Active Learning for Classification: An Optimistic Approach

Timothé Collet
Supelec, MaLIS Research group, France
GeorgiaTech-CNRS UMI 2958, France
Email : timothe.collet@supelec.fr

Olivier Pietquin
University Lille 1, France
LIFL (UMR 8022 CNRS / Lille 1), Sequel Team, France
IUF (Institut Universitaire de France)
Email : olivier.pietquin@univ-lille1.fr

Abstract—In this paper, we propose to reformulate the active learning problem occurring in classification as a sequential decision making problem. We particularly focus on the problem of dynamically allocating a fixed budget of samples. This raises the problem of the trade off between exploration and exploitation which is traditionally addressed in the framework of the multiarmed bandits theory. Based on previous work on bandit theory applied to active learning for regression, we introduce four novel algorithms for solving the online allocation of the budget in a classification problem. Experiments on a generic classification problem demonstrate that these new algorithms compare positively to state-of-the-art methods.

I. INTRODUCTION

We consider a supervised learning framework, in the context of a noisy two-class classification problem. Our work focuses on active learning, which is the process of choosing what instances to label in order to minimize the number of queries to an oracle. In an on-line context, an active learning algorithm successively chooses the best instance to present to the oracle guided by the information from all the previous queries. Thus, this is a sequential decision process [1].

The noisy aspect of this problem makes some instances more or less difficult to classify. —which may come from an intrinsic noise on the data or from an inability of the classifier to distinguish instances— and the fact that the noise is not the same for all instances. Indeed, it is relatively intuitive that little effort must be put into the least noisy instances as they are easy to classify. While it is less intuitive that little effort must also be put into very noisy instances too. Thus, this is an online allocation problem with respect to the noise value.

We can represent this problem using a multi-armed bandit setting, as introduced in [2] and surveyed in [3]. Although the degree of noise of an instance affects whether it will be chosen for a query, the noise is initially unknown. It can, however, be learned while we present instances to the oracle. We therefore have to make a trade-off between learning the true noise of instances and presenting instances according to the estimated noise. This leads to the approach of Optimism in the face of uncertainty and algorithms based on Upper Confidence Bounds introduced in [4], with the advantage of working under a finite budget.

Our work resembles the work of Carpentier and Munos [5] and Carpentier et al. [6]. The authors study the problem of estimating the mean values of several distributions uniformly well. This is equivalent to the problem of regression by a piecewise constant function and thus, can be seen as a classification problem with an infinite number of classes. This is done under the constraint of using a finite number of samples, referring thereby to active learning. Toward this, they model the problem under a multi-armed bandit setting, in which pulling an arm corresponds to taking a sample in one of the distributions. The goal is to define an allocation strategy that aims to minimize a loss function. The loss can be, for example, the maximum one-arm loss defined by the distance between the mean values and their estimate [5] or the weighted sum of this one-arm loss [6]. To minimize those losses, samples need to be allocated in proportion to the variance or the standard deviation of the loss. Because the variance/standard deviation is unknown, it has to be estimated at the same time as the allocation of the samples, resulting in the trade-off between to either use samples to learn the variance or to estimate the mean values. The authors use Optimism in the face of uncertainty, which is a common approach to solve this dilemma by computing high probability bounds on the value to estimate and sampling the arm with the highest bound.

This paper shows how to use the multi-armed bandit setting for active learning in classification by adapting the algorithms designed in [5] and [6] to the specific case of binary classification. To do so, the two kinds of loss have been redefined using a new one-arm loss, which represents the expected regret of the true risk. Indeed, the optimal risk of a noisy distribution is non-zero, and thus it is imperative that effort not be wasted when trying to decrease a risk that cannot be decreased. Therefore, the loss function is the expected difference between the risk and the optimal risk. Having redefined them, we address the fact that they are not inversely proportional to the number of samples any more. Another advantage of classification is that the shape of the distribution of samples is known. Indeed, the samples belong only to the Bernoulli distribution. This allows us to derive extremely tight bounds. Allocation strategies have to be defined in the full knowledge setting, in which the parameters of distributions are known in advance. In the adaptive allocation setting, because the parameters of the distributions cannot be accessed, we use the approach of Optimism in the face of uncertainty to define algorithms that sample according to this strategy while considering the fact that values are only estimated.

In Section II, we define several loss and pseudo-loss functions to be minimized. Then, for the full knowledge setting, we derive the optimal allocation strategies that minimizes those...
losses. In Section III, we next formalize the adaptive allocation setting, and then derive high probability bounds on the losses. We also present four algorithms for sampling arms according to these bounds. Section IV describes the class of problems that represents every problem with a fixed partition. We evaluate our algorithms on this class of problems and show that our algorithms perform better than algorithms initially designed for regression. We conclude in Section V.

II. RELATED WORK

In the past few years, the field of active learning of noisy (or not) binary classification algorithms has been largely studied, and is surveyed in [7]. Given that probabilistic classifiers output a probability of the instance having a particular label, Lewis and Gale [8] introduced the Uncertainty Sampling Algorithm which samples instances for which the classifier is least certain of class membership, i.e. the output is closest to 0.5. Kapoor et al. [9] [10] also consider the quality of estimation of this output.

Other authors make use of a set of classifiers. In [11], [12], and [13], the authors use a version space which is the set of classifiers consistent with all labels revealed so far, and a region of uncertainty being the region where there exist a pair of hypotheses that disagree, at each time step, a sample is taken in the region of uncertainty and the classifiers that are inconsistent with the sample are eliminated from the version space.

Follow up work has adapted this to noisy classification. In [14], a confidence interval on the performance of the classifiers is established and the classifiers eliminated are those for which it is not possible that they perform best. In [15] and [16], a probability of being best is assigned to each classifier, a sample is taken for which the classifiers disagree most, weighted by this probability, and a Bayesian update is applied to this probability with each sample. In [17] a sample is taken for which the volume of the version space will reduce the most.

III. ALLOCATION STRATEGY IN FULL KNOWLEDGE

In this section, we define four allocation strategies that find the best labels to assign to subsets using a budget of \( n \) samples in the full knowledge setting. In this setting, the allocation strategies give the number of samples to take in each subset according to the true subset parameters, regardless of the value of their estimation. We thus define four kinds of losses to measure the quality of the predictions. We then give the batch and online allocation strategies that minimize those losses.

Let \( X \) be an instance space and \( Y = \{0, 1\} \) be the set of possible labels. When we present an instance \( x \in X \) to the oracle, its label \( y \in Y \) is given.

The instance space is partitioned into \( K \) subsets, this partition does not change during the progress of the algorithm. Because the goal is to assign the best label to each subset, the hypotheses space is defined by the set of piecwise constant functions over this partition.

Let \( N = \{X_i, ..., X_K\} i \in \{1, ..., K\} : X_i \neq \emptyset, \bigcup_{i=1}^{K} X_k = X, \forall i, j \in \{1, ..., K\} : i \neq j \implies X_i \cap X_j = \emptyset \) be the fixed partition of \( X \), and \( H = \{ f : X \rightarrow Y, f(x) = \sum_{k=1}^{K} \{x \in X_k\} y_k, y_k \in Y, X_k \in N \} \) be the hypotheses space.

We formalize this problem under a K-armed bandit setting in which each subset \( X_i \in N \) is an arm characterized by a Bernoulli distribution \( \nu_k \) with mean value \( \mu_k \). Indeed, samples taken in a given subset can only have a value of 0 or 1. At each round, or time step, \( t \geq 1 \), an allocation strategy selects an arm \( k_t \), which corresponds to picking an instance randomly in a subset and presenting it to the oracle, and receives a sample \( y_{k_t} \sim \nu_{k_t} \), independently of the past samples. Let \( \{w_k\}_{k=1}^{K} \) denote the weight of each subset, with \( \sum_{k=1}^{K} w_k = 1 \). For example, in a semi-supervised context using pool-based sampling, each weight is proportional to the number of unlabelled data points in each subset, while in membership query synthesis, the weights are the sizes or areas of subsets.

We write \( T_{k,t} = \sum_{s=1}^{t} \{k_s = k\} \) as the number of times arm \( k \) has been pulled up to time \( t \), this way \( (T_{k,t})_{k \in \{1, ..., K\}} \) denotes the allocation strategy. Let \( \hat{\mu}_{k,t} = \frac{1}{T_{k,t}} \sum_{s=1}^{T_{k,t}} y_{k,s} \) be the empirical estimate of the mean \( \mu_k \) at time \( t \).

Let us now show how to derive losses that measure the need of sample. Usually, in a classification setting, we judge the performance of an algorithm by measuring the risk incurred. Here, the risk is based on the binary loss \( L_{0/1}(y, f(x)) = 1 \) if \( f(x) \neq y \) and 0 otherwise.

\[
R_k(y) = \begin{cases} 
1 - \mu_k & \text{if } y = 1 \\
\mu_k & \text{if } y = 0 
\end{cases} \tag{1}
\]

is the one-arm true risk, and

\[
\hat{R}_{k,n}(y) = \begin{cases} 
1 - \hat{\mu}_{k,n} & \text{if } y = 1 \\
\hat{\mu}_{k,n} & \text{if } y = 0 
\end{cases} \tag{2}
\]

is the one-arm estimated risk.

The best label the algorithm can assign to arm \( k \) using the samples received is \( \lceil \hat{\mu}_{k,n} \rceil \), which minimizes the estimated risk, where \( \lceil \cdot \rceil \) is the round operator. However, the optimal label the algorithm should assign to arm \( k \) should be \( \lceil \mu_k \rceil \). This incurs a regret in the true risk \( R_k(\lceil \hat{\mu}_{k,n} \rceil) - R_k(\lceil \mu_k \rceil) \).

In order to define a strategy that allocates samples according to the \( \lceil \mu_k \rceil \) values regardless of their estimates, the regret is expected over all the samples.

This gives us the following definition of the one arm loss for classification, as the expected regret of the one arm true risk,

\[
L_{k,n} = [R_k(\lceil \hat{\mu}_{k,n} \rceil) - R_k(\lceil \mu_k \rceil)], \tag{3}
\]

where the expectation is taken over all the samples.

Remark: It is important to consider the regret in the true risk. If we only consider the true risk of \( \lceil \hat{\mu}_{k,n} \rceil \), samples could be allocated to decrease a risk which is already optimal.

We now introduce two kinds of losses. As in [5] and [6], each of them is based on the one-arm losses:

- the sum of the one-arm losses,

\[
L_n^*(\{T_{k,n}\}_{k \in \{1, ..., K\}}) = \sum_{k=1}^{K} w_k L_{k,n}, \tag{4}
\]
• the maximum one-arm loss among subsets,

\[ I^m_n((T_{k,n})_{k \in \{1, \ldots, K\}}) = \max_k w_k L_{k,n}. \tag{5} \]

Note that

\[ L_{k,n} = [(1 - \mu_k)|\hat{\mu}_{k,n}|=1 + \mu_k|\hat{\mu}_{k,n}|=0 \]
\[ - (1 - \mu_k)|\mu_k|=1 - \mu_k|\mu_k|=0 \]
\[ = 2|\mu_k - 0.5|(|\hat{\mu}_{k,n}| - |\mu_k|). \tag{8} \]

This one-arm loss takes an interesting form as it separates theoretically two behaviours we expected. First, the \(|\hat{\mu}_{k,n}| \neq |\mu_k|\) part results in a higher loss for \(\mu_k\) close to 0.5, as the estimation has a higher chance of being on the other side of 0.5. Second, the \(|\mu_k - 0.5| \neq \mu_k\) part results in a higher loss for \(\mu_k\) far from 0.5, as the performance of the classifier is able to increase by a greater value.

The objective is now to build algorithms that minimize those losses. However, the method we use to find the best allocation strategy makes assumptions about the shape of the loss, mainly it has to be strictly decreasing with each \(T_{k,t}\). In order to get a function to minimize with a more convenient shape, we prefer to bound those losses by a pseudo-loss. The algorithms we build aim to minimize those pseudo-losses instead of the losses defined previously. The idea is thus to bound the probability \(|\hat{\mu}_{k,t}| \neq |\mu_k|\). Two methods can be applied. Either we use Bienaymé-Tchebychev’s inequality, and obtain a pseudo-loss which is inversely proportional to the number of samples in the subset, or we use the fact that the estimated mean in one subset follows a binomial distribution (labels are either 0 or 1) resulting in a non-linear function of \(\frac{1}{T_{k,t}}\), but allows the bound to be very tight.

### A. Bienaymé-Tchebychev’s Inequality

In this subsection, we give two online allocation criteria based on pseudo-losses that bound the losses defined in equations (4) and (5) while being inversely proportional to the number of samples in each subset \(T_{k,n}\).

We first state that

\[ (|\hat{\mu}_{k,n}| \neq |\mu_k|) \leq (|\hat{\mu}_{k,n} - \mu_k| > |\mu_k - 0.5|). \tag{9} \]

Then, using Bienaymé-Tchebychev’s inequality,

\[ (|\hat{\mu}_{k,n} - \mu_k| > |\mu_k - 0.5|) < \frac{\sigma_k^2}{T_{k,n} (\mu_k - 0.5)^2} \tag{10} \]

where, \(\sigma_k^2\) is the variance of the distribution \(\nu_k\). As we know that \(\nu_k\) is a Bernoulli distribution, \(\sigma_k^2 = \mu_k (1 - \mu_k)\).

We therefore define the two following pseudo-losses:

\[ \hat{I}^{s,BT}_n((T_{k,n})_{k \in \{1, \ldots, K\}}) = \sum_{k=1}^K w_k \frac{\sigma_k^2}{T_{k,n} (|\mu_k| - 0.5)} \tag{11} \]

and

\[ \hat{I}^{m,BT}_n((T_{k,n})_{k \in \{1, \ldots, K\}}) = \max_{k \in \{1, \ldots, K\}} \frac{2w_k}{T_{k,n} (|\mu_k| - 0.5)} \sigma_k^2 \tag{12} \]

Let \(T_{k,n}^{s,BT}\) and \(T_{k,n}^{m,BT}\) be the optimal number of samples to take in each subset in order to minimize \(\hat{I}^{s,BT}_n\) and \(\hat{I}^{m,BT}_n\), respectively, under the constraint that \(\sum_{k=1}^K T_{k,n}^{s,BT} = n\) and \(\sum_{k=1}^K T_{k,n}^{m,BT} = n\):

\[ T_{k,n}^{s,BT} = \frac{\sqrt{w_k \sigma_k}}{\sum_{i=1}^K \sqrt{w_i \sigma_i}} \frac{\sigma_k}{\sqrt{|\mu_k| - 0.5}} n \tag{13} \]

and

\[ T_{k,n}^{m,BT} = \frac{w_k \sigma_k^2}{\sum_{i=1}^K w_i \sigma_i^2} \frac{\sigma_k^2}{\sqrt{|\mu_k| - 0.5}} n \tag{14} \]

In the online allocation setting, at each round \(t\) a full knowledge algorithm would sample the arm \(k^{s,BT}_t\) or \(k^{m,BT}_t\) depending on the pseudo-loss considered. With

\[ k^{s,BT}_t \in \arg \max_{1 \leq k \leq K} \frac{T_{k,t}^{s,BT}}{T_{k,t} \sqrt{|\mu_k| - 0.5}} \tag{15} \]
Figure 1 illustrates the criteria in 15 and 16. In both cases, a mean value of 0.5 for one of the subsets is problematic because all of the samples are absorbed by one subset. For this subset the numerator value is infinite. These pseudo-losses are therefore not recommended if one subset has a value of 0.5. To counter that we saturate the value \( \frac{w_k \sigma_k^2}{\ell_{k,t}} = 0 \) of equation (10) by 1 because it represents a probability and therefore cannot be greater than 1. In this case, the one-arm pseudo-loss is thus not inversely proportional to \( T_{k,t} \). We will see in the next section that it is not a problem because this approach works with any type of functions. Another solution is to find a better bound for the probability \( \frac{\left( \mu_k, \ell_{k,t} \right) \neq [\mu_k]}{\mu_k} \), which implies working with one-arm pseudo-losses not inversely proportional to \( T_{k,t} \).

**B. Binomial distribution**

In this section we bound the losses defined in equations (4) and (5) with the knowledge that the distributions of estimated mean values are binomial, allowing us to give a tight bound to the probability \( \left( \mu_k, \ell_{k,t} \right) \neq [\mu_k] \), and thus, giving an allocation strategy that is very close to the optimal towards the true equation (10) by 1 because it represents a probability and therefore cannot be greater than 1. In this case, the one-arm pseudo-loss is thus not inversely proportional to \( T_{k,t} \).

Let \( I_{k,n} = \mu_k \) and \( I_{l,n} = \mu_k \), respectively under the constraint that \( \sum_k T_{k,n} = n \) and \( \sum_{k=1}^{K} \frac{T_{k,n}}{2} = n \).

Note that the probability given above is a step function of \( T_{k,n} \) and thus is not a strictly decreasing function of \( T_{k,n} \). That is not convenient as we require this condition in the later. That is why we bound this probability by bounding the truncated value \( T_{k,n} / 2 \). Then,

\[
\left( \mu_k, \ell_{k,t} \right) \neq [\mu_k] \leq \max_{k \in \{1, \ldots, K\}} l_k(T_{k,n}, \mu_k) = \max_{k \in \{1, \ldots, K\}} l_k(T_{k,n}, \mu_k) = \max_{k \in \{1, \ldots, K\}} l_k(T_{k,n}, \mu_k).
\]

We therefore define the two following pseudo-losses:

\[
\ell^{s,bin}_n(T_{k,n}) = \sum_{k=1}^{K} l_k(T_{k,n}, \mu_k)
\]

and

\[
\ell^{m,bin}_n(T_{k,n}) = \max_{k \in \{1, \ldots, K\}} l_k(T_{k,n}, \mu_k),
\]

with

\[
l_k(T_{k,n}, \mu_k) = 0.5 \left( 1 - I_{k,n} - \mu_k \right)_{\max} \left( \mu_k - 0.5 \right) \left( 1 - I_{k,n} - \mu_k \right)_{\max} \left( \mu_k - 0.5 \right).
\]

The equalities above are true only if \( l_k \) and \( \frac{\partial l_k}{\partial T_{k,t}} \) are strictly decreasing function of \( T_{k,t} \).

Figure 3 illustrates the criteria in 31 and 32. We can see that it follows the expected behaviour. In both cases, the value of the criteria at \( \mu_k = 0.5 \) is 0, which means that a full knowledge algorithm will allocate no samples to a subset with such a value. This reflects the fact that there is no gain in terms of risk of doing this. The value of the criteria at an extreme side

![Graph showing the criteria in 31 and 32](image-url)
Let \( \phi \) and \( \psi \) be one of the online criteria corresponding to allocation strategies in [5] and [6] are regrouped in Table I. Let \( f_{k,t} \) be one of the online criteria used to allocate samples in full knowledge. \( f_{k,t} \) is shown for each pseudo-loss in Table I, in the first column are the criteria derived from the pseudo-losses introduced in [5] and [6] for the case of regression; in the other two columns are criteria derived from pseudo-losses that use either Bienaymé–Tchebychev’s inequality or the knowledge of the binomial distribution. The rows correspond to the two losses on which pseudo-losses are based on, \( i.e. \) if we consider the maximum or the sum of the true risk among subsets.

The \( \mu_k \) values are, however, unknown, therefore we cannot build an algorithm that picks \( T_{k,n} \) samples in each subset and attain optimality. We thus use an optimistic approach to estimate the \( \mu_k \) and at the same time allocate samples as close to the optimum as possible.

### IV. Allocation Strategy Using Estimated Means

#### Core algorithm

**Require:** \( \delta \)

- Pull each arm once

For \( t = K + 1 \) to \( n \) do

- Compute \( B_{k,t} = \mu_k - \epsilon_k \) with \( \epsilon_k \) such as \( \epsilon_k \) coming from Table I, depending on which algorithm you consider, for each arm \( 1 \leq k \leq K \)

End for

Ensure: \( [\hat{\mu}_{k,n}] \) for all arms \( 1 \leq k \leq K \)

#### Table I: full knowledge criteria

<table>
<thead>
<tr>
<th>( f_{k,t} )</th>
<th>Regression</th>
<th>Classif. (B.T.)</th>
<th>Classif. (Bin.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_{nk} )</td>
<td>( \frac{w_k \sigma_k^2}{\sqrt{n}} )</td>
<td>( \frac{w_k \sigma_k^2}{\sqrt{n}} )</td>
<td>( I_k(T_{k,t,n}, \mu_k) )</td>
</tr>
<tr>
<td>( I_{nk}^c )</td>
<td>( \frac{w_k \sigma_k^2}{\sqrt{n}} )</td>
<td>( \frac{w_k \sigma_k^2}{\sqrt{n}} )</td>
<td>( \frac{w_k}{\phi_{k,t}}(T_{k,t}, \mu_k) )</td>
</tr>
<tr>
<td>( T_{k,t} )</td>
<td>( \frac{w_k \sigma_k^2}{\sqrt{n}} )</td>
<td>( \frac{w_k \sigma_k^2}{\sqrt{n}} )</td>
<td>( \frac{w_k}{\phi_{k,t}}(T_{k,t}, \mu_k) )</td>
</tr>
</tbody>
</table>

In this section, we introduce four algorithms derived from the full knowledge criteria of the previous section. We now refer to the setting of adaptive allocation, in which we address how to learn the parameters of the distributions and allocate the samples optimally with respect to these parameters. This is known as the exploration/exploitation dilemma. We solve this problem, using Optimism in the face of uncertainty, which computes a high probability bound on the value to be maximized, and then samples the arm with the highest bound. Accordingly, if the value to estimate is close to the upper bound, then the sample is well chosen, otherwise, the bound will tighten up, the upper bound will decrease, and the arm will not be sampled next time, the uncertainty has reduced. We therefore use this approach in our problem to build adaptive algorithms that allocate samples closest to the optimal.

Each of these algorithms follows the same core, which is described in Fig. 4. The difference between them lies in the criterion depending on the pseudo-loss considered. They take one parameter as input: \( \delta \) which defines the confidence level of the bound. The amount of exploration of the algorithms can be adapted by properly tuning \( \delta \).

Usually, the high probability bounds are derived from general concentration inequalities for which the shape of the distribution is unknown. In our case, we search a Bayesian credible interval on the criterion, which are functions of the
\( \mu_k \) values. Moreover, we know that the estimated means are drawn from a Bernoulli distribution.

Let us state that Beta distributions provide a family of conjugate prior probability distributions for binomial distributions.

The uniform distribution Beta(1,1) is taken as the prior probability distribution, because we have no information about the true distribution.

Using the Bayesian inference:

\[
(\mu_k = x|\hat{\mu}_k,t, T_{k,t}) = \frac{x^{T_{k,t}\hat{\mu}_k,t}(1-x)^{T_{k,t}(1-\hat{\mu}_k,t)}}{\text{Beta}(T_{k,t}\hat{\mu}_k,t + 1, T_{k,t}(1-\hat{\mu}_k,t) + 1)}
\]  
(33)

Obviously,

\[
(f(T_{k,t}, \mu_k) > e_k|\hat{\mu}_k,t, T_{k,t}) = \frac{(\mu_k, f(T_{k,t}, \mu_k) > e_k|\hat{\mu}_k,t, T_{k,t})}{(\mu_k, f(T_{k,t}, \mu_k) > e_k|\hat{\mu}_k,t, T_{k,t})}.
\]  
(34)

(35)

Let \( I_k = \{\mu_k|f(T_{k,t}, \mu_k) > e_k\} \), then

\[
(f(T_{k,t}, \mu_k) > e_k|\hat{\mu}_k,t, T_{k,t}) = \frac{\int_{x \in I_k} x^{T_{k,t}\hat{\mu}_k,t}(1-x)^{T_{k,t}(1-\hat{\mu}_k,t)}dx}{\text{Beta}(T_{k,t}\hat{\mu}_k,t + 1, T_{k,t}(1-\hat{\mu}_k,t) + 1)}.
\]  
(36)

(37)

The method is thus for a fixed \( e_k \) to find the domain on which \( f(T_{k,t}, \mu_k) > e_k \), and compute the probability \( (f(T_{k,t}, \mu_k) > e_k|\hat{\mu}_k,t, T_{k,t}) \) and then to invert the function \( \delta = (f(T_{k,t}, \mu_k) > e_k|\hat{\mu}_k,t, T_{k,t}) = g(e_k) \) to get the \( e_k \) corresponding to a fixed \( \delta \).

This method is illustrated in Figure 5. The circle dotted curve represents the criterion of the pseudo-loss \( L_{m, \text{bin}} \), while the star dotted curve represent the density of probability of \( \mu_k \), with \( T_{k,t} = 10 \) and \( \hat{\mu}_k,t = 0.7 \). From a fixed \( e_k = 0.03 \) we found the domain \( I_k = \{\mu_k|0.3 \leq \mu_k \leq 0.43 \text{ or } 0.57 \leq \mu_k \leq 0.7\} \) on which \( l_k(T_{k,t}, \mu_k) > e_k \). The probability \( (f(T_{k,t}, \mu_k) > e_k|\hat{\mu}_k,t, T_{k,t}) \) is thus the area under the curve of the density of probability of \( \mu_k \) on the domain \( I \), shown by the green (grey) area on the figure.

Figure 5: method to obtain the high probability bound of a criterion

V. RESULTS

In this section, we evaluate empirically the algorithms introduced in the previous section on a built-in problem. We first demonstrate the performance of those algorithms in full knowledge, where the distribution parameters are known, to establish the goal standard (the best we can expect from those algorithms). Then, we evaluate the algorithms for adaptive allocation and check if the exploration/exploitation trade-off is well achieved.

Any classification problem which involves a fixed partition can be modelled by the following parameters:

- the number of subsets \( K \),
- the mean value of the labels drawn from each subset \( (\mu_k)_{k \in \{1, \ldots, K\}} \),
- the relative importance of each subset \( (w_k)_{k \in \{1, \ldots, K\}} \).

Indeed, the relative position of a subset to another has no influence on the problem, this is the reason why we could model it under a K-armed bandit problem. The fact that we only care about the label of samples belonging only to \( \{0, 1\} \) implies that the distribution is Bernoulli and so each subset is only characterized by its mean value of labels.

To evaluate our algorithms we randomly generate problems belonging to the class described above:

- the number of subsets \( K \) is drawn uniformly in \( \{1, \ldots, K_{max}\} \), in the following \( K_{max} \) is arbitrarily set to 50,
- for each subset \( k \), \( \mu_k \) is drawn uniformly in \( [0, 1] \),
- for each subset \( k \), \( w_k \) is drawn uniformly in \( [0, 1] \), then the \( w_k \) values are normalized to sum to 1.

We generate this way a batch of 1000 problems. We then run our algorithms on each of these problems for 300 time steps and evaluate their performance by computing the true risk of the current predictions at each time step. The global performance of the algorithms results from averaging the true risks of each problem at each time step. The results are displayed with the time steps range starting at 25 for clarity.

During our evaluations, the algorithms are called as follows:

- Random sampling is the algorithm for which samples are taken uniformly in each subset, regardless of the mean value of distributions (baseline),
- CH-AS and MCUCB are algorithms introduced in [5] and [6] respectively.
- \( m \) BT is the algorithm based on the maximum one-arm loss bounded using Bienaymé-Tchebytchev’s inequality,
- \( s \) BT is the algorithm based on the sum of the one-arm losses bounded using Bienaymé-Tchebytchev’s inequality.
Tchebytchev’s inequality (algorithms which one-arm losses are bounded using Bienaymé-binomial (are bounded using the knowledge of the distribution being 
we want to be decreasing. they tend to minimize is different than the evaluation function 
constraints necessary to design a criterion aiming to decrease 
the distribution of samples in subsets following more precisely 
the pseudo-losses bounding the losses in a tighter way, and thus 
taken from a classiﬁcation point of view. The loss function 
sum of the one-arm losses bounded using the knowledge of the binomial distribution,

- \textit{m binomial} is the algorithm based on the maximum one-arm loss bounded using the knowledge of the binomial distribution,

- \textit{s binomial} is the algorithm based on the sum of the one-arm losses bounded using the knowledge of the binomial distribution.

A. Evaluation of the Algorithms in Full Knowledge

First, we evaluate the algorithms in full knowledge. At each time step, the algorithm picks a sample in the arm for which the criterion from Table I —depending on which algorithm— is maximum. The results of the evaluation are shown in Figure 6. We can see that the algorithms designed in this paper perform better than their counterparts —with respect to the kind of loss they use— introduced in [5] and [6]. That is, \textit{m} binomial and \textit{m} BT perform better than CH-AS and \textit{s} binomial and \textit{s} BT perform better than MCUCB. This was expected since those algorithms are designed for regression and the evaluation is taken from a classiﬁcation point of view. The loss function they tend to minimize is different than the evaluation function we want to be decreasing.

It can also be seen that algorithms which one-arm losses are bounded using the knowledge of the distribution being binomial (ie \textit{m} binomial and \textit{s} binomial) performs better than algorithms which one-arm losses are bounded using Bienaymé-Tchebytchev’s inequality (ie \textit{m} BT and \textit{s} BT). This comes from the pseudo-losses bounding the losses in a tighter way, and thus the distribution of samples in subsets following more precisely the optimal distribution for the actual loss. All in keeping the constraints necessary to design a criterion aiming to decrease this loss.

Because \textit{m} binomial and \textit{s} binomial perform best, we focus on them in the following evaluation.

B. Evaluation of the Algorithms Under Adaptive Allocation

Let us now evaluate the results of the algorithms for adaptive allocation. One notable feature is we can control the exploration/exploitation tradeoff through the value of the parameter \( \delta \). A low value of \( \delta \) will result in more exploration, a \( \delta \) of 0 means that the strategy is equivalent to random sampling. On the other side, a high value of \( \delta \) will result in more exploitation, the algorithm being much more conﬁdent about its estimation of the mean values. The exploration is necessary, otherwise the algorithms would rush into a certain subset and never sample others. Let us detail this part, in the pure exploitation setting, the ﬁrst estimation of the pseudo-losses up to 2 points in each subset is always zero as the mean values estimates are either 0, 1 or 0.5. The ﬁrst subset that is sampled more than 2 times will see its estimated pseudo-loss become positive which makes this subset the one to sample next and so on. This one subset would thus drain all the samples.

We first evaluated our algorithms with a low value for \( \delta \) resulting in high probability bounds, which was intended. We ran our algorithms with \( \delta = 0.1 \). We display the results of this run on Fig. 7. We can see that all the algorithms have similar performance, slightly better than Random Sampling. An in-depth study showed that the lack of performance in this case was due to a lack of exploitation. This is not common because in most of UCB-like algorithms the conﬁdence intervals are wanted to contain the true value with high probability, which correspond to a low value of \( \delta \), usually 0.1 or 0.01. To include more exploitation in our algorithms we have to increase the value of \( \delta \).

We thus evaluate our algorithms with \( \delta = 0.9 \) and display the results on Fig. 9. In this case the algorithms are more conﬁdent about their estimation. We can see that \textit{m} binomial have largely increased its performance, supporting our previous justiﬁcation. The fact that a high value of \( \delta \) is not penalizing the algorithms may come from knowing the distribution which leads to extremely tight bounds.

Another thing we can see is that for this value of \( \delta \), the algorithm \textit{s} binomial does not perform as good as we expected, indeed, it performs worse than MCUCB and CH-AS. Each algorithm may not have the same optimal values of \( \delta \). We have thus tuned separately each algorithm using a grid search to retrieve the optimal value of \( \delta \) for each one of them. We then evaluate the algorithms with those values of \( \delta \) and display

![Figure 6: True risk of the algorithms in the full knowledge setting](image1)

![Figure 7: True risk of the algorithms in the adaptive allocation setting with \( \delta = 0.1 \)](image2)
Finally, we can see that the algorithms built in this paper for the specific problem of classification, with suited pseudo-losses and confidence bounds, behave better than those designed for regression in [5] and [6].

VI. CONCLUSION

The paper proposes a method to use the Optimism in the face of uncertainty approach in an active learning problem for classification. It introduces four algorithms which perform comparatively well and open a new avenue of research. The framework established in this paper is related to the problems encountered in text classification and resembles the problem of parameters estimation in multi-binomial distributions. Working with a fixed partition generates new questions that state-of-the art on active learning do not have. In Uncertainty Sampling the intent is to sample close to the boundary ($\mu(x)$ close to 0.5) because this will redefine it, whereas in our work the partition cannot change. This leads to finding new loss functions. Our future work concern will be about an adaptive partition of the space as well as the combination of the information providing from several different partitions, we will also study the multi-class case.

REFERENCES