Partial Data Querying Through Racing Algorithms

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

Partial data querying through racing algorithms

Vu-Linh Nguyen[1], Sébastien Destercke[1], and Marie-Hélène Masson[1,2]

[1] UMR CNRS 7253 Heudiasyc,
Sorbonne Université, Université de Technologie de Compiègne CS 60319 - 60203
Compiègne cedex, France linh.nguyen@hds.utc.fr
[2] Université de Picardie Jules Verne, France

Abstract. This paper studies the problem of learning from instances characterized by imprecise features or imprecise class labels. Our work is in the line of active learning, since we consider that the precise value of some partial data can be queried to reduce the uncertainty in the learning process. Our work is based on the concept of racing algorithms in which several models are competing. The idea is to identify the query that will help the most to quickly decide the winning model in the competition. After discussing and formalizing the general ideas of our approach, we study the particular case of binary SVM and give the results of some preliminary experiments.

Keywords: partial data, data querying, active learning, racing algorithms

1 Introduction

Although classical learning schemes assume that every instance is fully specified, there are many cases where such an assumption is unlikely to hold, and where some features or the label (class) of an instance may be only partially known. The problem of learning from imprecise data has gained an increasing interest with applications in different fields such as image or natural language processing [2, 3, 4]. The imprecision of the data leads to uncertainties in the learning process and in the decision making.

This work explores an issue related to partially specified 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 is known as active learning and has already been largely treated [6]. However, we are not aware of such works for partial data. The present proposal is based on the concept of racing algorithms [5], initially used to select an optimal configuration of a given lazy learning model, and since then applied to other settings such as multi-armed bandits. 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 composed of different possible models. As the data are partial, the performance of each model is uncertain

(i.e. interval-valued) and several candidate models can be optimal. The race will consist in iteratively making queries, i.e., in asking to an oracle the precise value of a partial data. The key question is then to identify those queries that will help the most to reduce the set of possible winners in the race and to converge quickly to the optimal model. We illustrate this general approach using binary SVM classifiers.

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. Section 4 is focused on the particular case of binary SVM. Finally, some experiments in Section 5 demonstrate the effectiveness of our proposals.

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 using n input/output samples $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are respectively the input and the output spaces[3]. 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 is to assume that a model m_j is said to be better than m_k (denoted $m_j \succ m_k$) if

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

or in other words if the risk of m_j is lower than the risk of m_k.

In this work, we are however interested in the case where data are partial, that is where general samples are of the kind $(\mathbf{X}_i, Y_i) \subseteq \mathcal{X} \times \mathcal{Y}$. In such a case, Equations (1), (2) and (3) are no longer well-defined, and there are different ways to extend them. Two of the most common ways to extend them is either to use a minimin (optimistic) or a maximin (pessimistic) approach [7]. That is, if we extend Equation (1) to a lower bound

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

[3] 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. 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. However, such an approach, due to its conservative nature, may lead to sub-optimal model.

In this paper, we are interested into another kind of approach, where we do not search for a unique optimal model but rather consider sets of potentially optimal models. In this case, we can say that a model m_j is better than m_k (still denoted $m_j \succ m_k$) if

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

which is a direct extension of Equation (3). That is, $m_j \succ m_k$ if and only if it is better under every possible precise instances $(\mathbf{x}_i, y_i)$ 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 $\succ$.

Example 1. Figure 1 illustrates a situation where $\mathcal{Y}$ consists of two different classes (gray and white), and $\mathcal{X}$ of two dimensions. Only imprecise data are numbered. Squares are assumed to have precise features. Stripped squares have unknown labels. Assuming that $\mathcal{M} = \{m_1, m_2\}$ (the models could be decision stumps, or one-level decision trees), we would have that $m_2 = m^*_{Mm}$ is the maximin model and $m_1 = m^*_{mm}$ the minimin one. The two models would however be incomparable according to (6), hence $\mathcal{M}^* = \mathcal{M}$ in this case.

3 Partial data querying: a racing approach

Both the minimin and maximin approaches have the same goal: obtaining a unique model from partially specified data. The idea we consider in this paper

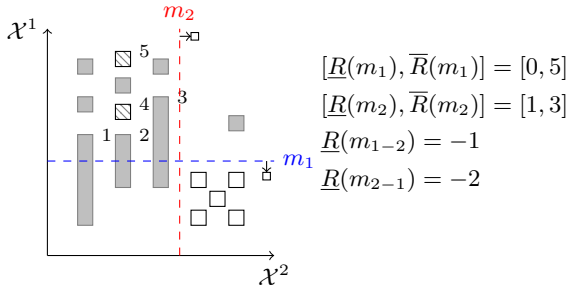


Fig. 1. Illustration of partial data and competing models

is different. We want to identify and query those data that will help the most to reduce the set $\mathcal{M}^*$. Whether an information is useful for the race is formalized in what follows. Let us 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.

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. 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_k and a data $(\times_{j=1}^P X_i^j, Y_i)$, the result of a query can have an effect on the interval $[\underline{R}(m_k), \overline{R}(m_k)]$, depending on whether it changes the interval $[\underline{\ell}(Y_i, m_k(\mathbf{X}_i)), \overline{\ell}(Y_i, m_k(\mathbf{X}_i))]$. Similarly, when assessing whether the model m_k is preferred to m_ℓ, the query can have an influence on the value $\underline{R}(m_{\ell-k})$ or not. This can be formalized 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:

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

and

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

Of course, when $j = p + 1$, X_i^j is to be replaced by Y_i. $E_{Q_i^j}$ simply tells us whether or not the query can affect our evaluation of m_k performances, while $J_{Q_i^j}(m_k, m_\ell)$ informs us whether the query can help to differentiate m_k and m_ℓ.

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 3) and to Q_1^1 (the first feature of point 3). For these two questions, we have

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

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

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_1^1 can actually lead to declare m_2 better than m_1).

The effect of a query being now formalized, we can propose a method inspired by racing algorithms to select the best query. An initial set of models can be created by sampling several times a precise data set $(\mathbf{x}_i, y_i) \in (\mathbf{X}_i, Y_i)$ and then learning several optimal models according to this selection. Algorithm 1 summarises the general procedure applied to find the best query and to update the race. 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. In the next sections, we illustrate our proposed setting with the popular SVM algorithm.

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

$k^* = \arg\min_{k \in \{1,\ldots,R\}} \underline{R}(m_i)$;
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 $k, \ell \in \{1, \ldots, R\} \times \{1, \ldots, R\}$, $k \neq \ell$ **do**
  Compute $\underline{R}(m_{\ell - k})$;
  if $\underline{R}(m_{\ell - k}) > 0$ **then** remove m_ℓ from $\{m_1, \ldots, m_R\}$;

4 Application to binary SVM

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 $m_k = (\mathbf{w}_k, c_k)$ corresponds to the "maximum-margin" hyperplane $\mathbf{w}_k \mathbf{x} + c_k$ with $\mathbf{w}_k \in \mathbb{R}^p$ and $c_k \in \mathbb{R}$. For convenience sake, we will use $(\mathbf{w}_k, c_k)$ and m_k interchangeably from now on. We will also focus on the case of imprecise features but precise labels, and will denote y_i the label of training instances for short, instead of Y_i. We will also focus on the classical $0-1$ loss function defined as follows for an instance $(\mathbf{x}_i, y_i)$:

$$\ell(y_i, m_k(\mathbf{x}_i)) = \begin{cases} 0 \text{ if } y_i \cdot m_k(\mathbf{x}_i) \geq 0 \\ 1 \text{ if } y_i \cdot m_k(\mathbf{x}_i) < 0, \end{cases} \tag{10}$$

where $m_k(\mathbf{x}_i) = \mathbf{w}_k\mathbf{x}_i + c_k$.

4.1 Instances inducing imprecision in empirical risk

Before entering into the details of how risk bounds (4)-(6) and query effects (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 $\underline{\ell}(y_i, m_k(\mathbf{X}_i)) = \overline{\ell}(y_i, m_k(\mathbf{X}_i)) = \ell(y_i, m_k(\mathbf{X}_i))$, then $E_{Q_i^j}(m_k) = J_{Q_i^j}(m_k, m_l) = 0$ for all $j = 1, \ldots, P$

Definition 1. *Given a SVM model m_k, an instance $(\mathbf{X}_i, y_i)$ is called an imprecise instance w.r.t m_k (or shortly, imprecise instance when m_k is fixed) if and only if*

$$\exists \mathbf{x}_i^{'}, \mathbf{x}_i^{''} \in \mathbf{X}_i \ s.t \ m_k(\mathbf{x}_i^{'}) \geq 0 \ and \ m_k(\mathbf{x}_i^{''}) < 0. \tag{11}$$

Instances that do not satisfy Definition 1 will be called precise instances (w.r.t m_k). Being precise means that the sign of $m_k(x_i)$ is the same for all $x_i \in \mathbf{X}_i$, which implies that the loss $\underline{\ell}(y_i, m_k(\mathbf{X}_i)) = \overline{\ell}(y_i, m_k(\mathbf{X}_i))$ 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.

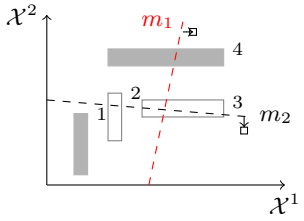


Fig. 2. Illustration of interval-valued instances

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

$$\underline{m}_k(\mathbf{X}_i) := \inf_{\mathbf{x}_i \in \mathbf{X}_i} m_k(\mathbf{x}_i) \text{ and } \overline{m}_k(\mathbf{X}_i) := \sup_{\mathbf{x}_i \in \mathbf{X}_i} m_k(\mathbf{x}_i) \tag{12}$$

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

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

$$\underline{m}_k(\mathbf{X}_i) < 0 \text{ and } \overline{m}_k(\mathbf{X}_i) \geq 0. \tag{13}$$

Furthermore, we have that the hyperplane $m_k(\mathbf{x}_i) = 0$ *intersects with the region* $\mathbf{X}_i$ *if and only if* (13) *holds. In other words,* $\exists \mathbf{x}_i \in \mathbf{X}_i$ *s.t.* $m_k(\mathbf{x}_i) = 0$*.*

This proposition means that to determine whether an instance $(\mathbf{X}_i, y_i)$ is imprecise, we only need to compute values $\underline{m}_k(\mathbf{X}_i)$ and $\overline{m}_k(\mathbf{X}_i)$, which can be easily done using Proposition 2. Note that, due to a lack of space, the proofs of proposition are omitted in this conference version.

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

$$\overline{m}_k(\mathbf{X}_i) = \sum_{w_k^j \geq 0} w_k^j b_i^j + \sum_{w_k^j < 0} w_k^j a_i^j + c_k$$
$$\underline{m}_k(\mathbf{X}_i) = \sum_{w_k^j \geq 0} w_k^j a_i^j + \sum_{w_k^j < 0} w_k^j b_i^j + c_k.$$

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

4.2 Empirical risk bounds and single effect

We are now going to investigate the practical computation of $\underline{R}(m_k)$, $\overline{R}(m_k)$ for $k = 1, \ldots, M$, as well as the value $E_{Q_i^j}(m_k)$ of a query on a model m_k. Equations (4) (resp. (5)) implies that the computation of $\underline{R}(m_k)$ (resp. $\overline{R}(m_k)$) can be done by computing $\underline{\ell}(y_i, m_k(\mathbf{x}_i))$ (resp. $\overline{\ell}(y_i, m_k(\mathbf{x}_i))$) for $i = 1, \ldots, n$ and then by summing the obtained values, therefore we can focus on computing $\underline{\ell}(y_i, m_k(\mathbf{x}_i)$ and $\overline{\ell}(y_i, m_k(\mathbf{x}_i)$ for a single instance. Similarly, $E_{Q_i^j}(m_k) = 1$ only when the interval $[\underline{\ell}(y_i, m_k(\mathbf{x}_i), \overline{\ell}(y_i, m_k(\mathbf{x}_i)]$ can be modified by querying X_i^j, therefore we can also focus on a single instance to evaluate it. When $\mathbf{X}_i$ is imprecise w.r.t. m_k, we have $\underline{\ell}(y_i, m_k(\mathbf{X}_i)) = 0$ and $\overline{\ell}(y_i, m_k(\mathbf{X}_i)) = 1$. Let us now consider the problem of computing, for a query Q_i^j, the effect $E_{Q_i^j}(m_k)$ it can have on the empirical risk bounds. In the case of 0-1 loss, the only case where $E_{Q_i^j}(m_k) = 1$ is the one where $[\underline{\ell}(y_i, m_k(\mathbf{x}_i)), \overline{\ell}(y_i, m_k(\mathbf{x}_i))]$ 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_k. According to Proposition 1, this means that either $\underline{m}_k(\mathbf{X}_i^{'})$ should become positive, or $\overline{m}_k(\mathbf{X}_i^{'})$ should become negative after query Q_i^j. This is formalised 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 model* $m_k(\mathbf{x}_i)$ *s.t.* $\mathbf{X}_i$ *is imprecise, then* $E_{Q_i^j}(m) = 1$ *if and only if one of the following conditions holds*

$$\underline{m}_k(\mathbf{X}_i) \geq -|w_k^j|(b_i^j - a_i^j) \tag{14}$$

or

$$\overline{m}_k(\mathbf{X}_i) < |w_k^j|(b_i^j - a_i^j). \tag{15}$$

$\underline{R}(m_k), \overline{R}(m_k)$, needed in the line 1 of Algorithm 1 to identify the most promising model k^*, are computed easily using the values of $\underline{\ell}(y_i, m_k(\mathbf{X}_i)) = 0$ and $\overline{\ell}(y_i, m_k(\mathbf{X}_i)) = 1$, while Equations (14)-(15) provide us easy ways to estimate the values of $E_{Q_i^j}(m_{k^*})$, needed in line 3 of Algorithm 1.

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 $J_{Q_i^j}(m_l, m_k)$ of a possible effect of Q_i^j. To do that, note that $\underline{R}(m_{k-l})$ can be rewritten as

$$\underline{R}(m_{k-l}) = \inf_{\mathbf{x}_i \in \mathbf{X}_i, i=1,\ldots,n} (R(m_k) - R(m_l)) = \sum_{i=1}^{n} \underline{\ell}_{k-l}(\mathbf{x}_i, y_i) \tag{16}$$

with

$$\underline{\ell}_{k-l}(y_i, \mathbf{X}_i) = \inf_{\mathbf{x}_i \in \mathbf{X}_i} \Big(\ell(y_i, m_k(\mathbf{x}_i)) - \ell(y_i, m_l(\mathbf{x}_i) \Big), \tag{17}$$

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 $\underline{R}(m_k)$ and $\overline{R}(m_k)$. Also, $J_{Q_i^j}(m_l, m_k) = 1$ 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 $\ell(y_i, m_k(\mathbf{X}_i)) - \ell(y_i, m_l(\mathbf{X}_i)$ is a well-defined value, as each loss is precise, and in this case $J_{Q_i^j}(m_l, m_k) = 0$. 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 will proceed to the more complex case where it is imprecise w.r.t. both. Note that imprecision with respect to each model can be easily established using Proposition 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* $m_k(\mathbf{X}_i)$ *and* $m_l(\mathbf{X}_i)$ *s.t* $(\mathbf{X}_i, y_i)$ *is imprecise w.r.t. one and only one model, then we have*

$$\underline{\ell}_{k-l}(\mathbf{X}_i) = \ell(y_i, m_k(\mathbf{X}_i)) - 1 \qquad \textit{if } \mathbf{X}_i \textit{is imprecise w.r.t. } m_l \tag{18}$$

$$\underline{\ell}_{k-l}(\mathbf{X}_i) = -\ell(y_i, m_l(\mathbf{X}_i)) \qquad \textit{if } \mathbf{X}_i \textit{is imprecise w.r.t. } m_k. \tag{19}$$

Let us now study under which conditions a query Q_i^j can increase $\underline{\ell}_{k-l}(\mathbf{X}_i)$, hence under which conditions $J_{Q_i^j}(m_l, m_k) = 1$. The two next propositions respectively address the case of imprecision w.r.t. m_l and m_k. Given a possible query Q_i^j on $\mathbf{X}_i$, the only possible way to increase $\underline{\ell}_{k-l}(\mathbf{X}_i)$ 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 $\ell(y_i, m_l(\mathbf{X}_i^{'})) = 0$ ($\ell(y_i, m_k(\mathbf{X}_i^{'})) = 1$) 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* $m_k(\mathbf{x}_i)$ *and* $m_l(\mathbf{x}_i)$ *s.t.* $(\mathbf{X}_i, y_i)$ *is imprecise w.r.t.* m_l*, the question* Q_i^j *is such that* $J_{Q_i^j}(m_l, m_k) = 1$ *if and only if one of the two following condition holds*

$$y_i = 1 \textit{ and } \underline{m}_l(\mathbf{X}_i) \geq -|w_l^j|(b_i^j - a_i^j) \tag{20}$$

or

$$y_i = -1 \textit{ and } \overline{m}_l(\mathbf{X}_i) < |w_k^j|(b_i^j - a_i^j). \tag{21}$$

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

$$y_i = 1 \textit{ and } \overline{m}_k(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j) \tag{22}$$

or

$$y_i = -1 \textit{ and } \underline{m}_k(\mathbf{X}_i) \geq -|w_k^j|(b_i^j - a_i^j). \tag{23}$$

In summary, if $\mathbf{X}_i$ is imprecise w.r.t. only one model, estimating $J_{Q_i^j}(m_l, m_k)$ 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 done easily using our previous results of Section 4.1 concerning the empirical risk.

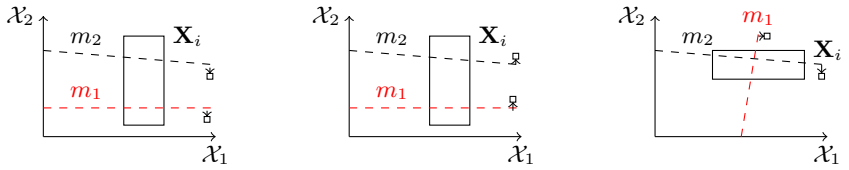
Imprecision with respect to both models Given $\mathbf{X}_i$ and two models m_k, m_l, we will adopt the following notations:

$$m_{k-l}(\mathbf{X_i}) > 0 \text{ if } m_k(\mathbf{x}_i) - m_l(\mathbf{x}_i) > 0 \quad \forall \mathbf{x}_i \in \mathbf{X}_i \tag{24}$$

$$m_{k-l}(\mathbf{X_i}) < 0 \text{ if } m_k(\mathbf{x}_i) - m_l(\mathbf{x}_i) < 0 \quad \forall \mathbf{x}_i \in \mathbf{X}_i. \tag{25}$$

In the other cases, this means that there are $x_i^{'}, x_i^{''} \in \mathbf{X}_i$ 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 $(\ell(y_i, m_k(\mathbf{x}_i), \ell(y_i, m_l(\mathbf{x}_i))$, 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

Fig. 3. Illustrations for the different possible cases corresponding to the difference $m_1(x) - m_2(x)$

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. 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 $\underline{\ell}_{k-l}(\mathbf{X}_i)$ 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 the given models, then the minimal difference value is*

$$\underline{\ell}_{k-l}(\mathbf{X}_i) = \begin{cases} \min(0, -y_i) & \text{if } m_{k-l}(\mathbf{X}_i) > 0 \\ \min(0, y_i) & \text{if } m_{k-l}(\mathbf{X}_i) < 0 \\ -1 & \text{else} \end{cases} \tag{26}$$

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

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

if $\underline{\ell}_{k-l}(X_i) = -1$ ***and*** $y_i = 1$***:***

$$\overline{m}_k(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j) \text{ or } \underline{m}_l(\mathbf{X}_i) \geq -|w_l^j|(b_i^j - a_i^j) \tag{27}$$

if $\underline{\ell}_{k-l}(X_i) = -1$ ***and*** $y_i = -1$***:***

$$\underline{m}_k(\mathbf{X}_i) \geq -|w_k^j|(b_i^j - a_i^j) \text{ or } \overline{m}_l(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j). \tag{28}$$

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

$$\overline{m}_k(\mathbf{X}_i) < |w_l^j|(b_i^j - a_i^j) \text{ and } \underline{m}_l(\mathbf{X}_i) \geq -|w_l^j|(b_i^j - a_i^j) \tag{29}$$

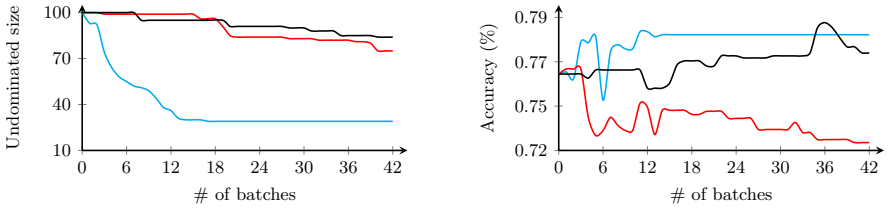
if $\underline{\ell}_{k-l}(X_i) = 0$ ***and*** $m_{k-l}(\mathbf{X_i}) > 0$***:***

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

For instance, in Figure 3.(a) and 3.(b), $J_{Q_i^1}(m_1, m_2) = 0$ for both cases, while $J_{Q_i^2}(m_1, m_2) = 0$ for 3.(a) and $J_{Q_i^2}(m_1, m_2) = 1$ for 3.(a)

5 Experiments

To demonstrate the usefullness of our approach, we run experiments on a "contaminated" version of a standard benchmark data set, namely Parkinson (195 instances, 22 features, binary labels) which contains precise features and labels. 10% of data have been used for training and 90% for testing, since querying partial data is most useful when only few data are available. For each feature x_i^j in the training set, a biased coin is flipped in order to decide whether or not this example will be contaminated; the probability of contamination is $p = 0.4$. 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 $\overline{D}^j = \max_i(x_i^j)$. To evaluate the efficiency of our proposal, we query interval data using three approaches : our racing algorithm, a random querying strategy (each time, interval examples will be chosen randomly) and the most partial querying strategy (each time, examples with the largest imprecision will be queried). Firstly, we randomly generate 100 completions of interval-valued data. From each completion, one linear SVM model is trained and the set of such SVM models is considered as the initial set of undominated models. To limit the computational cost, at each iteration of the racing algorithm, we choose to perform 2 queries (batch query) instead of only one. After each batch, we discard the dominated models and determine the best potential model. In case of multiple minimal risk models, the one with minimum value of $\overline{R}_m$ will be chosen as the best potential model. The accuracy of the best potential model is computed on the test set. The learning process is repeated 10 times and the average size of the sets of models and the average accuracy of the best potential model are given in Figure 4(a) and Figure4(b), respectively. The experimental results show that, using our approach, the size of the undominated set can be quickly reduced and that the accuracy of the best potential model converges very fast to the one obtained when knowing all precise data, while the reduction of the size of the set and the convergence of the accuracy is slower and less stable for other querying strategies.



(a) size of the set of undominated models

(b) Acc. of the best potential model

Fig. 4. Results of the experiments

6 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 determined. We have proposed to use a racing algorithms approach in which several models are competing and some of them are discarded as long as new precise information become available. These general concepts have been illustrated in the case of binary SVM and the first experiments have shown the interest of the method. Future works will focus on the case of interval-valued features and set-valued labels.

References

1. 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. E. Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal on Approximate Reasoning*, 55(7):1519–1534, 2014.
5. O. Maron and A. W. Moore. The racing algorithm: Model selection for lazy learners. In *Lazy learning*, pages 193–225. Springer, 1997.
6. B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
7. M. Troffaes. Decision making under uncertainty using imprecise probabilities. *International Journal of Approximate Reasoning*, 45(1):17–29, 2007.