Optimistic Active Learning for Classification

Timothé Collet[1] and Olivier Pietquin[2]

[1] Supelec, MaLIS Research group, GeorgiaTech-CNRS UMI 2958, France
timothe.collet@supelec.fr
[2] University Lille 1, LIFL (UMR 8022 CNRS / Lille 1), SequeL Team, France
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 multi-armed bandits theory. Based on previous work on bandit theory applied to active learning for regression, we introduce two 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.

1 Introduction

We place ourselves in the supervised learning framework, especially in a noisy 2-class classification problem. Our work focuses on active learning, which is the process consisting in driving the choice of the examples that need to be labelled in order to minimize the number of queries to the oracle. To do so in an online context the algorithm successively chose the best example to present to the oracle taking into account the information provided from all the previous samples, thus, it can be seen as a sequential decision process. The noisy aspect of this problem —which may come from an intrinsic noise on the data or from an inability of the classifier to distinguish examples— and the fact that the noise is not the same for all examples, implies that some examples can be more or less difficult to classify. Indeed, it is relatively intuitive that little effort must be put into the least noisy examples as they are easy to classify. While it is less intuitive that little effort must also be put into very noisy examples too. Thus, this is an online allocation problem with respect to the noise value, and can be represented by a multi-armed bandit setting, introduced in [14] and surveyed in [3]. A major issue of this problem is that we do not know in advance the noise of an example, but it can be learnt while we present examples to the oracle. We therefore have to make a trade-off between learning the noise of examples and presenting examples according to the noise. Leading to the approach of *Optimism in the face of uncertainty* and algorithms based on *Upper Confidence Bounds* introduced in [1], with the advantage of working under a finite budget.

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 [12]. In [8], the authors introduced the Uncertainty Sampling Algorithm. It uses probabilistic classifiers, which output a probability of the example having a particular label. The idea is to sample examples for which the classifier is least certain of class membership, *i.e.* the probability is the closest to 0.5. In [7], the authors add to this the quality of estimation of this output. The more the output is close to 0.5, and the less the quality of the estimation is, the highest the probability of giving the wrong label, and so, the more the algorithm tends to sample the example. Other authors make use of a set of classifiers. In [6], [13] and [10], the authors use a version space being 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 disagrees, at each time step, a sample is taken in the region of uncertainty and all the classifiers inconsistent with the sample are eliminated from the version space. Some work adapt this to noisy classification. In [2], 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 [11] and [9], a probability of being the best is affected to each classifier, a sample is taken where the classifiers disagree most, and a Bayesian update is applied to this probability with each sample.

In [5] and [4], the authors study the problem of estimating uniformly well the mean values of several distributions which is equivalent to the problem of regression by a piece-wise 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. To this end, they model the problem under a multi-armed bandit setting, in which pulling an arm correspond to taking a sample in one of the distributions. The goal is to define an allocation strategy which aims to minimize a loss function. In [5] the loss is the maximum one-arm loss defined by the distance between the mean values and their estimate. Whereas in [4], the loss is the weighted sum of this one-arm loss. To minimize those losses samples need to be allocated in proportion to the variance or the standard deviation depending on the loss. The variance/standard deviation being unknown it has to be estimated at the same time as the allocation of the samples, resulting in a dilemma between using 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 [4] 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, thus efforts could be spent for nothing 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 had too deal with the fact that they aren't inversely proportional to the number of samples any more. An other advantage of classification is that the shape of the distribution of samples is known. Indeed, the samples belong only to $\{0, 1\}$, the distribution is thus Bernoulli. This allows us to derive extremely tight bounds. In the adaptive allocation setting, the parameters of the distributions cannot be accessed, we use the approach of *Optimism in the face of uncertainty*. Allocation strategies have to be defined in the full knowledge setting, in which the parameters of distributions are known in advance, and afterwards build algorithms that sample according to this strategy plus some uncertainty.

In Section II, we define several loss and pseudo-loss functions which have to be minimized. Then, we place ourselves in the full knowledge setting and find the optimal allocation strategies which minimizes those losses. In Section III, we place ourselves in the adaptive allocation setting, and define high probability bounds on the losses. We then present our algorithms which sample arms according to these bounds. In Section IV, we describe a toy problem and show that it is representative of more general problems. Then, we evaluate our algorithms on this toy problem and show that our algorithms perform better than algorithms initially designed for regression. In Section V, we show a conclusion.

2 Allocation strategy in full knowledge

After formalizing our problem under a K-armed bandit setting, we define several kinds of losses to be minimized. We then give the optimal allocation strategy in full knowledge as well as an online criteria to sample according to this strategy.

Let X be an instance space and $Y = \{0, 1\}$ be the set of possible labels. Let an oracle label $x \in X$ with $y \in Y$. Let $N = \{X_1, ..., X_K | \cup_{k=1}^{K} X_k = X, \cap_{k=1}^{K} X_k = \emptyset\}$ be a fixed clustering of X, and $\mathcal{H} = \{f : X \to Y, f(x) = \sum_{i=1}^{K} \mathbb{1}_{\{x \in X_k\}} y_k, y_k \in Y, X_k \in N\}$ be the hypotheses space defined by piecewise constant functions. The goal is to learn the hypothesis which predictions are as close as possible to the oracle's, with as few queries as possible to it.

This problem can be formalized under a K-armed bandit setting where each cluster is an arm k = 1,...,K characterized by a Bernoulli distribution ν_k with mean value μ_k. Indeed, samples taken in a given cluster 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 example randomly in a cluster and presenting it to the oracle, and receives a sample $y_{k,t} \sim \nu_k$, independently of the past samples. Let $\{w_k\}_{k=1,...,K}$ denote the relative importance of a cluster, summing to 1, coming from the knowledge of how much a cluster will be solicited while using the classifier (not learning it). Most of the time, we will take the distribution of unlabelled data among clusters for $\{w_k\}_{k=1,...,K}$. The goal is to define a strategy that finds the best labels to assign to clusters using a budget of n samples.

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

We now introduce two kinds of losses, as in [5] and [4], each of them is based on a one-arm loss: the maximum one-arm loss among clusters and the sum of the one-arm losses. Let us now see how the one-arm loss is built.

Usually, in a classification setting, we judge on 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. Thus, the one-arm true risk is $R_k(y) = 1 - \mu_k$ if $y = 1$ and μ_k if $y = 0$, and the one-arm empirical risk is $\hat{R}_{k,n}(y) = 1 - \hat{\mu}_{k,n}$ if $y = 1$ and $\hat{\mu}_{k,n}$ if $y = 0$. In order to minimize the empirical risk, which is the best an algorithm can do knowing only the samples received, the algorithm always assigns the label $[\hat{\mu}_{k,t}]$ to arm k at round t. The risk of the algorithm must always be compared to the best risk that can be reached, otherwise efforts could be spent for nothing when trying to decrease a risk that cannot be decreased. Here, the best risk of arm k can be reached with the label $[\mu_k]$. We therefore define the one arm loss for classification which is the expected regret of the one arm true risk,

$$L_{k,n} = \mathbb{E}[R_k([\hat{\mu}_{k,n}]) - R_k([\mu_k])] = 2|\mu_k - 0.5|\mathbb{P}([\hat{\mu}_{k,n}] \neq [\mu_k]), \tag{1}$$

where the expectation is taken over all the samples.

The two kinds of losses now become

$$L_n^s((T_{k,n})_{k\in\{1,\dots,K\}}) = \sum_{k=1}^{K} w_k L_{k,n} \tag{2}$$

$$\text{and } L_n^m((T_{k,n})_{k\in\{1,\dots,K\}}) = \max_k w_k L_{k,n}. \tag{3}$$

The objective would now be to build algorithms that minimize those losses. However, the method we use to find the best allocation strategy is based on some conditions on the shape of the loss. In order to get a function to minimize with a more convenient shape, we prefer to bound those losses by a pseudo-loss.

We therefore use the knowledge that the mean values of each cluster follow a binomial distribution, allowing us to give a tight bound to the probability $\mathbb{P}([\hat{\mu}_{k,t}] \neq [\mu_k])$ while keeping pseudo-losses for which the one-arm pseudo-loss is strictly decreasing with $T_{k,t}$.

Let $\mathcal{I}_{1-\mu_k}(T_{k,n} - \lfloor T_{k,n}\hat{\mu}_{k,n}\rfloor, \lfloor T_{k,n}\hat{\mu}_{k,n}\rfloor + 1)$ be the cumulative distribution function of $T_{k,n}\hat{\mu}_{k,n}$ following the binomial distribution with parameters $T_{k,n}, \mu_k$. Then,

$$\mathbb{P}([\hat{\mu}_{k,n}] \neq [\mu_k]) = \mathbb{1}_{[\mu_k]=0}\mathbb{P}(\hat{\mu}_{k,n} \geq 0.5) + \mathbb{1}_{[\mu_k]=1}\mathbb{P}(\hat{\mu}_{k,n} < 0.5) \tag{4}$$

$$= \mathbb{1}_{[\mu_k]=0}(1 - \mathcal{I}_{1-\mu_k}(T_{k,n} - \lfloor T_{k,n}/2\rfloor, \lfloor T_{k,n}/2\rfloor + 1)) \tag{5}$$

$$+ \mathbb{1}_{[\mu_k]=1}\mathcal{I}_{1-\mu_k}(T_{k,n} - \lfloor T_{k,n}/2\rfloor, \lfloor T_{k,n}/2\rfloor + 1). \tag{6}$$

Note that the probability given above is a step function of $T_{k,n}/2$ and so 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 $\lfloor T_{k,n}/2 \rfloor$. Then,

$$\mathbb{P}([\hat{\mu}_{k,n}] \neq [\mu_k]) \leq \mathbb{1}_{[\mu_k]=0}(1 - \mathcal{I}_{1-\mu_k}(T_{k,n}/2+1, T_{k,n}/2)) \tag{7}$$
$$+ \mathbb{1}_{[\mu_k]=1}\mathcal{I}_{1-\mu_k}(T_{k,n}/2, T_{k,n}/2+1). \tag{8}$$

We therefore define the two following pseudo-losses:

$$\tilde{L}_n^{s,bin}((T_{k,n})_{k\in\{1,\dots,K\}}) = \sum_{k=1}^{K} l_k(T_{k,n}, \mu_k) \tag{9}$$
$$\text{and } \tilde{L}_n^{m,bin}((T_{k,n})_{k\in\{1,\dots,K\}}) = \max_{k\in\{1,\dots,K\}} l_k(T_{k,n}, \mu_k), \tag{10}$$

$$\text{with } l_k(T_{k,n}, \mu_k) = 2w_k|\mu_k - 0.5|[\mathbb{1}_{[\mu_k]=0}(1 - \mathcal{I}_{1-\mu_k}(T_{k,n}/2+1, T_{k,n}/2)) \tag{11}$$
$$+\mathbb{1}_{[\mu_k]=1}\mathcal{I}_{1-\mu_k}(T_{k,n}/2, T_{k,n}/2+1)] \tag{12}$$

the one-arm pseudo-loss.

One quality of this pseudo-loss is that while it has a convenient shape, it remains very tight. This means that minimizing this pseudo-loss acts almost as good as minimizing the true losses from equations (2) and (3).

Note that l_k as well as $\partial l_k/\partial T_{k,n}$ are both strictly decreasing functions of $T_{k,n}$. Thus, they admit inverses l_k^{-1} and $l_k^{'-1}$ respectively.

Let us remind that the objective is to find allocation strategies that minimize the pseudo-losses. Let $T_{k,n}^{*s,bin}$ and $T_{k,n}^{*m,bin}$ be the optimal number of samples to take in each cluster in order to minimize $\tilde{L}_n^{s,bin}$ and $\tilde{L}_n^{m,bin}$ respectively under the constraint that $\sum_{k=1}^{K} T_{k,n}^{*s,bin} = n$ and $\sum_{k=1}^{K} T_{k,n}^{*m,bin} = n$. Then,

$$T_{k,n}^{*s,bin} = l_k^{'-1}(c^*, \mu_k) \text{ and } T_{k,n}^{*m,bin} = l_k^{-1}(R^*, \mu_k), \tag{13}$$

with c^* and R^* such that $\sum_{k=1}^{K} l_k^{'-1}(c^*, \mu_k) = n$ and $\sum_{k=1}^{K} l_k^{-1}(R^*, \mu_k) = n$.

In the online allocation setting, at each round t a full knowledge algorithm would the arm $k_t^{s,bin}$ or $k_t^{m,bin}$ depending on the pseudo-loss considered. Under the condition that l_k and $\frac{dl_k}{dT_{k,t}}$ are strictly decreasing function of $T_{k,t}$, we have

$$k_t^{s,bin} \in \arg\max_{1\leq k\leq K} \frac{T_{k,t}^{*s,bin}}{T_{k,t}} = \arg\max_k l_k(T_{k,t}, \mu_k) \tag{14}$$
$$\text{and } k_t^{m,bin} \in \arg\max_{1\leq k\leq K} \frac{T_{k,t}^{*m,bin}}{T_{k,t}} = \arg\max_k \frac{\partial l_k}{\partial T_{k,t}}(T_{k,t}, \mu_k). \tag{15}$$

However, the μ_k values are unknown, therefore we cannot build an algorithm that picks $T_{k,n}^*$ samples in each cluster and attain optimality.

We thus use an optimistic approach to estimate the μ_k and at the same time allocate samples as close as possible to the optimum.

3 Allocation strategy using estimated means

In this section, we introduce two algorithms derived from the full knowledge criterion of the previous chapter. The full knowledge setting is not a realistic approach as if the means of the distribution were known in advance, then the optimal labels could be easily deducted from them. Thus, we now refer to the setting of adaptive allocation, in which we have to deal with learning the parameters of the distributions and allocate the samples optimally with respect to these parameters. This problem is usually called the exploration/exploitation dilemma. In order to solve this problem, we use the *Optimism in the face of uncertainty* approach which computes a high probability bound on the value to maximize, and sample the arm with the highest bound. This way, if the value to estimate is close to the upper bound, then the sample is well chosen, otherwise, the bound tighten up, the upper bound decreases, 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.

Input: δ
Initialize: Pull each arm twice
for *t = 2K +1, ..., n* **do**
  Compute $B_{k,t} = w_k e_k$ with e_k such as $\mathbb{P}(f(T_{k,t}, \mu_k) > e_k | \hat{\mu}_{k,T_{k,t}}) = \delta$
  for each arm $1 \leq k \leq K$ Pull an arm $k_t \in argmax_{1 \leq k \leq K} B_{k,t}$
end
Output: $[\hat{\mu}_{k,n}]$ for all arms $1 \leq k \leq K$

Algorithm 1: Core algorithm

Each algorithm follows the same core, which is described in Algorithm 1, where the difference lies in the criteria f which should be replaced by $l_k(T_{k,t}, \mu_k)$ or $\frac{\partial l_k}{\partial T_{k,t}}(T_{k,t}, \mu_k)$ depending on the pseudo-loss considered. They take one parameter as input: δ which defines the confidence level of the bound. The amount of exploration of the algorithms can be adapted by properly tuning δ.

Usually, the high probability bounds are derived from general concentration inequalities where the shape of the distribution is unknown. Here, we search a confidence interval on the criterion which are functions of the μ_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 :

$$\mathbb{P}(\mu_k = x | \hat{\mu}_{k,t}, T_{k,t}) = \frac{x^{T_{k,t}\hat{\mu}_{k,t}}(1-x)^{T_{k,t}(1-\hat{\mu}_{k,t})}}{Beta(T_{k,t}\hat{\mu}_{k,t}, T_{k,t}(1-\hat{\mu}_{k,t})+1)} \tag{16}$$

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

$$\mathbb{P}(f(T_{k,t}, \mu_k) > e_k | \hat{\mu}_{k,t}, T_{k,t}) = \int_{x \in I_k} x^{T_{k,t}\hat{\mu}_{k,t}}(1-x)^{T_{k,t}(1-\hat{\mu}_{k,t})} dx. \tag{17}$$

4 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, 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 clustering can be modelled by the following parameters: *(i)* the number of clusters K, *(ii)* the mean value of the labels drawn from each clusters $(\mu_k)_{k\in\{1,\dots,K\}}$, *(iii)* the relative importance of each cluster $(w_k)_{k\in\{1,\dots,K\}}$.

Indeed, the relative position of a cluster to an other 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 cluster is only characterized by its mean value of labels.

To evaluate our algorithms we use the following parameters: *(i)* $K = 16$, being large enough to get some diversity on clusters but not too large as it would be useless, *(ii)* $\forall k, \mu_k = \frac{k}{16}$, this way we represent the largest variety of values possible, and thus our evaluation concerns all the values, *(iii)* the relative importance of each cluster $(w_k)_{k\in\{1,\dots,K\}} = \frac{1}{16}$ thus, our evaluation concerns equally all the values.

At each time step, each algorithm pick a sample in a cluster according to its allocation strategy and select the best hypothesis. Then its prediction is evaluated using the true risk, knowing the true mean values and the weights vector.

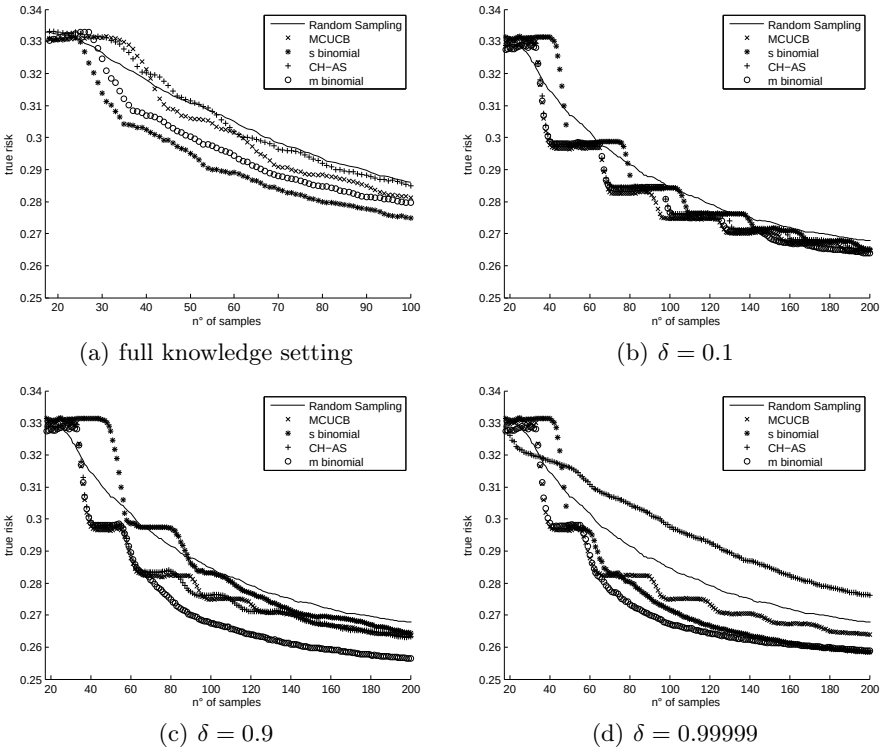
During our evaluations, the algorithms are called as follows:

- Random sampling is the algorithm for which samples are taken uniformly in each cluster, regardless of the mean value of distributions (baseline),
- CH-AS and MCUCB are algorithms introduced in [5] and [4] respectively.
- **m binomial** is the algorithm based on the **maximum one-arm loss** bounded using the knowledge of the **binomial distribution**,
- **s binomial** is the algorithm based on the **sum of the one-arm losses** bounded using the knowledge of the **binomial distribution**,

4.1 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 online allocation criteria from equations (14) and (15) is maximum. The results of the evaluation are shown in Figure 1(a). We can see that the methods introduced by [5] and [4] —respectively MCUCB and CH-AS— do not perform better than random sampling. This is due to the fact that the algorithms are designed for regression and the evaluation is taken from a classification point of view. Those algorithms will allocate more samples to clusters with a mean value far to 0.5 than it has to. Indeed,

the goal of regression is to estimate precisely the mean value, whereas the goal of classification is to be able to predict a good label, so being able to know if a mean value is closer to 0.86 or 0.87 is of no interest for classification because in both cases the predicted label will be 1.



(a) full knowledge setting

(b) $\delta = 0.1$

(c) $\delta = 0.9$

(d) $\delta = 0.99999$

Fig. 1. True risk of the algorithms in the full knowledge setting (1(a)) or in the adaptive allocation setting with different values of δ (1(b), 1(c) and 1(d))

4.2 Evaluation of the algorithms under adaptive allocation

Let us now evaluate the results of the algorithms for adaptive allocation. One notable feature of those algorithms is the ability to control the amount of exploration versus exploitation through value of the parameter δ. A low value of δ will result in more exploration and the algorithms will become close to random sampling. On the other side, a high value of δ will result in more exploitation and so, the algorithm will be much more confident about the first estimation of the mean values. The exploration is necessary because, in the pure exploitation setting, when having received 2 samples from each clusters, all the estimations are either 0, 1 or 0.5, sampling a 3rd point in a random cluster will change

the estimation of its mean value to a one with an increased corresponding loss, keeping it the next cluster to sample. This state will not change and this first random cluster would drain all the samples.

We first evaluated our algorithms with a low value for δ resulting in high probability bounds, which was intended. We ran our algorithms with $\delta = 0.1$. We display the results of this run on Figure 1(b).

A strange phenomenon appears in Figure 1(b), we see sort of steps. We remind here that we evaluate the algorithms via 1207 trials and display the average true risk. In fact, these steps are due to the lack of exploitation. Indeed, exploring implies to sample clusters which will not help to decrease the true risk. One could wonder why the arrangement in a way that it highlights two phases, one decreasing phase followed by a constant phase. First let us clear up the fact that this does not reveal an exploitation phase and an exploration phase, everything happens at the same time. But, the important exploration makes the clusters be sampled equally. One step (two phases) correspond to one number of samples in each cluster, i.e. we sample each cluster once then each cluster once again and so on. Now, inside a step the exploitation still plays a role, whereas the clusters are samples equally, the algorithm still starts by the most important clusters. Anyway, the steps are due to the fact that the exploration prevails on the exploitation. If we want to improve the algorithm we have to increase the value of δ.

We now evaluate our algorithms with $\delta = 0.9$. We display the results on Figure 1(c).

We can see that, apart from the first one, the steps have disappeared. The remaining step correspond to the sampling of the third sample in each cluster. This step remains because it is very hard to decide with very few information. Here, the algorithm has already taken 2 samples, so the possible values of the estimated means are 0, 1, and 0.5. Even the cluster requiring most samples will have such a value, thus, the algorithm cannot decide to leave aside one cluster and take a fourth sample while other clusters only have two. This is why this step in the risk is unavoidable.

An other thing we can see is that for this value of δ, the algorithm m binomial have an acceptable performance while s binomial perform poorly. To improve s binomial, we increase again the value of δ.

We now evaluate our algorithms with $\delta = 0.99999$. We display the results on Figure 1(d).

We can see that s binomial improved. One could ask if this value of δ is not exaggerated because its definition tells that it must have a small value, in order to derive high probability bounds. But, here, we see that the algorithms behave well with this value of δ. The only effect of a high value would be that the algorithm exploit to much, and the performance would be affected. Moreover, we can see that this value of δ is not well suited for CH-AS, which see its performance affected.

Finally, we can see that the algorithms built in this paper behave better than those designed for regression in [5] and [4].

5 Conclusion

The paper propose a method to use the *Optimism in the face of uncertainty* approach in an active learning problem for classification. It introduces two 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 clustering 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 ($\hat{\mu}(x)$ close to 0.5) because this will redefine it, whereas in our work the clustering remains the same. This leads to finding new loss functions. Our future work concern will be about an adaptive clustering of the space as well as the combination of the information providing from several different clustering .

References

1. P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.
2. M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic Active Learning. In *Proceedings of the International Conference on Machine Learning*, pages 65–72, 2006.
3. S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.
4. A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, and P. Auer. Upper-Confidence-Bound Algorithms for Active Learning in Multi-Armed Bandits. In *Algorithmic Learning Theory*, pages 189–203, 2011.
5. A. Carpentier and R. Munos. Finite Time Analysis of Stratified Sampling for Monte Carlo. In *Advances in Neural Information Processing Systems*, 2011.
6. D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.
7. A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Active Learning with Gaussian Processes for Object Categorization. In *International Conference on Computer Vision*, pages 1–8, 2007.
8. D. Lewis and W. Gale. A Sequential Algorithm for Training Text Classifiers. In *SIGIR Conference*, pages 3–12, 1994.
9. M. Naghshvar, T. Javidi, and K. Chaudhuri. Noisy Bayesian Active Learning. In *Allerton Conference on Communication, Control, and Computing*, pages 1626–1633, 2012.
10. R. Nowak. Generalized Binary Search. In *Allerton Conference on Communication, Control, and Computing*, pages 568–574, 2008.
11. R. Nowak. Noisy Generalized Binary Search. In *Advances in Neural Information Processing Systems*, pages 1366–1374, 2009.
12. B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
13. H.S. Seung, M. Opper, and H. Sompolinsky. Query by Committee. In *Proceedings of Computational Learning Theory*, pages 287–294, 1992.
14. W. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933.