1-2}(\mathbf{X_i}) < 0$ (c) Non-constant sign

Figure 3: Illustrations for the different possible cases corresponding to the difference $m_1(\mathbf{x}) - m_2(\mathbf{x})$

Since $m_k(\mathbf{x}_i) - m_l(\mathbf{x}_i)$ is also of linear form (with weights $w_k^j - w_l^j$), we can easily determine whether the sign of $m_{k-l}(\mathbf{X_i})$ is constant: it is sufficient to compute the interval

$$\left[\inf_{\mathbf{x}_i \in \mathbf{X}_i} (m_k(\mathbf{x}_i) - m_l(\mathbf{x}_i)), \sup_{\mathbf{x}_i \in \mathbf{X}_i} (m_k(\mathbf{x}_i) - m_l(\mathbf{x}_i))\right]$$

that can be computed similarly to $[\underline{m}_k(\mathbf{X}_i), \overline{m}_k(\mathbf{X}_i)]$ in Section 4.1 (Proposition 2). If zero is not within this interval, then $m_{k-l}(\mathbf{X_i}) > 0$ if the lower bound is positive, otherwise $m_{k-l}(\mathbf{X_i}) < 0$ if the upper bound is negative. The next proposition indicates how to easily compute the lower bound $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$ for the different possible situations.

Proposition 7. *Given $(\mathbf{X}_i, y_i)$ with $X_i^j = [a_i^j, b_i^j]$ and two models m_k, m_l s.t. $(\mathbf{X}_i, y_i)$ is imprecise w.r.t. both models, then the minimal difference value is*

$$\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = \begin{cases} \min(0, -y_i) & \text{if } m_{k-l}(\mathbf{X_i}) > 0 \\ \min(0, y_i) & \text{if } m_{k-l}(\mathbf{X_i}) < 0 \\ -1 & \text{if } m_{k-l}(\mathbf{X_i}) \text{ can take both signs} \end{cases} \tag{29}$$

Proof. First note that when neither $m_{k-l}(\mathbf{X_i}) > 0$ nor $m_{k-l}(\mathbf{X_i}) < 0$ hold, then there are values $\mathbf{x}_i$ for which $m_k(\mathbf{x}_i)$ and $m_l(\mathbf{x}_i)$ are either positive and negative, or negative and positive, or of the same sign. Hence there is always a value $\mathbf{x}_i$ such that $\ell_k(y_i, \mathbf{x}_i) = 0$ and $\ell_l(y_i, \mathbf{x}_i) = 1$.

Let us then deal with the situation where $m_{k-l}(\mathbf{X_i}) > 0$ (the case $m_{k-l}(\mathbf{X_i}) < 0$ can be treated similarly). In this case, there are values $\mathbf{x}_i \in \mathbf{X}_i$ such that $m_k(\mathbf{x}_i)$ and $m_l(\mathbf{x}_i)$ have the same sign (0/1 loss difference is then null), or $m_k(\mathbf{x}_i)$ is positive and $m_l(\mathbf{x}_i)$ negative, but no values for which $m_k(\mathbf{x}_i)$ is negative and $m_l(\mathbf{x}_i)$ positive. When $m_k(\mathbf{x}_i)$ is positive and $m_l(\mathbf{x}_i)$ negative, the loss difference is -1 if $y_i = +1$, and 1 if $y_i = -1$. □

The next question is to know under which conditions a query Q_i^j can increase $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$ (or equivalently $\underline{R}(m_{k-l})$), or in other words to determine a pair (i, j) s.t $J_{Q_i^j}(m_k, m_l) = 1$. Proposition 7 tells us that $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$ can be either 0 or -1 if $m_{k-l}(\mathbf{X_i}) > 0$ or $m_{k-l}(\mathbf{X_i}) < 0$, and is always -1 if $m_{k-l}(\mathbf{X_i})$ can take both signs. The next proposition establishes conditions under which $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$ can increase.

Proposition 8. *Given* $(\mathbf{X}_i, y_i)$ *with* $X_i^j = [a_i^j, b_i^j]$ *and two models* m_k *and* m_l *s.t* $(\mathbf{X}_i, y_i)$ *is imprecise w.r.t both of the given models, then* $J_{Q_i^j}(m_k, m_l) = 1$ *if the following conditions hold*

if $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = -1$ ***and*** $y_i = 1$***:***

$$\overline{m}_k(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j) \text{ or } \underline{m}_l(\mathbf{X}_i) \geq -|w_l^j|(b_i^j - a_i^j) \tag{30}$$

if $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = -1$ ***and*** $y_i = -1$***:***

$$\underline{m}_k(\mathbf{X}_i) \geq -|w_k^j|(b_i^j - a_i^j) \text{ or } \overline{m}_l(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j). \tag{31}$$

if $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = 0$ ***and*** $m_{k-l}(\mathbf{X_i}) < 0$***:***

$$\overline{m}_k(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j) \text{ and } \underline{m}_l(\mathbf{X}_i) \geq -|w_l^j|(b_i^j - a_i^j) \tag{32}$$

if $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = 0$ ***and*** $m_{k-l}(\mathbf{X_i}) > 0$***:***

$$\underline{m}_k(\mathbf{X}_i) \geq -|w_k^j|(b_i^j - a_i^j) \text{ and } \overline{m}_l(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j). \tag{33}$$

Proof. Let us first investigate the case where $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = -1$ and $y_i = 1$ (the case $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = -1$ and $y_i = -1$ is similar). In this case, $J_{Q_i^j}(m_k, m_l) = 1$ if and only if Q_i^j can either increase $\underline{\ell}_k(y_i, \mathbf{X}_i) = 0$ or decrease $\overline{\ell}_l(y_i, \mathbf{X}_i) = 1$, that is become precise for at least one of them, with $\underline{\ell}_k(y_i, \mathbf{X}_i^{'}) = 1$ or $\overline{\ell}_l(y_i, \mathbf{X}_i^{'}) = 0$. The conditions are then obtained by following arguments similar to those of Proposition 3.

The second case $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = 0$ only happens when either $m_{k-l}(\mathbf{X}_i) < 0$ or $m_{k-l}(\mathbf{X}_i) > 0$, and we will treat the first case. According to Proposition 7, this means that $y_i = -1$. Also, since according to Proposition 4 the value 0 is an upper bound of $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$ when $\mathbf{X}_i$ is imprecise with either m_k or m_l, to go from $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = 0$ to $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i^{'}) = 1$, we need a value $x_i^j \in X_i^j$ such that $m_k(\mathbf{X}_i^{'}) < 0$ and $m_l(\mathbf{X}_i^{'}) > 0$, as $y_i = -1$. Again, we can get the conditions to have such a value by deriving arguments similar to those of Proposition 3. □

For instance, in Figure 3(a) and 3(b), $J_{Q_i^1}(m_2, m_1) = 0$ and $J_{Q_i^2}(m_2, m_1) = 1$ for both cases. The whole procedure is summed up in the Algorithm 1. Algorithm 2 summarizes how to determine the query effect Q_i^j, which can be considered as the main computational difficulty when performing the querying step (line $2-3$ in Algorithm 1). Determining the set of undominated models (line $6-8$ in Algorithm 1) is summarized in Algorithm 3.

Let us now study the complexity of the whole approach. Lines 2 and 4 of Algorithm 2 are in $\mathcal{O}(p)$, since they correspond to linear operations. Iterations from 5-10 are in $\mathcal{O}(R \times p)$, since we must check all undominated models once. Iterations from 13-15 are also in $\mathcal{O}(R \times p)$, for the same reason. Thus, one run of Algorithm 2 is in $\mathcal{O}(R \times p)$. If we have I partial features in the data, then loop

2-3 of Algorithm 1 takes $\mathcal{O}(I \times R \times p)$ in the case of SVM, so it remains linear in each of the parameter. Algorithm 3 corresponds to lines 6-8 of Algorithm 1, and computing $\underline{R}(m_{k-l})$ can be done in $\mathcal{O}(n \times p)$ since we must compute $\underline{\ell}$ for each data point. Finally, since this must be done for every pair of models in the worst case, performing Algorithm 3 is in $\mathcal{O}(R^2 \times n \times p)$, which is quadratic in R and linear in the other parameters. This can be approximated by only comparing intervals $[\underline{R}(m_k), \overline{R}(m_k)]$ of every models, that would bring down the complexity to $\mathcal{O}(R \times n \times p)$, but would provide a super-set of the set of undominated models.

Algorithm 2: Determining the query effect $Value(Q_i^j)$

Input: partial data $(\mathbf{X}_i, y_i)$, set $\mathcal{M} = \{m_1, \ldots, m_R\}$ of models, the best potential model m_{k^*}

Output: the query effect $Value(Q_i^j)$

initialize $E_{Q_i^j}(m_{k^*}) = 0$, $J_{Q_i^j}(m_k, m_{k^*}) = 0$, $Value(Q_i^j) = 0$, $\forall k \neq k^*$;
check whether $(\mathbf{X}_i, y_i)$ is imprecise w.r.t m_{k^*} using Prop. 1 and 2;
if $(\mathbf{X}_i, y_i)$ *is imprecise w.r.t* m_{k^*} **then**
    compute $E_{Q_i^j}(m_{k^*})$ using Prop. 3 ;
    foreach $k \neq k^*$ **do**
        if $(\mathbf{X}_i, y_i)$ *is imprecise w.r.t* m_k **then**
            use Prop. 7 to get $\underline{\ell}_{k-k^*}(y_i, \mathbf{X}_i)$;
            use Prop. 8 to get $J_{Q_i^j}(m_k, m_{k^*})$;
        else
            use Prop. 5 to get $J_{Q_i^j}(m_k, m_{k^*})$;
    compute $Value(Q_i^j)$ using Definition 1;
else
    foreach $k \neq k^*$ **do**
        if $(\mathbf{X}_i, y_i)$ *is imprecise w.r.t* m_k **then**
            use Prop. 6 to get $J_{Q_i^j}(m_k, m_{k^*})$;
    compute $Value(Q_i^j)$ using Definition 1;

5. Application to binary SVM: set-valued labels

This section investigates the computations of racing algorithms to query set-valued labels when using binary SVM with precise features and when labels are partially given. Let us first note that, in the binary case, the problem of querying partial label data is identical to classical active learning as label data is either precise or fully partial (completely missing). One suitable technique in such a case is query-by-committee [17]. However, the strategies of query-by-committee technique and our racing technique are different. The previous one focus on

Algorithm 3: Determining the undominated set

Input: data $(\mathbf{X}_i, y_i)$, set $\mathcal{M} = \{m_1, \ldots, m_R\}$ of models
Output: the set of undominated model $\mathcal{M}^*$

foreach $k, l \in \{1, \ldots, R\} \times \{1, \ldots, R\}$, $k \neq l$ **do**
  $\underline{R}(m_{k-l}) = 0$;
  foreach *data* $(\mathbf{X}_i, y_i)$ **do**
    if $(\mathbf{X}_i, y_i)$ *is imprecise w.r.t both* m_k *and* m_l **then**
      use Prop. 7 to get $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$;
    else if $(\mathbf{X}_i, y_i)$ *is imprecise w.r.t only one of* m_k *and* m_l **then**
      use Prop. 4 to get $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$;
    else
      compute $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = \ell_k(y_i, \mathbf{X}_i) - \ell_l(y_i, \mathbf{X}_i)$ using (15)
    $\underline{R}(m_{k-l}) = \underline{R}(m_{k-l}) + \underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$;
  if $\underline{R}(m_{k-l}) > 0$ **then** remove m_k from $\{m_1, \ldots, m_R\}$;

missing labels that are the least consensual or the most ambiguous among a given set of models, while racing algorithms focus on labels having the most effect on reducing the uncertainty about the best potential model performance, as well as its difference to other models. From such intuitions, we could hope that, in practice, query-by-committee provide a quick reduction on the size of the set of undominated models while racing algorithms give faster convergence on determining the best potential model. In any case, it is worth exploring whether the two techniques perform similarly or if they show significant differences.

Before investigating the detailed computations of racing algorithms, let us recall that we focus here on binary SVM with 0/1 loss function (11). Also, as the output is partially given and inputs are precise, from now on and to facilitate exposure, we will adopt the notation $(\mathbf{x}_i, Y_i)$ where $Y_i \subseteq \{-1, 1\} = \mathcal{Y}$ and $\mathbf{x}_i \in \mathcal{X}$. Let us first note that, in case of precise label (i.e, $Y_i = \lambda$), it is clear that the corresponding loss score is precisely given as in (34) and querying such an instance is redundant.

$$\ell_l(Y_i, \mathbf{x}_i) = \underline{\ell}_l(Y_i, \mathbf{x}_i) = \overline{\ell}_l(Y_i, \mathbf{x}_i) = \begin{cases} 0 \text{ if } Y_i * m_l(\mathbf{x}_i) \geq 0, \\ 1 \text{ otherwise.} \end{cases} \tag{34}$$

We are now going to determine the imprecise loss function,

$$[\underline{\ell}_l(Y_i, \mathbf{x}_i), \overline{\ell}_l(Y_i, \mathbf{x}_i)]$$

and investigate under which conditions an imprecise label can have an effect on the risk bounds.

Proposition 9. *Given a model* m_l *and an instance* $(\mathbf{x}_i, Y_i)$*, if* $Y_i = \{-1, 1\}$*, then the following results hold*

A1. $[\underline{\ell}_l(Y_i, \mathbf{x}_i), \overline{\ell}_l(Y_i, \mathbf{x}_i)] = [0, 1]$

A2. $E_{Q_i}(m_l) = 1.$

Proof. It is clear that, in the binary case, if $Y_i = \{-1, 1\}$, whatever the prediction of the given model is (either 1 or -1), there always exist element λ and $\lambda^{'}$ in $\mathbf{y}_i$ s.t

$$\ell_l(\lambda, \mathbf{x}_i) = 0 \text{ and } \ell_l(\lambda^{'}, \mathbf{x}_i) = 1,$$

or in other words, $[\underline{\ell}_l(Y_i, \mathbf{x}_i), \overline{\ell}_l(Y_i, \mathbf{x}_i)] = [0, 1]$. Furthermore, querying Y_i always help to modify $[\underline{\ell}_l(Y_i, \mathbf{x}_i), \overline{\ell}_l(Y_i, \mathbf{x}_i)]$ into single value (either to 0 or 1). Or, in other words, A2 holds. □

Proposition 9 simply points out that all partial labels give the same (interval-valued) losses and have an effect on modifying the corresponding losses. In the next Proposition, we show that if the predictions of two given models for a partially labelled instance are different, then the corresponding lower pairwise difference is -1 and the effect of querying such labels is 1. Otherwise, both values are 0.

Proposition 10. *Given two models m_k and m_l and an imprecise instance $(\mathbf{x}_i, Y_i)$ ($Y_i = \{-1, 1\}$) then the following properties hold*

B1. *if $m_k(\mathbf{x}_i) = m_l(\mathbf{x}_i)$ then*

$$\underline{\ell}_{k-l}(Y_i, \mathbf{x}_i) = 0 \text{ and } J_{Q_i}(m_k, m_l) = 0.$$

B2. *if $m_k(\mathbf{x}_i) \neq m_l(\mathbf{x}_i)$ then*

$$\underline{\ell}_{k-l}(Y_i, \mathbf{x}_i) = -1 \text{ and } J_{Q_i}(m_k, m_l) = 1.$$

Proof. **B1** follows from the fact that if $m_k(\mathbf{x}_i) = m_l(\mathbf{x}_i)$, then $\ell_{k-l}(\lambda, \mathbf{x}_i) = 0$ for all $\lambda \in Y_i$. Furthermore, for any $\lambda^* \in Y_i$ to be returned after performing Q_i, we always have $\ell_{k-l}(\lambda^*, \mathbf{x}_i) = 0$, or in other words $J_{Q_i}(m_k, m_l) = 0$.

We are now going to give the proof for **B2**. Let us first notice that when $m_k(\mathbf{x}_i) \neq m_l(\mathbf{x}_i)$, there always exists $\lambda \in Y_i$ (i.e $\lambda = m_l(\mathbf{x}_i)$) s.t $\ell_{k-l}(\lambda) = -1$. Then it is clear that $\underline{\ell}_{k-l}(Y_i, \mathbf{x}_i) = -1$. Furthermore, if $\lambda^* = m_l(\mathbf{x}_i)$ is the given label after performing Q_i, then the pairwise difference $\ell^{Q_i}_{k-l}(\lambda^*, \mathbf{x}_i) = 1$. In other words, we have $J_{Q_i}(m_k, m_l) = 1$. □

Propositions 9 and 10 provide an interesting property of $Value(Q_i)$. In fact, for any given partial label Y_i, the corresponding total effect ($Value(Q_i)$) is exactly $1 + u_i$ where u_i is the number of models in the undominated set that give predictions against the best potential model (m^*). This means that while query-by-committee do consider consensus between all models for each instance, racing algorithms are based on the consensus of each model w.r.t. to the best potential model, for all instances. Again, we can see similarities and differences between the two approaches, and comparing them makes sense.

The whole procedure is again summed up in the Algorithm 1. Similar to the case of interval-valued features, we summarize how to determine the query effect Q_i (line $2-3$ in Algorithm 1) and the set of undominated models (line $6-8$ in Algorithm 1) in Algorithm 4 and 5, respectively. The complexity analysis is similar to the one of interval-valued features.

Algorithm 4: Determining the query effect $Value(Q_i)$

Input: partial data $(\mathbf{x}_i, Y_i)$ with $Y_i = \{-1, 1\}$, set $\mathcal{M} = \{m_1, \ldots, m_R\}$ of models, the best potential model m_{k^*}

Output: the query effect $Value(Q_i)$

initialize $E_{Q_i}(m_{k^*}) = 1$;
foreach $k \neq k^*$ **do**
    use Prop. 10 to get $J_{Q_i}(m_k, m_{k^*})$;
compute $Value(Q_i)$ using Definition 1;

Algorithm 5: Determining the undominated set

Input: data $(\mathbf{x}_i, Y_i)$, set $\mathcal{M} = \{m_1, \ldots, m_R\}$ of models

Output: the set of undominated model $\mathcal{M}^*$

foreach $k, l \in \{1, \ldots, R\} \times \{1, \ldots, R\}$, $k \neq l$ **do**
    $\underline{R}(m_{k-l}) = 0$;
    foreach *data* $(\mathbf{x}_i, Y_i)$ **do**
        if $(\mathbf{x}_i, Y_i)$ *is imprecise* **then**
            use Prop. 10 to get $\underline{\ell}_{k-l}(Y_i, \mathbf{x}_i)$;
        else
            compute $\underline{\ell}_{k-l}(Y_i, \mathbf{x}_i) = \ell_k(Y_i, \mathbf{x}_i) - \ell_l(Y_i, \mathbf{x}_i)$ using (34)
        $\underline{R}(m_{k-l}) = \underline{R}(m_{k-l}) + \underline{\ell}_{k-l}(Y_i, \mathbf{x}_i)$;
    if $\underline{R}(m_{k-l}) > 0$ **then** remove m_k from $\{m_1, \ldots, m_R\}$;

6. Experiments

We run experiments on a "contaminated" version of 7 standard benchmark (binary classes) data sets that are described in Table 1. The next two Sections present the details of the experiments and the results obtained in the two cases of interval-valued features and set-valued labels.

6.1. Interval-valued features case

Given a data set, we randomly chose a training set $\mathbf{D}$ consisting of 10% of instances and the rest (90%) as a test set $\mathbf{T}$. For each training instance $\mathbf{x}_i \in \mathbf{D}$, and each dimension $j = 1, ..., p$, a biased coin is flipped in order to decide whether or not x_i^j will be contaminated; the probability of contamination is α (α

Table 1: Data set used in the experiments

Name	# instances	# features
parkinsons	197	22
vertebral-column	310	6
ionosphere	351	34
climate-model	540	18
breast-cancer	569	30
blood-transfusion	784	4
banknote-authentication	1372	4

is fixed to 0.4 in all the experiments). Note that the probability that an instance has at least one contaminated feature is equal to $1 - 0.6^p$ (the complement of having no features contaminated), which is quite high: 0.87 when $p = 4$, our lowest number of features in any data set. In case x_i^j is contaminated, a width q_i^j will be generated from a uniform distribution. Then, the generated interval valued data is $X_i^j = [x_i^j + q_i^j(\underline{D}^j - x_i^j), x_i^j + q_i^j(\overline{D}^j - x_i^j)]$ where $\underline{D}^j = \min_i(x_i^j)$ and $\overline{D}^j = \max_i(x_i^j)$.

The set of undominated models is generated as follows: we randomly choose 100 precise replacements from the interval-valued training data. From each replacement, one linear SVM model is trained. The set of such 100 models is considered as the initial set $\mathcal{M}$ of undominated models.

After each query, the efficiency of the querying scheme is assessed based on the two following criteria:

- the proportion on the test set of identical predictions between the current best potential model and a reference model. The reference model is chosen to be the one in the initial undominated set that has the best accuracy on the fully precise training set. It is thus the model towards which the race must converge. The best potential model is the minimin model in the race. In case of multiple minimal risk models, the one with the minimum value of $\overline{R}(m)$ will be chosen as the best potential model;

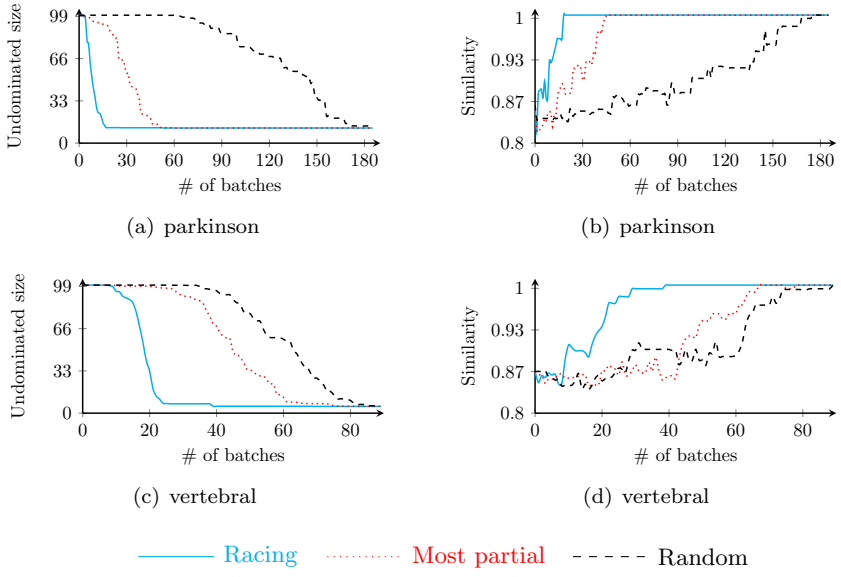
- the size of the undominated set.

To make comparisons about the convergence of the two criteria, two base-line algorithms are also used to query interval-valued features:

- a random querying strategy where, each time, an interval feature to be queried is chosen randomly;

- the most partial querying strategy i.e, each time, the feature with the largest imprecision (i.e., the largest sampled value q) is queried.

Because the training set is randomly chosen and contaminated, the results may be affected by random components. Then, for each data set, we repeat the above procedure 10 times and compute the average results.



(a) parkinson (b) parkinson

(c) vertebral (d) vertebral

Figure 4: Experiments for interval-valued features data with preferred model

6.2. Set-valued labels case

Experiments for the case of set-valued labels is performed in a similar way. Firstly, we randomly chose a training set $\mathbf{D}$ consisting of 20% of instances and the rest (of 80%) as a test set $\mathbf{T}$. Then, each label y_i in the training set $\mathbf{D}$ will be contaminated with probability α (α is fixed to 0.8 in all the experiments). Since the label is binary, if a label is contaminated, it becomes completely missing.

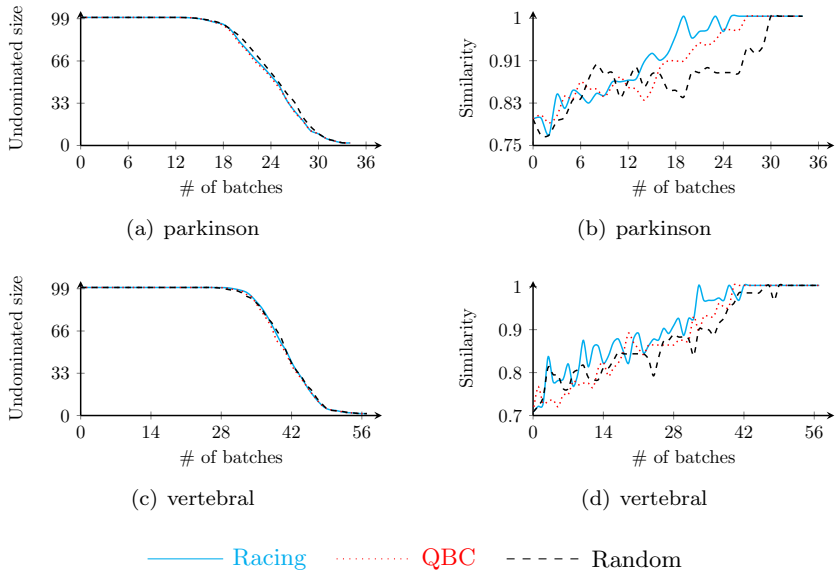
To make comparisons, the two following base-line querying schemes are also used:

- a random querying strategy, where, each time, a set-valued label is chosen randomly

- and a query-by-committee (QBC) strategy which picks up the instance with a set-valued label associated to the highest disagreement among the predictions given by the models in the race;

For each cases, we only show the results for two data sets (Parkinsons and Veretbral), as all data sets display similar behaviours. The experimental results for the case of interval-valued features and set-valued labels are given in Figures 4 and 5, respectively. The other results can be found in the Appendix.

In the case of set-valued labels, we can see that there are only slight differences between the methods. This result was expected, since, in the case of binary classification, partial labels are completely missing labels. Querying partial labels is thus equivalent to standard active learning methods like QBC. A lot of queries are needed to significantly reduce the set of undominated models



(a) parkinson (b) parkinson

(c) vertebral (d) vertebral

Figure 5: Experiments for set-valued labels data with preferred model

and to converge through the best model. Also, the random strategy has performances that are often comparable to the active learning ones. In contrast, the performances of our approach are much better than the others in the case of interval-valued features. One can see that the size of the set of undominated models is very quickly reduced and that our racing algorithm converges faster than the other approaches to the winning model.

It should be noted that the two previous sections provide an illustration of our approach to a particular learning method, i.e., binary SVM, but that the method can be applied in principle to any other learning method. Of course, whether or not the racing can be efficiently achieved or can improve quickly the prediction qualities vary from models to models, and can even depend on the aspects of data that are partial: in the case of binary SVM, our method is much more interesting when features are partial. We think it is however mainly due to two reasons: binary SVM are rather robust with respect to changes in the labels of data, as their learning rely only on a handful of precise points (the support vectors), and partial labels take a very restricted form (in contrast with partial features) that is equivalent to having missing labels. Therefore, what happens for SVM and labels will not necessarily happen for multi-class methods more sensitive to misspecified labels, such as decision trees.

In order to provide some insights about the potential difficulties of adapting our method to other models, the next section discuss briefly computational issues by building upon the results obtained for SVM.

7. Discussion on computational issues

The reader may have noticed that the section devoted to SVM with interval-valued features was quite longer, and presented more complex methods than the one about set-valued labels. Such an observation extends beyond SVM, and we try in this section to give some reasons why we may expect the problem of interval-valued features to be more complex than the problem of set-valued labels. As with the previous sections, we will stick to the case of $0-1$ loss functions. We will first provide some general remarks about the implementation of our generic approach, and then will shortly discuss how results obtained for the SVM case could be extended to monotone models in general.

7.1. General discussion

A first remark is that when we have a partial data $(\mathbf{X}_i, y_i)$ with interval-valued features, a query Q_i^j will not make the data precise unless only one feature is partial, but will transform $\mathbf{X}_i$ into $\mathbf{X}'_i = \times_{k\neq j} X_i^k \times x_i^j$. In contrast, querying a partial data $(\mathbf{x}_i, Y_i)$ with set-valued label Y_i guarantees that the queried data becomes the precise data $(\mathbf{x}_i, y_i)$, hence guaranteeing that the loss with respect to any model m_l will also become precise.

Let us now consider the problem of computing bounds of loss functions and potential effect of queries, with a focus on pairs of models and on the case where partial data will induce imprecision in the loss functions of both models, which constitute the most difficult aspects of our approach (our conclusions also apply to other calculations, yet these are typically easier to solve for both interval-valued features and set-valued labels).

Let us first consider the computations of $\underline{\ell}_{k-l}$: in the case of set-valued label Y_i, we do have

$$\underline{\ell}_{k-l}(Y_i, \mathbf{x}_i) = \begin{cases} 0 & \text{if } m_k(\mathbf{x}_i) = m_l(\mathbf{x}_i) \vee \{m_k(\mathbf{x}_i), m_l(\mathbf{x}_i)\} \cap Y_i = \emptyset \\ -1 & \text{else} \end{cases} \tag{35}$$

as the first case describes the only situations where we cannot find a label $\lambda \in Y_i$ such that $m_k(\mathbf{x}_i) = \lambda$ and $m_l(\mathbf{x}_i) \neq \lambda$. These conditions are rather easy to check in practice. In contrast, when one has interval-valued features, or more generally set-valued features $\mathbf{X}_i$ with a precise label y_i, we have that

$$\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = \begin{cases} 1 & \text{if } \forall \mathbf{x}_i \in \mathbf{X}_i, m_k(\mathbf{x}_i) \neq y_i \wedge m_l(\mathbf{x}_i) = y_i \\ -1 & \text{if } \exists \mathbf{x}_i \in \mathbf{X}_i \text{ s.t. } m_k(\mathbf{x}_i) = y_i \wedge m_l(\mathbf{x}_i) \neq y_i \\ 0 & \text{else} \end{cases} \tag{36}$$

with the last case corresponding to the situation where we can only find[2] $\mathbf{x}_i \in \mathbf{X}_i$ such that either $m_k(\mathbf{x}_i) = m_l(\mathbf{x}_i) = y_i$, or $m_k(\mathbf{x}_i) \neq y_i$ and $m_l(\mathbf{x}_i) \neq y_i$. In contrast with Equation (35) whose conditions are easily checked provided

[2] In addition to those possible $\mathbf{x}_i$ for which $m_k(\mathbf{x}_i) \neq y_i$ and $m_l(\mathbf{x}_i) = y_i$.

$m_k(\mathbf{x}_i)$ and $m_l(\mathbf{x}_i)$ are easy to compute (this is the greatest majority of model-based learning methods), identifying which case of Equation (36) does apply is more complex and highly depends on the properties of the considered learning method.

Similar conclusions can be drawn to compute the effect $J_{Q_i^j}(m_l, m_k)$ of a possible query. In the case of a set-valued label Y_i, we can directly extend the observation made in Proposition 10 for SVM to have that

$$J_{Q_i}(m_k, m_l) = 1 \text{ iff } \underline{\ell}_{k-l}(Y_i, \mathbf{x}_i) = -1$$

where $\underline{\ell}_{k-l}(Y_i, \mathbf{x}_i) = -1$ is given by the general and usually easy to estimate Equation (35). In contrast, we cannot extend Proposition 8 to arbitrary models when we have interval-valued features. Of course we still have that $J_{Q_i^j}(m_k, m_l) = 0$ when $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = 1$, as it cannot be increased by any query. Yet, in the other cases, one must check that the conditions to have an increase of $\underline{\ell}_{k-l}(y_i, \mathbf{X}_i)$ are met at least for one value $x_i^j \in X_i^j$, and we do not see how to provide a generic, efficient algorithmic procedure to check them without considering the specificities of the considered model.

7.2. The case of monotone models

In the case of the SVM methods, Proposition 7 uses the fact that linear functions are monotonic in every dimension $\mathcal{X}^j$. Note that our analysis should extend easily to all monotonic models, such as logistic regression or models based on the Choquet [18] and more generally on non-additive and fuzzy integrals [8].

As an illustration of this fact, let us consider the case of the logistic regression model. Keeping $\mathcal{X} = \mathbb{R}^p$ and the output space $\mathcal{Y} = \{-1, 1\}$ encoding the two possible classes, the logistic regression corresponding to a model m_k can be read[3] as

$$m_k(\mathbf{x}_i) = \ln \frac{P_k(1|\mathbf{x}_i)}{P_k(-1|\mathbf{x}_i)} = \sum_{j=0}^{p} w_k^j x_i^j,$$

with $P_k(.|\mathbf{x}_i)$ the posterior probabilities induced by model m_k, and vector $\mathbf{w}_k$ its parameters with the convention $x_i^0 = 1$. This model obviously shares with the SVM that it is monotone in each of its parameters, and in the case of the $0-1$ loss function, we also have

$$\ell_k(y_i, \mathbf{x}_i) = \begin{cases} 0 \text{ if } y_i \cdot m_k(\mathbf{x}_i) \geq 0 \\ 1 \text{ if } y_i \cdot m_k(\mathbf{x}_i) < 0. \end{cases} . \tag{37}$$

Indeed, if $m_k(\mathbf{x}_i) > 0$, we have $P_k(1|\mathbf{x}_i) \geq P_k(-1|\mathbf{x}_i)$, hence predicting $\hat{y}_i = 1$. If we consider now that the features $\mathbf{x}_i$ are imprecisely known (as said in the previous section, the major computational difficulties will mostly happen in the case of set-valued features), and that $X_i^j = [a_i^j, b_i^j]$ (note that we still have

[3] The adopted formulation allows us to better shows the similarities with the SVM case.

$X_i^0 = [1, 1]$), we can again easily determine when $(\mathbf{X}_i, y_i)$ will be imprecise (1) w.r.t. a model m_k and (2) w.r.t. both models m_k and m_ℓ. Clearly, for the first case, we will have

$$[\underline{m}_k(\mathbf{X}_i), \overline{m}_k(\mathbf{X}_i)] = \left[\sum_{w_k^j \geq 0} w_k^j b_i^j + \sum_{w_k^j < 0} w_k^j a_i^j, \sum_{w_k^j \geq 0} w_k^j a_i^j + \sum_{w_k^j < 0} w_k^j b_i^j \right],$$

and $(\mathbf{X}_i, y_i)$ will be imprecise w.r.t. m_k if and only if it contains the value 0 (arguments are similar to the one of the SVM case). Let us now consider the case of not one but two models m_k and m_ℓ, $(\mathbf{X}_i, y_i)$ being imprecise w.r.t. both of them (in the other situations, the same remarks as the one done for the SVM case apply). Without loss of generality, we can assume that $y_i = 1$, and we then have that

$$\underline{\ell}_{k-\ell}(y_i, \mathbf{X}_i) = \begin{cases} 1 & \text{if } \forall \mathbf{x}_i, m_k(\mathbf{x}_i) < 0 \wedge m_\ell(\mathbf{x}_i) > 0 \\ -1 & \text{if } \exists \mathbf{x}_i, m_k(\mathbf{x}_i) > 0 \wedge m_\ell(\mathbf{x}_i) < 0 \\ 0 & \text{else .} \end{cases}$$

It is clear that the first case will never happen, as $(\mathbf{X}_i, y_i)$ is imprecise w.r.t. m_k (so there is an $\mathbf{x}_i$ for which m_k is positive). To check the second condition, we have to know whether we can find $\mathbf{x}_i$ with $m_\ell(\mathbf{x}_i) < 0$, under the constraint that $m_k(\mathbf{x}_i) > 0$. This comes down to solve the following linear optimisation problem

$$\inf_{\substack{\mathbf{x}_i \in \mathbf{X}_i \\ m_k(\mathbf{x}_i) > 0}} \sum_{j=0}^{p} w_\ell^j x_i^j$$

and to check whether it is negative, in which case the lower bound is -1, and 0 otherwise. The methodology is here slightly different than in the SVM case, but still takes advantage of the monotonicity and linearity of the model. Completely implementing our proposal in the case of logistic regression would of course require some additional work (left here to the interested reader), but seems quite doable in the light of the above remarks.

8. Conclusion

This paper has explored an issue related to partially specified data: what is the best information to query so that an optimal model can be quickly derived. We have proposed a generic method, inspired from the idea of racing algorithms, to identify what partial data, feature or label should be queried (i.e., whose precise value should be obtained). The method search to differentiate, as soon as possible, different competing models. In principle, it can be applied to any learning method, but the computational complexity of applying it may vary between different learning methods, especially in the case of partially specified features, while the case of set-valued labels should present comparable complexities for most learning methods.

To illustrate this generic method, we have detailed its implementation for the specific case of binary SVM, and have performed various experiments to demonstrate the efficiency of our method. While it clearly outperformed other approaches in the case of partial features, demonstrating the potential usefulness of our approach in some cases, all tested approaches (including the random one) were comparable in the case of set-valued labels. However, it should be kept in mind that in the specific case of binary labels, learning and querying from partial data comes down to classical active learning. The picture may be quite different for multi-class problems.

Our future research efforts will mainly concentrate on applying this approach to various learning methods. Decision trees seem particularly interesting, as we are optimistic about the possibility to propose implementation that are computationally reasonable, and as those multi-class classifiers are well known to be highly sensitive to training data. This means that they could strongly benefit from our approaches. Logistic regression models, or their extension to non-linear functions [18] could also be explored, as in this case we can probably use the same monotonicity properties as in the SVM case.

References

[1] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[2] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 919–926. IEEE, 2009.

[3] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536, 2011.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[5] A. R. T. Donders, G. J. van der Heijden, T. Stijnen, and K. G. Moons. Review: a gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[6] M. Elahi, F. Ricci, and N. Rubens. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20:29–50, 2016.

[7] P. Fitzpatrick. *Advanced calculus*, volume 5. American Mathematical Soc., 2006.

[8] M. Grabisch and J.-M. Nicolas. Classification by fuzzy integral: Performance and tests. *Fuzzy sets and systems*, 65(2-3):255–271, 1994.

[9] V. Heidrich-Meisner and C. Igel. Hoeffding and bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 401–408. ACM, 2009.

[10] E. Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal on Approximate Reasoning*, 55(7):1519–1534, 2014.

[11] E. Hüllermeier and J. Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

[12] O. Maron and A. W. Moore. The racing algorithm: Model selection for lazy learners. In *Lazy learning*, pages 193–225. Springer, 1997.

[13] V.-L. Nguyen, S. Destercke, and M.-H. Masson. Partial data querying through racing algorithms. In *Integrated Uncertainty in Knowledge Modelling and Decision Making: 5th International Symposium, IUKM 2016*, pages 163–174. Springer, 2016.

[14] F. Olsson. A literature survey of active machine learning in the context of natural language processing. Technical report t2009:06, Swedish Institute of Computer Science, 2009.

[15] M. Raman-Sundström. A pedagogical history of compactness. *The American Mathematical Monthly*, 122(7):619–635, 2015.

[16] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[17] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[18] A. F. Tehrani, W. Cheng, K. Dembczyński, and E. Hüllermeier. Learning monotone nonlinear models using the choquet integral. *Machine Learning*, 89(1-2):183–211, 2012.

[19] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2002.

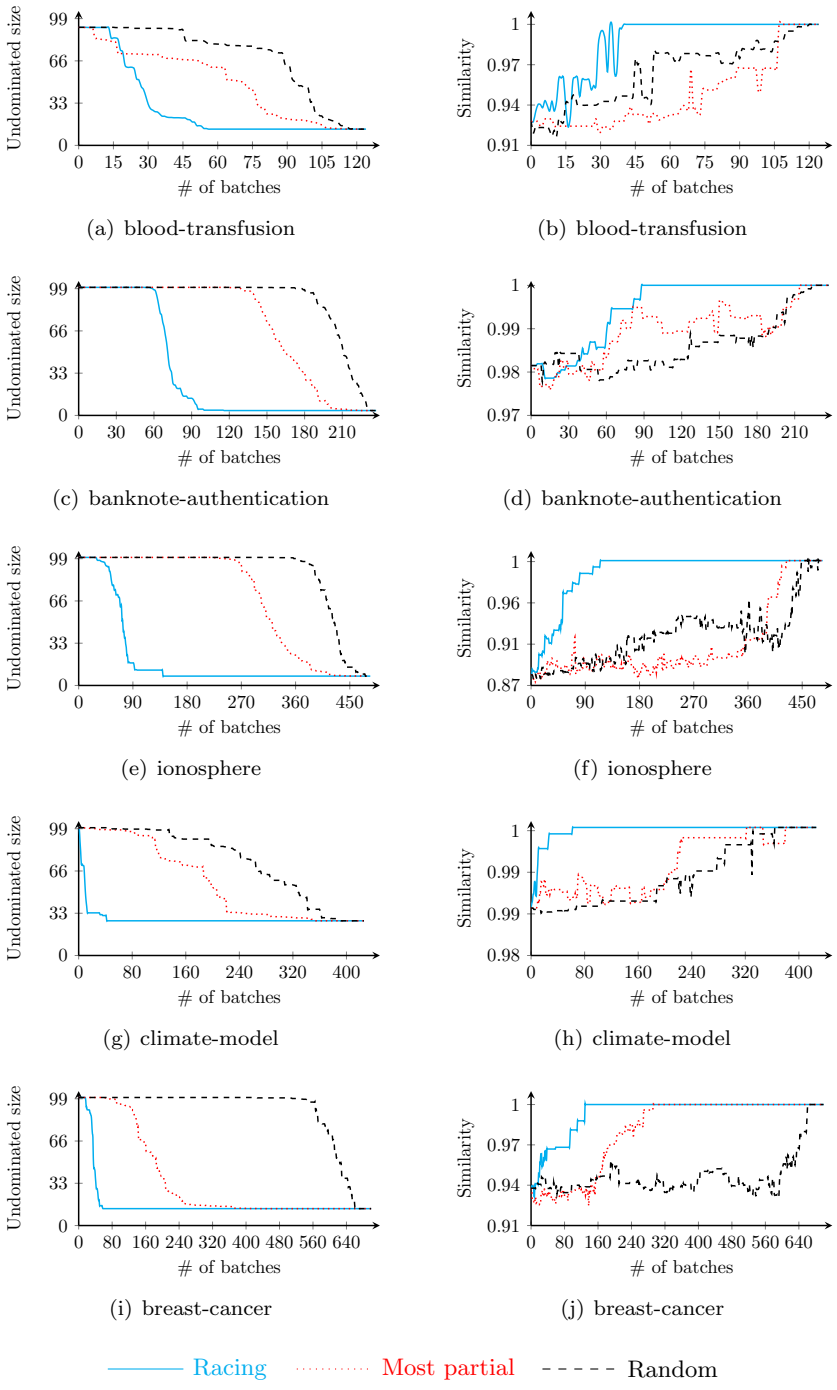
[20] M. C. Troffaes. Decision making under uncertainty using imprecise probabilities. *International Journal of Approximate Reasoning*, 45(1):17–29, 2007.

[21] L. V. Utkin and F. P. Coolen. Classification with support vector machines and kolmogorov–smirnov bounds. *Journal of Statistical Theory and Practice*, 8(2):297–318, 2014.

[22] A. Wiencierz and M. Cattaneo. On the validity of minimin and minimax methods for support vector regression with interval data. In *9th international symposium on imprecise probability: Theories and applications*, pages 325–332, 2015.

[23] Z. Ye, P. Liu, J. Liu, X. Tang, and W. Zhao. Practice makes perfect: An adaptive active learning framework for image classification. *Neurocomputing*, 196:95–106, 2016.

Appendix A. Experimental results



(a) blood-transfusion (b) blood-transfusion

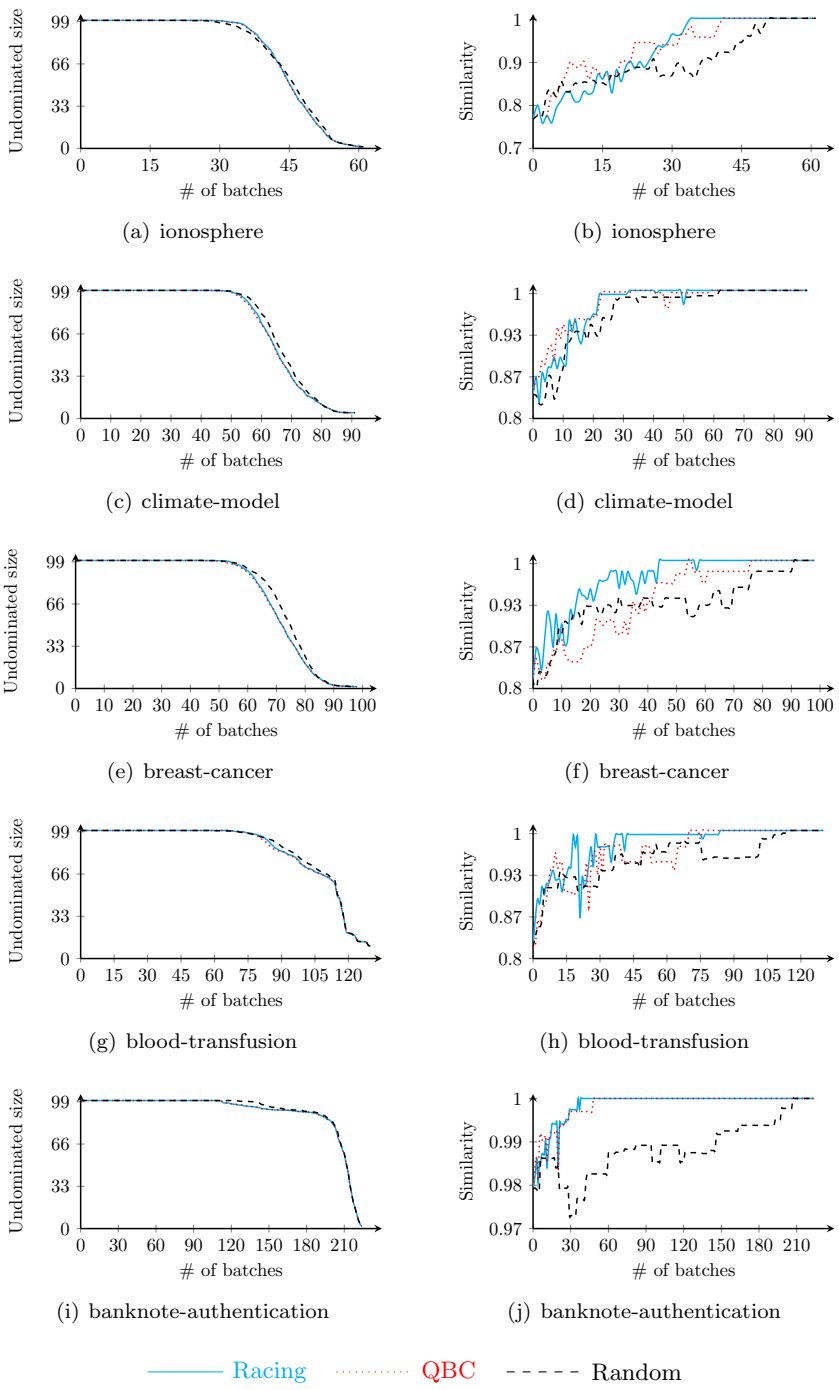
(c) banknote-authentication (d) banknote-authentication

(e) ionosphere (f) ionosphere

(g) climate-model (h) climate-model

(i) breast-cancer (j) breast-cancer

Figure A.6: Experiments for interval-valued features data with preferred model



(a) ionosphere (b) ionosphere

(c) climate-model (d) climate-model

(e) breast-cancer (f) breast-cancer

(g) blood-transfusion (h) blood-transfusion

(i) banknote-authentication (j) banknote-authentication

Figure A.7: Experiments for set-valued labels data with preferred model