A Memory-Restricted Learning Algorithm

Badih Ghattas *  Gonzalo Perera†

January 24, 2004

Abstract

In this paper we present an algorithm for supervised learning under severe memory restrictions: at each step of the learning process, the total amount of information to be kept in memory must be smaller than a fixed quantity $M$. This is of particular interest for applications where data correspond to signals or very high-dimensional vectors. Our algorithm proceeds as a sequence of cycles: for each cycle, a training sample sequence of fixed size $T$ is available and a prediction rule is chosen from the combination of a given class of statistical models $\mathcal{F}$ and a set of experts. The choice is made by means of a matrix of credit scores $c_j(x, h)$, which measures, at the cycle $j$, how much should we trust on the advice of predictor $h$ to decide the predicted value to be assigned to the input $x$. The particular choice of the predictor $f_j$ corresponding to the model $\mathcal{F}$ at cycle $j$ is made by maximization of the mean of a given reward function $R$ (or minimization of a cost function). A fixed validation sample of size $V$ is also assumed to be available, and for the validation of the procedure at cycle $j$, each value of $x$ corresponding to the evaluation sample is given as an input to the prediction rule and its performance is measured by means of the reward function $R$. The credit to be assigned to each predictor $h$ for step $j + 1$ at the point $x$ is the sum of $c_j(x, h)$ and the overall reward obtained for the predictor $h$ when it was used to predict the value corresponding to the input $x$. Once this is done, only the matrix of

---

*Institut de Mathématiques de Luminy, CNRS, Marseille, France, e-mail: ghattas@lumimath.univ-mrs.fr
†Institut de Mathématiques de Luminy, CNRS, Marseille, France and Universidad de la República, Montevideo, Uruguay. e-mail: perera@iml.univ-mrs.fr, gperera@fing.edu.uy.
credits, the class $\mathcal{F}$ and the last choice of the model will be kept in memory; any other data will be removed and a new training sample corresponding to a new iteration of the algorithm will be used. No use of an arbitrary large training sample is allowed. The asymptotic behaviour of our algorithm is derived and in particular its performance is compared to that of a similar procedure without memory restriction (when arbitrary large training samples are available).
1 Introduction: basic ideas on Supervised Learning Algorithms.

The main purpose of this paper is to learn how to predict the value of a random variable \( Y \) in terms of the values of a set of explicative random variables, denoted by \( X \), under severe restrictions on the capacity of storage of data.

Let us first briefly explain the general framework for Supervised Learning Algorithms.

As usual, let \( P \) denote the probability on an underlying probability space \((\Omega, \mathcal{A})\), where the couple \((X, Y)\) is defined, where \( X \) takes values in an arbitrary measurable space \( S_X \) and \( Y \) takes values on a measurable space \( S_Y \).

Some notation we will use:

- \( \rho \) is the joint distribution of the couple \((X, Y)\), that is, for any measurable sets \( A \subset S_X \) and \( B \subset S_Y \), we have
  \[
  \rho(A \times B) = P(X \in A, Y \in B).
  \]

- \( p(\cdot | X) \) denotes the conditional probability distribution of \( Y \) given \( X \), defined in a rigorous way as the almost surely (a.s., for short) unique measurable function of \( X \) satisfying:
  \[
  E(1_{\{Y \in B\}}1_{\{X \in A\}}) = E(p(B/X)1_{\{X \in A\}})
  \]
  for any measurable sets \( A \) and \( B \).

- \( \pi \) is the marginal law of \( X \) (i.e., \( \pi(A) = P(X \in A) \)); then we can write
  \[
  \rho(A \times B) = \int_A p(B/x)\pi(dx) = \int_A \int_B p(dy/x)\pi(dx).
  \]
We will assume that both $\rho$ and $p$ are unknown. In this paper we will also assume that $\pi$ is unknown but the key point of the prediction problem concerns the process of learning $p$ (and hence, $\rho$).

In the prediction problem, we will observe the value of $X(\omega)$ and we are requested to guess the value of $Y(\omega)$. We will often call $X$ the input or the pattern, and $Y$ output or label. In general, our prediction will be $f(X(\omega))$ where $f$ is a measurable function from $S_X$ on $S_Y$, that we will call predictor. The main problem is to find a “good” predictor, what previously requires to set a criterion to determine whether a given predictor is “good” or not. If we have a criterion to quantify how much we “lose” by predicting a value $u$ for $X(\omega) = x$ where the true value was $Y(\omega) = y$ and if this quantification is denoted by $L(x, u, y)$, we may introduce the loss function

$$L: S_X \times S_Y \times S_Y \implies \mathbb{R}.$$ We will assume in the sequel that $L(x, u, y) \geq 0$ and that $L(x, u, y) = 0$ if and only if $u = y$.

From now on $(X,Y)$ will denote a generic random vector distributed according to $\rho$. As said before, a predictor or prediction rule is in general a measurable function $f : S_X \implies S_Y$ and its quality is measured by means of:

$$\tau_L(f) = E\{L(X, f(X), Y)\} = \int_{S_X} \int_{S_Y} L(x, f(x), y)\rho(dx, dy) = \int_{S_X} \left(\int_{S_Y} (L(x, f(x), y))p(dy/x)\right)\pi(dx).$$

Hence, we will say that a predictor $f$ is better than a predictor $g$ if $\tau_L(f) \leq \tau_L(g)$.

For instance, if we take $L(x, u, y) = 1_{\{u \neq y\}}$ then $\tau_L = P(f(X) \neq Y)$ (the overall error rate).

As another classical example if $S_Y$ is a normed space and $||.||$ denotes its norm, if we assume that $E\{||Y||^2\} < \infty$ and $L(x, u, y) = ||u - y||^2$, then $\tau_L = E\{(f(X) - Y)^2\}$ (the mean integrated squared error, MISE, for short).

In general, if we assume that, for any $x$ in a set of $\pi$-probability one, there exists an unique value $f^*(x)$ such that

$$\int_{S_Y} L(x, f^*(x), y)p(dy/x) \leq \int_{S_Y} L(x, u, y)p(dy/x) \text{ a.s. with respect to } u \in S_Y,$$
and if the function \( f^* : S_X \rightarrow S_Y \) is measurable, then \( f^* \) is the optimal predictor, since a straightforward computation shows that \( \tau_L(f^*) \leq \tau_L(f) \) for any predictor \( f \).

In the case of the overall error rate and when \( p(.|x) \) is unimodal, \( f^*(x) \) is called conditional mode or Maximum A Posteriori (MAP) given \( x \), and corresponds to the value of \( u \) that maximizes \( p(u|x) \). In the case of the MISE, \( f^*(X) = E(Y/X) \). In general, if for instance \( S_Y \) is a topological vector space and \( L \) satisfies some regularity conditions with respect to \( u \), the existence of \( f^*(x) \) can be shown.

The problem is that, one is not able to look for a predictor on the whole set of functions from \( S_X \) to \( S_Y \) (that may be a very huge set) but only on a given class of functions \( \mathcal{F} \) that corresponds to the kind of predictors that we may practically compute. In such a case, the optimal predictor \( f^* \) may benot included in \( \mathcal{F} \) and , therefore, the best predictor that we will be able to find is \( f^{**} \), such that

\[
 f^{**} = \arg\min_{f \in \mathcal{F}} \tau_L(f). 
\]

As we will see later, this predictor \( f^{**} \) is not available in practice, since the law \( \rho \) is unknown and should be estimated from data, what makes that in practice one is not able to minimize \( \tau_L \) but only an empirical estimation of it.

**Remark 1.1:** It is also a common practice (and in fact, this will be our case), to consider a reward function \( R(x, u, y) \), that gives the reward to be assigned to a prediction of \( u \) for the value \( x \) when the real value is \( y \), instead of the loss function. Despite of the fact that the practical motivation may make one approach more appealing than the other, from the mathematical point of view, they are completely equivalent, since if \( L \) is a loss function and \( C \) is a suitable constant, then \( C - L \) is a reward function.

**Remark 1.2:** It is also common (and wise), in practice, to make use of the advice of experts, that may be human experts or previously tested algorithms. In our case we will use experts advice, and we will think an expert as a transition matrix \( A(.|.) \), that gives, for any input \( x \) the probability \( A(y|x) \) that this expert assigns to the output \( y \). If an expert is used just by means of a MAP procedure, then, we will consider that his answer for an input value
\( x \) is the (unique) value \( a(x) \in S_X \) such that \( A(a(x)/x) \geq A(y/x) \) for any \( y \) (if such an \( a(x) \) is not unique, then some ordering or sampling procedure may be used to choose only one).

In \textit{supervised learning}, the predictor \( \hat{f}_n \) that we can use in practice, is based on a \textit{training sample} \((X_1, Y_1), ..., (X_n, Y_n)\), often assumed to be independent and identically distributed (\textit{iid}, for the sequel) according to the law \( \rho \). If a class \( \mathcal{F} \) of functions is used, then we will take as our prediction rule \( \hat{f}_n \), the element of \( \mathcal{F} \) that minimizes
\[
\tau_{L,n}(f) = \frac{1}{n} \sum_{i=1}^{n} L(X_i, f(X_i), Y_i),
\]
i.e.
\[
\hat{f}_n = \text{argmin}_{f \in \mathcal{F}} \tau_{L,n}(f).
\]
As an estimation of the performance of this prediction rule, we might take \( \tau_{L,n}(\hat{f}_n) \), but this is usually a biased estimation: it overestimates the performance of the predictor rule and may induce to believe that the predictor behaves better than what it really does when applied to new, fresh, data. This is also related to what is usually called “overfitting”: if, for instance, \( \mathcal{F} \) is as big as the whole family of functions from \( S_X \) to \( S_Y \), and \( X_1, ..., X_n \) contains \( n \) different values, it is clear that \( \hat{f}_n(X_i) = Y_i \) for any \( i \) and that \( \tau_{L,n}(\hat{f}_n) = 0 \) (perfect fitting over the training sample), but when applied to new data the results may be catastrophic (the prediction follows so closely the particular features of the training sample, that is statistically very poor).

This type of problems will be detected if the performance of the predictor rule is measured by means of a new sample, called the \textit{evaluation sample} \( \varepsilon \), which is another \textit{iid} sample of the distribution \( \rho \), \((X_1^\varepsilon, Y_1^\varepsilon), ..., (X_m^\varepsilon, Y_m^\varepsilon)\), independent with respect to the training sample, and the performance of our predictor is estimated by means of
\[
\tau_{L,m}^\varepsilon(f) = \frac{1}{m} \sum_{i=1}^{m} L(X_i^\varepsilon, f(X_i^\varepsilon), Y_i^\varepsilon).
\]
Another type of performance estimation, based on well-known procedures such as cross-validation, bootstrap and other resampling techniques, may be used in practice to give unbiased and numerically efficient estimations, but
we refer to [6] for an extensive account.

Finally, it should be noticed that when the best of all predictors is \( f^* \) and the best of possible predictors on our class is \( f^{**} \), the real predictor we will use in practice is \( \hat{f}_n \). The loss of performance due to the difference between \( f^* \) and \( f^{**} \) is of a modellistic nature, it depends on how clever is our choice of \( \mathcal{F} \). If a bad choice of \( \mathcal{F} \) is made, no further sampling will allow to overcome this loss of performance. This is why the difference \( f^* - f^{**} \) is often called approximation error. On the other hand, the second loss of performance, due to the difference between \( f^{**} \) and \( \hat{f}_n \) is purely of statistical nature. If very large training samples were available (i.e., if \( n \) tends to infinity), under suitable hypothesis on our model, \( \hat{f}_n \) will tends to \( f^{**} \). This explains why the difference \( f^{**} - \hat{f}_n \) is often called estimation error. Indeed,

\[
\tau_L(f) = E\{L(X, f(X), Y)\}; \tau_{L,n}(f) = E_n\{L(X, f(x), Y)\},
\]

where \( E_n \) denotes expectation with respect to the empirical distribution \( F_n \) defined for any \( C \subset S_X \times S_Y \), \( C \) measurable, as:

\[
F_n(C) = \frac{1}{n} \sum_{i=1}^{n} 1_{\{(X_i, Y_i) \in C\}}.
\]

Let us denote \( L_\mathcal{F} \), the class of all the functions \( g : S_X \times S_Y \rightarrow \mathbb{R} \) such that there exists \( f \in \mathcal{F} \) for which \( g(x, y) = L(x, f(x), y) \) for any \( x, y \). If \( \mathcal{F} \) is such that the associated class of functions \( L_\mathcal{F} \) is a Glivenko-Cantelli class (defined as a classe of functions for which an uniform law of large number holds), i.e,

\[
\lim\sup_{n \to \infty} \left| E_n\{g(X, Y)\} - E\{g(X, Y)\} \right| = 0 \ a.s.,
\]

then since

\[
f^{**} = \arg\min_{f \in \mathcal{F}} E\{L(X, f(X), Y)\} = \arg\min_{g \in L_\mathcal{F}} E\{g(X, Y)\}
\]

\[
\hat{f}_n = \arg\min_{f \in \mathcal{F}} E_n\{L(X, f(X), Y)\} = \arg\min_{g \in L_\mathcal{F}} E_n\{g(X, Y)\}
\]

it is clear that \( \hat{f}_n \) will converge \( a.s. \) to \( f^{**} \). Exponential inequalities (Bernstein, Hoeffding, and others) and Vapnik-Chervonenkis theory on the dimension of a class of predictors are widely used tools to prove that a uniform law of large numbers holds, and we refer the reader to [4], [5], [10], for a detailed account of these techniques.
2 General description of the algorithm.

We will now describe the particular characteristics of our learning algorithm and its motivation. For the sake of simplicity, we will assume from now on that $X$ takes values in a finite set $S_X = \{1, \ldots, I\}$ and that $Y$ takes values in $S_Y = \{1, \ldots, J\}$. Despite the huge variety of procedures that have been proposed for supervised learning (cite linear, neurons, CART, SVM, Boosting) most of them do work well in many situations under the condition that the size of the training sample ($n$), is assumed to be arbitrary large. This means that massive information is available, allowing to drastic reduction of the estimation error and, with suitable modelling, very efficient learning (see, for instance, [4],[5], [6], [9]).

In some practical situations, however, this approach is no longer possible. For instance, for the implementation of an Acceptancy Policy in a Telecommunication Network dealing with VBR (Variable Bit Rate) workload, the "size" of a task must be estimated before deciding if it may be processed in the given network or not. In general, tasks are grouped in terms of classes, and available information concerning estimated size of each class of tasks will be used in the decision. To predict required sources for each class of tasks, a learning algorithm is often used, where training samples consist of many cases of tasks of different types (hence each $X$ is a whole signal consisting of the time series of bytes/time unit requested by the task during its execution) and the label $Y$ corresponds to a class of performances (in the most simple case, $Y$ is binary and $Y = 1$ means that the task requires too many sources and should be rejected, whereas $Y = 0$ means that this task is viable for the given network). Each $X$ being a signal, its storage takes a significative amount of memory and the fact that the variation of the performance according to time usually presents complex patterns, makes it in general very difficult to reduce its dimensionality by a clever selection of some parts of the signal. Therefore, if very large samples $X_1, \ldots, X_n$ must be stored to perform the learning algorithm, even under reasonable technological facilities, the memory required turns out to be out of reach. At this point, one of the most popular solutions consist on reduce the information that is taken in account from each $X$, reducing the whole signal to just some few parameters (mean, deviation, extremes, effective bandwidth estimations, transition matrix when a Markov model is possible, etc.). But in this approach more subtle stochastic characteristics of signal will be often disregarded and hence, effi-
ciency may be low (see, for instance, [1], [3], [7], [8]).

In this paper we take the inverse way. No particular simplification of \( X \) is made, but the size of training samples and the complexity of predictors will be limited. Our algorithm will deal with a sequence of training steps, indexed by \( j \), but at each step a limited quantity of information will be used. After each step, training data will be forgotten, keeping in memory only some parameters of the prediction system.

As predictors, we will have at hand \( k \) experts \( A_1, \ldots, A_k \) and a class of models \( \mathcal{F} \). The experts will be fixed and will not change their behaviour through the whole process: given one expert \( A_i \) and an input \( x \) at any step of the algorithm, the expert will always give the same advice and predict the same value of \( y \). We will call advisors, both experts and the optimal predictor chosen from \( \mathcal{F} \). Hence, we will have \( k + 1 \) advisors, where \( 1, \ldots, k \) correspond to the experts and \( k + 1 \) to the model. It must be noticed that while \( A_1, \ldots, A_k \) do not change over the whole execution of our algorithm the specific function \( f_j \) selected in \( \mathcal{F} \) at step \( j \), will change from one step of the algorithm to the following one. This will not be done by means of a simple adaptive scheme that gives a closed formula for \( f_{j+1} \) given \( f_j \), but by means of an optimization routine.

We will assume that a family of \( k + 1 \) reward functions is given. More precisely, denote \( H = \{1, \ldots, k + 1\} \); we will consider a function \( R : S_X \times S_Y \times S_Y \times H \rightarrow \mathbb{R} \) such that for any \( h = 1, \ldots, k \) \( R(\cdot, \cdot, \cdot, h) \) is a reward function with \( R(x, u, y, h) > 0 \) if \( u = y \) and \( R(x, u, y, h) < 0 \) if \( u \neq y \). \( R(x, u, y, h) \) will represent the reward to be assigned to the advisor \( h \) if he assigns for \( X = x \) the value \( u \) when the true value was \( Y = y \).

A key role in our algorithm is played by the credit matrix.

\[
(c_j(x, h))_{x \in S_X, h \in H}
\]

which encodes for the step \( j \) our confidence in the advisor \( h \) to predict the output of \( x \). More precisely denote:

\[
h_j(x) = \arg \max_{h \in H} c_j(x, h),
\]

the most credible advisor to predict \( x \) at step \( j \) (if there is more than one value of \( h \) where the maximum is reached, we may choose for instance the biggest of such values). Then:
At step \( j \), the prediction of the value to assign to \( x \) will be made by the advisor \( h_j(x) \) (recalling that the case \( h_j(x) = k + 1 \) corresponds to the model, i.e., \( \hat{f}_j(x) \)).

Once the training sample to be used at the step \( j \) is available, \((X^{(j)}_1, Y^{(j)}_1), \ldots, (X^{(j)}_T, Y^{(j)}_T)\) (which is assumed to be \( \text{iid} \) and to follow the law \( \rho \)), we choose \( f_j \) as the best candidate in \( \mathcal{F} \) according to the following criteria:

\[
\hat{f}_j = \arg\max_{f \in \mathcal{F}} \Gamma_{k+1,T}^j(f)
\]

where,

\[
\Gamma_{k+1,T}^j(f) = \frac{1}{T} \sum_{i=1}^T R(X_i^j, f(X_i^j), Y_i^j, k + 1)
\]

is the empirical version of the expected reward

\[
\Gamma_{k+1}(f) = \mathbb{E}[R(X, f(X), Y, k + 1)] = \sum_{x \in S_X, y \in S_Y} R(x, f(x), y, k + 1) \rho(x, y)
\]

(\( \Gamma \) is analogous to the the expected loss \( \tau_L \) if we think in terms of a loss function \( L \) instead of a reward function).

**Remark 2.1:** Even if our method has been inspired from learning problems on network administration, where a merely objective learning seems to be adequate, it may be appealing to apply this system in a subjective context (for instance, for behavioural systems). In that context, in order to allow the expression of subjective profiles, that the reward to be credited to different advisors by a given decision may be different (further, this difference on the credited reward may be taken as patterns to identify such profiles). That is why we have included the “h” component on the function \( R \) and that is why we prefer to speak about “reward” instead of “loss” or “cost”.

Once the model has been fitted, we will proceed to the validation of the prediction rule and we will update the credit matrix. Since each expert \( A_1, \ldots, A_k \) uses a MAP criterion, we will denote by \( y_1(x), \ldots, y_k(x) \) the answer that each expert gives to the input \( x \). In the cycle \( j \) of our algorithm, we use a validation sequence \((X^{(j)}_1Y^{(j)}_1), \ldots, (X^{(j)}_pY^{(j)}_p)\) (\( \text{iid} \) and distributed according to \( \rho \)), independent with respect to the training sequence of the same cycle \( j \) and independent with respect to both training and validation samples of
previous cycles.

Then the credits will be updated as following:

- For each \( i = 1, ..., V \), compute \( h_j(X_i^{v,j}) \).
- Compute the prediction for each observation of the validation sequence by means of \( y_h(X_i^{v,j}) \) if \( h_j(X_i^{v,j}) = h \leq k \) or by means of \( \hat{f}_j(X_i^{v,j}) \) if \( h_j(X_i^{v,j}) = k + 1 \). In any case, let us denote by \( f_j(X_i^{v,j}) \) the predicted output.
- Update the credits as follows

\[
c_{j+1}(x, h) = c_j(x, h) + \frac{1}{V} \sum_{i=1}^{V} R(X_i^{v,j}, f_j(X_i^{v,j}), Y_i^{v,j}, h) 1\{h_j(X_i^{v,j}) = h, X_i^{v,j} = x\}
\]

**Remark 2.2:** Observe that to update the credit \( c_j(x, h) \) we only use the observations of the validation sample where the input was \( x \) and the most credible expert was \( h \). In particular, if a value of \( x \) doesn’t appear in the validation sample, its credit is not changed, and if a given expert was less credible than others for any input of the validation, its credit not change.

Some final remarks on general notation: we will use the symbol ":=" for a definition that is set inside an equation. If \( \mathcal{C} \) is a collection of random variables, \( \sigma(\mathcal{C}) \) will denote the \( \sigma \)-algebra generated by \( \mathcal{C} \). If \( \mathcal{F}, \mathcal{L} \) are \( \sigma \)-algebras on \( \Omega \), \( \mathcal{F} \vee \mathcal{L} := \sigma(\mathcal{F} \cup \mathcal{L}) \). As usual, convergence in law may be thought both at the level of random variables or at the level of probability distributions and notation may mix both levels. For instance, if \( Z_1, ..., Z_n, ... \) is a sequence of random variables and we state

\[
\lim_{n} Z_n = N(0, 1) \text{ in law ,}
\]

we are saying that, with respect to the topology of the weak convergence of probability measures, the sequence of distribution measures \( P_{Z_n} \) converges to a standard gaussian probability measure.
3 Theoretical results.

In this section we will derive the asymptotic behaviour of our memory-restricted learning algorithm (MRLA, for short), when the number of iterations tends to infinity.

We will first obtain the limit of the credit matrix \( (c_j(x, h))_{x \in S_X, h \in H} \) when \( j \) tends to infinity. Then we will compare the performance of the MRLA to that of an algorithm based on the whole set of training sequences (i.e., with No Memory Restriction: we will call this algorithm NMR, for short). In particular we will show that, under reasonable assumptions, MRLA behaves almost as well as NMR and therefore it can be seen as a performant alternative that respects memory restrictions.

Let us set some notation and assumptions.
First of all, to avoid trivialities, we will assume that \( \pi(x) > 0 \) for any \( x \in S_X \). For each one of the experts indexed by \( h = 1, ..., k \) and any \( x \in S_X, y \in S_Y \), we will define

\[
\tau_h(x, y) = R(x, y_h(x), y, h)
\]

\[
\tau_h(x) = E\{\tau_h(X, Y) / X = x\} = \sum_{y \in S_Y} \tau_h(x, y)p(y/x)
\]

With the notation of the end of the previous section, denote

\[
f^{**} = \arg\max_{f \in \mathcal{F}} \Gamma_{k+1}(f), \hat{f}_j = \arg\max_{f \in \mathcal{F}} \Gamma^j_{k+1,T}(f).
\]

(We assume again that there those maximum values are attained at a unique element of \( \mathcal{F} \).

Define, for any \( x \in S_X, y \in S_Y \):

\[
\tau_{k+1}(x, y) = R(x, f^{**}(x), y, k + 1)
\]

\[
\tau_{k+1}(x) = E\{\tau_{k+1}(X, Y) / X = x\}
\]

\[
\tau^j_{k+1}(x, y) = R(x, \hat{f}_j(x), y, k + 1)
\]

\[
\tau^j_{k+1}(x) = \sum_{y \in S_Y} \tau^j_{k+1}(x, y)p(y/x)
\]

Observe that \( \tau^j_{k+1}(x, y) \) (resp. \( \tau^j_{k+1}(x) \)) is a random function of \((x, y)\) (resp. \(x\)).
Let us also call $\mathcal{S}$ the set of all the functions from $S_X$ to $S_y$. If $(X,Y)$ is random vector independent of the training sample and distributed according to $\rho$, we have that:

$$E\left(r^j_{k+1}(X)\right) = E\left(E\{R(X, \hat{f}_j(X), Y, k+1)/\hat{f}_j\}\right)$$

$$= \sum_{f \in \mathcal{S}} E\{R(X, \hat{f}_j(X), Y, k+1)/\hat{f}_j = f\} P(\hat{f}_j = f)$$

$$= \sum_{f \in \mathcal{S}} E\{R(X, f(X), Y, k+1)\} P(\hat{f}_j = f)$$

$$= \sum_{f \in \mathcal{F}} \Gamma_{k+1}(f) P(\hat{f}_j = f) \tag{1}$$

In the last equality, we have used the fact that $\hat{f}_j \in \mathcal{F}$.

$$E(r^j_{k+1}(x)) = \sum_{f \in \mathcal{F}} \sum_{y \in S_y} R(x, f(x), y, k+1)p(y|x)P(\hat{f}_j = f).$$

We will assume now on the following hypothesis:

- (H1) For any $x \in S_x$ there exists an unique $h(x) \in H$ such that $r_{h(x)}(x) > 0$, $r_h(x) < 0$ if $h \neq h(x)$.

**Remark 3.1:** Observe that if for a given value $x_0$, $r_h(x_0)$ depends only on the values of $R(x_0, \ldots, h)$. Hence, if $r_h(x_0) < 0$ for all the values of $h$, but there is only one $h$ corresponding to the maximum value $\max_{h \in H} r_h(x_0)$, then, we can find a suitable constant $C$ and modify $R$ by means of $R_{mod}(x, u, y, h) = R(x, u, y, h) + C1_{\{x=x_0\}}$, for any $x, u, y, h$ in such a way that for $R_{mod}$ assumption (H1) holds true (and no change is introduced on the rewards for other values of $x$). Therefore, (H1) essentially means that for any $x$, there is only one value of $h$ that maximizes $r_h(x)$. In practice, this is not a major restriction, since, again, this can be obtained by means of minor modifications of $R$ and a fixed procedure to choose one $h$ in case of “ties”.

Taking into account (H1), the following sets are well-defined

$$D_h = \{x \in S_X : r_h(x) > 0\}, \; h = 1, \ldots, k + 1.$$
and we have that

\[ \bigcup_{h=1}^{k+1} D_h = S_X, \quad D_h \cap D_l = \emptyset \text{ if } h \neq l. \]

The following lemma will play a key role in the rest of the paper. In two items of this lemma we will consider that \( T \) (size of the learning sequence) goes to infinity; since for the rest of the paper \( T \) will be fixed, we do not emphasize the dependence on \( T \) of \( \hat{f}_j, r_{k+1}^j, \).

We will also denote

\[
\gamma_{k+1}(x) := \sum_{f \in \mathcal{F}} \sum_{y \in S_Y} R(x, f(x), y, k + 1)p(y/x)P(f^1 = f)
\]

**Lemma 3.1** Let \( \mathcal{F} \) be any class of functions from \( S_X \) on \( S_Y \) and let \( (X, Y) \) be a random vector, independent of the training sequence, distributed according to \( \rho \). We have then that:

- **i)** For any \( j \), \( \lim_T \hat{f}_j(X) = f^{**}(X) \) in law
- **ii)** There exists a sequence of non-negative real numbers, \( (a(T))_{T \in \mathbb{N}} \)

such that \( \lim_T a(T) = 0 \) and, for any \( j \),

\[ E \left\{ \left( r_{k+1}^j(X) - r_{k+1}(X) \right)^2 \right\} \leq a(T). \]

- **iii)** Fix now \( T \): for any \( x \), \( \lim_n \frac{1}{n} \sum_{j=1}^n r_{k+1}^j(x) = \gamma_{k+1}(x) \) a.s.
- **iv)** With the same notation as above,

\[ E \left\{ \left( \gamma_{k+1}(X) - r_{k+1}(X) \right)^2 \right\} \leq a(T), \]

and for any \( x \in S_X \),

\[ |\gamma_{k+1}(x) - r_{k+1}(x)| \leq \left( \frac{a(T)}{\pi(x)} \right)^{\frac{1}{2}}. \]

**Proof:**

Fix \( j \). Observe first that

\[
\Gamma_{k+1, T}^j(f) - \Gamma_{k+1}^j(f) = \sum_{x \in S_X, y \in S_Y} R(x, f(x), y, k + 1)(\rho_{T}(x, y) - \rho(x, y)),
\]
where
\[ \rho_T^j(x, y) = \frac{1}{T} \text{Card}\{ i : 1 \leq i \leq T : X_i^j = x, Y_i^j = y \}. \]

Since \( E\{ \rho_T^j(x, y) \} = \rho(x, y) \) for any \( x, y \), we deduce that
\[ E\{ \Gamma_{k+1,T}(f) \} = \Gamma_{k+1}(f). \]  
(2)

In addition, by the Law of Large Numbers,
\[ \lim_{T} |\rho_T^j(x, y) - \rho(x, y)| = 0, \text{ for any } j, x, y, \]
and, therefore
\[ \lim_{T} \max_{(x,y) \in S_x \times S_y} |\rho_T^j(x, y) - \rho(x, y)| = 0, \]
what implies in turn that
\[ \lim_{T} \max_{f \in \mathcal{F}} |\Gamma_{k+1,T}^j(f) - \Gamma_{k+1}(f)| = 0 \text{ a.s.} \]  
(3)

Since the domain of \( R \) is finite, \( R, \Gamma_{k+1} \), and \( \Gamma_{k+1,T} \) are bounded, and hence
\[ \lim_{T} E\{ \max_{f \in \mathcal{F}} |\Gamma_{k+1,T}^j(f) - \Gamma_{k+1}(f)| \} = 0. \]  
(4)

Define
\[ C = \max_{f \in \mathcal{F}, f \neq f^{**}} \Gamma_{k+1}(f). \]

Since this maximum is taken over a finite number of functions taking values on a finite set and \( f^{**} \) is the only global maximum of \( \Gamma_{k+1} \), we have that
\[ C < \Gamma_{k+1}(f^{**}) \]

Since \( \mathcal{F} \) is a finite set of finite-valued functions, it is compact with respect to the topology of the pointwise convergence, and the sequence of random functions \( f^j \) is pre-compact with respect to the topology of the convergence in distribution. Call \( \mathcal{LP} \) the set of probability measures on \( S \) that are obtained as limit in distribution of subsequences (with respect to \( T \)) of \( \tilde{f}_j \). Since \( \tilde{f}_j \) belongs to the closed set \( \mathcal{F} \), all the elements of \( \mathcal{LP} \) are concentrated in \( \mathcal{F} \). Let \( \nu \) be an element of \( \mathcal{LP} \) and \( g \) be a random element of \( \mathcal{F} \) that is
distributed according to \( \nu \). Let \( \hat{f}^T_j, \ldots, \hat{f}^T_n, \ldots \) denote the subsequence of \( \hat{f}_j \) that converges in law to \( g \). We have that:

\[
\Gamma_{k+1}(\hat{f}^T_j) = \sum_{x \in S} \sum_{y \in S} R(x, \hat{f}^T_j(x), y, k + 1) \rho(x, y) \\
= \sum_{f \in \mathcal{F}} \sum_{x \in S} \sum_{y \in S} R(x, f(x), y, k + 1) \rho(x, y) 1_{\{f^T_j = f\}} \\
= \sum_{f \in \mathcal{F}} \Gamma_{k+1}(f) 1_{\{f^T_j = f\}}
\]

Hence,

\[
E\{\Gamma_{k+1}(\hat{f}^T_j)\} = \sum_{f \in \mathcal{F}} \Gamma_{k+1}(f) P(\hat{f}^T_j = f),
\]

and it follows that

\[
\lim_{n} E\{\Gamma_{k+1}(\hat{f}^T_j)\} = \sum_{f \in \mathcal{F}} \Gamma_{k+1}(f) P(g = f)
\]

(6)

It is clear that

\[
\sum_{f \in \mathcal{F}} \Gamma_{k+1}(f) P(g = f) \leq CP(g \neq f^{**}) + \Gamma_{k+1}(f^{**}) P(g = f^{**}) := D
\]

Assume that \( P(g \neq f^{**}) > 0 \); then \( D < \Gamma_{k+1}(f^{**}) \). Let \( \delta > 0 \) be such that \( \delta < \Gamma_{k+1}(f^{**}) - D \).

By (6), if \( n \) is large enough we have that

\[
E\{\Gamma_{k+1}(\hat{f}^T_j)\} < D + \frac{\delta}{3}.
\]

(7)

Observe now that

\[
\Gamma_{k+1, T_n}(\hat{f}^T_j) - \Gamma_{k+1}(\hat{f}^T_j) = \sum_{f \in \mathcal{F}} (\Gamma_{k+1, T_n}(f) - \Gamma_{k+1}(f)) 1_{\{f^T_j = f\}},
\]

and therefore

\[
|\Gamma_{k+1, T_n}(\hat{f}^T_j) - \Gamma_{k+1}(\hat{f}^T_j)| \leq \max_{f \in \mathcal{F}} |\Gamma_{k+1, T_n}(f) - \Gamma_{k+1}(f)|.
\]

(8)

\( \exists \)From (4) and (8) we obtain

\[
\lim_{n} |E\{\Gamma_{k+1, T_n}(\hat{f}^T_j)\} - E\{\Gamma_{k+1}(\hat{f}^T_j)\}| = 0.
\]

(9)
Using (2), (7) and (9), we have that for $n$ large enough

$$E\{\Gamma_{k+1,T_n}^{j}(\hat{f'}^T_j)\} < D + \frac{2\delta}{3} < \Gamma_{k+1}(f^{**}) - \frac{\delta}{3} = E\{\Gamma_{k+1,T_n}(f^{**})\} - \frac{\delta}{3} \quad (10)$$

We conclude that for $n$ large enough

$$E\{\Gamma_{k+1,T_n}(\hat{f'}^T_j)\} < E\{\Gamma_{k+1,T_n}(f^{**})\},$$

what contradicts the fact that

$$\Gamma_{k+1,T_n}(\hat{f'}^T_j) \geq \Gamma_{k+1,T_n}(f^{**}) \text{ a.s.}$$

Thus,

$$P(g \neq f^{**}) = 0,$$

what proves (i).

Let us now prove (ii). Since $R$ is bounded, both $\tau_{k+1}(X)$ and $\tau_{k+1}^j(X)$ are bounded. Thus, if $\tau_{k+1}^j(X)$, as $T$ goes to infinity, converges in probability to $\tau_{k+1}(X)$, it also converges in $L^2$, and

$$\lim_T E\{(\tau_{k+1}^j(X) - \tau_{k+1}(X))^2\} = 0.$$ 

But, by its definition, $\hat{f_j}$ is obtained by a fixed maximization procedure applied to the training sequence $(X_1^j, Y_1^j), ..., (X_T^j, Y_T^j)$, which is iid and distributed according to $\rho$. Therefore, the law of $\hat{f_j}$ depends only on $\rho$, $R$ and $T$, but it does not depend on $j$. This implies that

$$E\{(\tau_{k+1}^j(X) - \tau_{k+1}(X))^2\}$$

does not depend on $j$, and setting

$$a(T) = E\{(\tau_{k+1}^j(X) - \tau_{k+1}(X))^2\},$$

part (ii) follows.

Therefore, it is enough to show that convergence in probability holds. Conditioning with respect to $X = x$, and using that $\tau_{k+1}(x)$ is deterministic, it is clear that it suffices to show that $\tau_{k+1}^j(x)$ converges in distribution to
\( r_{k+1}(x) \). Let \( B \) be the finite set corresponding to the values that \( R \) may take. For any \( b \in B \), denote
\[
C_b = \{ f \in \mathcal{F} : \sum_{y \in S_Y} R(x, f(x), y, k + 1)p(y|x) = b \}.
\]
Then
\[
P(r^j_{k+1}(x) = b) = P(\hat{f}_j(x) \in C_b),
\]
and, by (i),
\[
\lim_{T} P(\hat{f}_j \in C_0) = 1_{\{f^*_{n+1} \in C_0 \}} = 1_{\{\sum_{y \in S_Y} n(x, f^*(x), y, k+1)p(y|x) = b \}} = 1_{\{r_{k+1}(x) = b \}},
\]
what proves the convergence in distribution.

For (iii), fix \( x \). Then, \( r^j_{k+1}(x) \), as said before, is a given function of the training sequence corresponding to the cycle \( j \). Since training sequences of different cycles are independent and follow the same distribution on the set of sequences of size \( T \), we have that \( r^1_{k+1}(x), ..., r^n_{k+1}(x), ... \) is an iid sequence, with mean
\[
E\{r^j_{k+1}(x)\} = \sum_{f \in \mathcal{F}} \sum_{y \in S_Y} R(x, f(x), y, k + 1)p(y|x)P(\hat{f}_j = f)
\]
Since the law of \( \hat{f}_j \) does not depend on \( j \) we conclude that
\[
E\{r^j_{k+1}(x)\} = \gamma_{k+1}(x)
\]
and (iii) follows from the Law of Large Numbers.

Finally, using (iii)
\[
E\{(\gamma_{k+1}(X) - r_{k+1}(X))^2\} = \lim_{n} E\left\{ \left( \frac{1}{n} \sum_{j=1}^{n} (r^j_{k+1}(X) - r_{k+1}(X)) \right)^2 \right\}
\]
\[
= \left( \frac{1}{n} \sum_{j=1}^{n} ||r^j_{k+1}(X) - r_{k+1}(X)||^2 \right)^2
\]
\[
\leq \left( \frac{1}{n} \sum_{j=1}^{n} ||r^j_{k+1}(X) - r_{k+1}(X)||^2 \right)^2
\]
\[
\leq a(T) \text{ (by (ii))}
\]
Finally, pick any \( x_0 \in S_X \) and write down

\[
a(T) \geq E\{ (\gamma_{k+1}(X) - r_{k+1}(X))^2 \} = \sum_{x \in S_X} (\gamma_{k+1}(x) - r_{k+1}(x))^2 \pi(x) \\
\geq (\gamma_{k+1}(x_0) - r_{k+1}(x_0))^2 \pi(x_0)
\]

(11)

and we conclude that:

\[
|\gamma_{k+1}(x_0) - r_{k+1}(x_0)| \leq \left( \frac{a(T)}{\pi(x_0)} \right)^{\frac{1}{2}}. \Box
\]

We will also make use in the sequel the following lemma, that is an easy consequence of the Law of Large Numbers for Martingales (see, for instance, [11]).

**Lemma 3.2** Assume that \((\mathcal{F}_i)_{i \in \mathbb{N}}\) is a filtration (i.e., each \( \mathcal{F}_i \) is a sub-\( \sigma \)-algebra of the underlying \( \sigma \)-algebra \( \mathcal{A} \) and, for any \( i, \mathcal{F}_i \subset \mathcal{F}_{i+1} \)) and that \( \Delta_0, ..., \Delta_n, ... \) is a sequence of random variables such that:

1. \( \Delta_i \) is \( \mathcal{F}_{i+1} \)-measurable for any \( i \).
2. \( \mathbb{E}\{\Delta_i/\mathcal{F}_i\} = 0 \) for any \( i \).
3. There exists \( K < \infty \) such that \( \sup_i |\Delta_i| \leq K \), a.s.

Then, if \( M_n = \sum_{i=0}^{n-1} \Delta_i \), we have that

\[
\lim_{n} \frac{M_n}{n} = 0 \text{ a.s.}
\]

**Remark 3.3:** The following fact will also play a key role in the proof of our main results. For any \( x \in S_X \), define

\[
\lambda_h(x) = r_h(x)1_{\{h \leq k\}} + \gamma_{k+1}(x)1_{\{h = k+1\}}.
\]

Set

\[
\Lambda_h = \{ x \in S_X : \lambda_h(x) > 0 \}.
\]
It is clear that $\Lambda_h = D_h$ for $h \leq k$. On the other hand, by Lemma 3.1 (iii), for $T$ big enough, $\Lambda_{k+1} = D_{k+1}$. More precisely, define

$$\eta = \min\{|r_{k+1}(x)| : x \in S_X\}, \quad T_0 = \inf\{T \in \mathbb{N} : \left(\frac{a(T)}{\pi(x)}\right)^\frac{1}{2} < \eta \ \forall x \in S_X\}.$$ 

Take $T \geq T_0$. If $x \in D_{k+1}$, then $r_{k+1}(x) > \eta$ and by Lemma 3.1 (iii) and the definition of $T_0$, $\gamma_{k+1}(x) > 0$ and $x \in \Lambda_{k+1}$. If $x \notin D_{k+1}$, then $r_{k+1}(x) < -\eta$ and the same argument shows that $\gamma_{k+1}(x) < 0$, what implies that $x \notin \Lambda_{k+1}$. Therefore, if $T \geq T_0$,

$$\Lambda_h = D_h, \ h = 1, ..., k + 1.$$ 

We have then the first result, concerning the asymptotic behaviour of the credit matrix.

**Theorem 3.1** Let $T$ be a fixed value, $T \geq T_0$, with $T_0$ as in Remark 3.3. As $n$, tends to infinity we have that

$$\lim_{n} \frac{1}{n} c_n(x, h) = (\lambda_h(x)\pi(x)1_{\{h=h(x)\}})_{x \in S_X, h \in H}, \ \text{a.s.}$$

and

$$\lim_{n} h_n(x) = h(x) \ \text{a.s.}$$

(what implies that $h_n(x) = h(x)$ for all $n$ large enough, a.s.)

**Proof:** Define

$$U_h^j = \{x \in S_X : h_j(x) = h\},$$

(set of points where the best advisor at cycle $j$ is $h$)

$$T_j = \sigma \left(\{X_i^j, Y_i^j : 1 \leq i \leq T\}\right)$$

($\sigma$-algebra generated by the training sequence of cycle $j$),

$$V_j = \sigma \left(\{X_i^{v,j}, Y_i^{v,j} : 1 \leq i \leq V\}\right)$$

($\sigma$-algebra generated by the validation sequence of cycle $j$), and

$$F_j = \bigvee_{i=1}^{j-1} (T_i \vee V_i)$$

20
\((\sigma\text{-algebra generated by training and validation sequences up to cycle } j - 1)\).

Set
\[
\Delta_j = c_{j+1}(x, h) - c_j(x, h) - E\{c_{j+1}(x, h) - c_j(x, h) / \mathcal{F}_j\},
\]
that clearly satisfies all the hypotheses of Lemma 3.2.

We have that:
\[
c_{j+1}(x, h) - c_j(x, h) = \frac{1}{V} \sum_{i=1}^{V} R(X_i^{v_{j}}, f_j(X_i^{v_{j}}, Y_i^{v_{j}}, h)1_{\{X_i^{v_{j}} = x, h(X_i^{v_{j}}) = h\}},
\]
\[
= \frac{1}{V} \sum_{i=1}^{V} R(x, f_j(x), Y_i^{v_{j}}, h)1_{\{X_i^{v_{j}} = x, x \in U_k^j\}}.
\]

We will use in the following lines that the validation sequence of cycle \(j\) is independent with respect to the training sample of cycle \(j\) and with respect to training and validation samples of previous cycles, and that \(U_k^j\) is \(\mathcal{F}_j\)-measurable. If \(h \leq k\) and \(x \in U_k^j\), \(f_j(x) = y_h(x)\) (deterministic) and
\[
E\{c_{j+1}(x, h) - c_j(x, h) / \mathcal{F}_j\} = E\{R(x, \hat{f}_j(x), Y_i^{v_{j}}, h)1_{\{X_i^{v_{j}} = x\}}1_{\{x \in U_k^j\}}\}
\]
\[
= E\{R(x, y_h(x), Y_i^{v_{j}}, h)1_{\{X_i^{v_{j}} = x\}}1_{\{x \in U_k^j\}}\}
\]
\[
= r_h(x)\pi(x)1_{\{x \in U_k^j\}}.
\]

Therefore,
\[
\Delta_j = c_{j+1}(x, h) - c_j(x, h) - r_h(x)\pi(x)1_{\{x \in U_k^j\}} \text{ for } h \leq k.
\]

For \(h = k + 1\), let us compute more carefully:
\[
E\{c_{j+1}(x, k+1) - c_j(x, k+1) / \mathcal{F}_j\} = E\{R(x, \hat{f}_j(x), Y_i^{v_{j}}, k+1)1_{\{X_i^{v_{j}} = x\}} / \mathcal{F}_j\}1_{\{x \in U_k^{j+1}\}}.
\]

But
\[
E\{R(x, \hat{f}_j(x), Y_i^{v_{j}}, k+1)1_{\{X_i^{v_{j}} = x\}} / \mathcal{F}_j\} = E\{E\{R(x, \hat{f}_j(x), Y_i^{v_{j}}, k+1)1_{\{X_i^{v_{j}} = x\}} / \mathcal{F}_j \vee \mathcal{T}_j\} / \mathcal{F}_j\}.
\]

Since \(\hat{f}_j\) is \(\mathcal{F}_j \vee \mathcal{T}_j\)-measurable and \((X_i^{v_{j}}, Y_i^{v_{j}})\) is independent of \(\mathcal{F}_j \vee \mathcal{T}_j\), we have that:
\[
E\{R(x, \hat{f}_j(x), Y_i^{v_{j}}, k+1)1_{\{X_i^{v_{j}} = x\}} / \mathcal{F}_j \vee \mathcal{T}_j\} = \sum_{y \in S_Y} R(x, \hat{f}_j(x), y, k+1)p(y/x)\pi(x),
\]

21
what implies in turn that:

\[
E\{R(x, \hat{f}_j(x), Y_i^{x,j}, k+1)1_{\{X_i^{x,j}=x\}}/\mathcal{F}_j\} = \sum_{y \in S_Y} E\{R(x, \hat{f}_j(x), y, k+1)/\mathcal{F}_j\}p(y/x)\pi(x).
\]

Observe now that \( \hat{f}_j \) only depends on \( T_j \) and is independent of \( \mathcal{F}_j \) (by its definition, \( \hat{f}_j \) only depends on the performance of the elements of the model class \( \mathcal{F} \) over the whole training sequence of cycle \( j \)), and thus,

\[
\sum_{y \in S_Y} E\{R(x, \hat{f}_j(x), y, k+1)/\mathcal{F}_j\}p(y/x)\pi(x) = \sum_{y \in S_Y} \sum_{f \in \mathcal{F}} R(x, f(x), y, k+1)P(\hat{f}_j = f)p(y/x)\pi(x).
\]

Using now as in Lemma 3.1 the fact that the law of \( \hat{f}_j \) does not depend on \( j \), we conclude that

\[
E\{R(x, \hat{f}_j(x), Y_i^{x,j}, k+1)1_{\{X_i^{x,j}=x\}}/\mathcal{F}_j\} = \sum_{y \in S_Y} \sum_{f \in \mathcal{F}} R(x, f(x), y, k+1)P(\hat{f}_1 = f)p(y/x)\pi(x)
\]

\[
= \gamma_{k+1}(x).
\]

Therefore, for \( h = k+1 \),

\[
E\{c_{j+1}(x, k+1) - c_j(x, k+1)/\mathcal{F}_j\} = \gamma_{k+1}(x)1_{\{x \in U_{k+1}^d\}},
\]

what shows that for any \( h = 1, ..., k, k+1 \) we have

\[
E\{c_{j+1}(x, h) - c_j(x, h)/\mathcal{F}_j\} = \lambda_h(x)1_{\{x \in U_{h}^d\}},
\]

and that

\[
\Delta_j = c_{j+1}(x, h) - c_j(x, h) - \lambda_h(x)\pi(x)1_{\{x \in U_{h}^d\}} \text{ for } h \leq k + 1.
\]

Summing up both terms of this last equation with respect to \( j \) and dividing by \( n \), we obtain as a consequence of Lemma 3.2 that:

\[
\lim_{n} \left( \frac{c_n(x, h)}{n} - \lambda_h(x)\pi(x)\nu_n(h) \right) = 0, \ a.s.
\]
where
\[ \nu_n(h) = \frac{1}{n} \text{Card}\{ j : 0 \leq j \leq n - 1, x \in U_j \}. \]

From now on, the rest of the proof is devoted to show the following two facts:

- a) \( \lim_n \nu_n(h) = 1_{\{h=h(x)\}} \), for any \( x, h \), and
- b) \( \lim_n h_n(x) = h(x) \).

(Observe that b) it is not a direct consequence of a), since \( \mu_n(h(x)) \) gives only the asymptotic frequency of \( h_n(x) = h(x) \).

To prove this, fix \( x \in S_X \). Let \( \varepsilon \) be an arbitrary element of \((0,1)\). Set
\[ a = \inf_{h \in H} |\lambda_h(x)| \pi(x). \]
We already know that for almost any \( \omega \) in our probability space, there exists \( n_\varepsilon(\omega) \) such that
\[
\max_{h \in H} \left| \frac{c_n(x,h)}{n}(\omega) - \lambda_h(x)\pi(x)\nu_n(h) \right| < \frac{1}{2} a \varepsilon
\]  
for any \( n \geq n_\varepsilon(\omega) \).

Fix \( \omega \) as before. Let us assume for a moment that:
\[
\text{There exists } n_1 \geq n_\varepsilon(\omega) \text{ such that } \nu_{n_1}(h(x))(\omega) \geq \varepsilon. \tag{14}
\]

By Remark 3.4, we know that \( \Lambda_h = D_h \) for any \( h \). Hence, \( \lambda_h(x) > 0 \) if and only if \( h = h(x) \). We have then that:
\[
\frac{c_{n_1}(x,h(x))}{n_1}(\omega) > \lambda_{h(x)}(x)\pi(x)\nu_{n_1}(h(x))(\omega) - \frac{ae}{2} \\
\geq \frac{ae}{2} \geq \lambda_{h(x)}(x)\pi(x)\nu_{n_1}(h)(\omega) + \frac{ae}{2} \\
> \frac{c_{n_1}(x,h)}{n_1}(\omega) \text{ for any } h \neq h(x)
\]

This implies that \( h_{n_1}(x)(\omega) = h(x) \) and hence,
\[
\nu_{n_1+1}(h(x))(\omega) = \frac{n_1 \nu_n(h(x))(\omega) + 1}{n_1 + 1} > \varepsilon.
\]
Therefore, the same argument may be applied to \( n_1 + 1 \) instead of \( n_1 \), and we conclude that
\[
h_n(x) = h(x) \text{ for any } n \geq n_1
\]
what clearly implies
\[
\lim_{n} \nu_n(h) = 1_{\{h(x)\}}(\omega), \text{ for any } h.
\]

It is enough now to show that, on a set of probability one, there exists \( \varepsilon \in (0, 1) \) such that (14) holds true.

Let us call \( A \) to subset of \( \Omega \) where (14) does not hold for any \( \varepsilon \in (0, 1) \). It is clear that
\[
A = \{ \omega \in \Omega : \lim_{n} \nu_n(h(x))(\omega) = 0 \}.
\]

We will prove that \( P(A) = 0 \). Observe that the reward function \( R \) is bounded (indeed, its domain is a finite set) and hence \( \frac{c_n(x, h)}{n} \) is bounded, allowing to interchange limits and expectations in the following lines.

By the definition of \( h_n(x) \) we have that, for any \( \omega \) in \( \Omega \) and \( h \) in \( H \),
\[
\frac{c_n(x, h_n(x))}{n}(\omega) \geq \frac{c_n(x, h)}{n}(\omega)
\]
what implies that for any \( h \),
\[
E\{1_A \frac{c_n(x, h_n(x))}{n}\} \geq E\{1_A \frac{c_n(x, h)}{n}\} \tag{15}
\]
Using that
\[
\lim \max_{h \in H} \left| \frac{c_n(x, h)}{n} - \lambda_h(x) \pi(x) \nu_n(h) \right| = 0 \text{ a.s.}
\]
and taking limits in (15) we deduce that, for any \( h \),
\[
\limsup_n E\{1_A \lambda h_n(x) \pi(x) \nu_n(h_n(x))\} \geq \limsup_n E\{1_A \lambda h(x) \pi(x) \nu_n(h)\}.
\]
Since the right-hand side of the last inequality is non-negative for \( h = h(x) \), so is the left-hand side, hence
\[
\limsup_n E\{1_A \lambda h_n(x) \pi(x) \nu_n(h_n(x))\} \geq 0.
\]
But if we take
\[
a(x) = \max_{h \neq h(x)} r_h(x),
\]

24
which is negative, it is easy to check that the left-hand side of the last inequality is smaller than
\[
\limsup_n E(1_{A} a(x)\pi(x)1_{\{h_n(x)\neq h(x)\}})
\]  
and therefore (16) must be non-negative.

We will show that (16) is negative if \(P(A) > 0\), leading to a contradiction. Taking into account that \(a(x)\) is negative, using Fatou’s lemma for negative functions and the fact that
\[
\limsup_n 1_{h_n(x)\neq h(x)} = 1 \text{ over } A
\]
we conclude that (16) is smaller than
\[
E(1_{A} a(x)\pi(x)),
\]
which is negative if \(P(A) > 0\).

**Remark 3.4:** It must be noticed that in the previous result we have assumed that \(T\) is big enough (i.e., \(T \geq T_0\)), but fixed.

Next result shows that a CLT holds for the convergence of Theorem 3.1. It is based on the following version of the CLT for martingales (see [11]).

**Lemma 3.3** *Under the assumptions of Lemma 3.2, if in addition there exists a non-negative constant \(\sigma^2\) such that
\[
\lim_n \frac{1}{n} \sum_{i=0}^{n-1} E\{\Delta_i^2/\mathcal{F}_i\} = \sigma^2 \text{ in probability,}
\]
then
\[
\lim_n \frac{1}{\sqrt{n}} M_n = N(0, \sigma^2) \text{ in law.}
\]

**Remark 3.5:** A straightforward argument gives the multivariate version of Lemma 3.3., that may be stated as follows. Assume that \((\Delta_1, \ldots, \Delta_i(d))_{i \in \mathbb{N}}\) is a \(d\)-dimensional sequence such that \((\Delta_i(s))_{i \in \mathbb{N}}\) satisfies the assumptions of Lemma 3.2 for each \(s = 1, \ldots, d\) with respect to the same filtration \((\mathcal{F}_i)_{i \in \mathbb{N}}\) and that there exists a covariance matrix \(M\) such that
\[
\lim_n \frac{1}{n} \sum_{i=0}^{n-1} E\{\Delta_i(s)\Delta_i(t)/\mathcal{F}_i\} = M(s, t) \text{ in probability, for any } s, t = 1, \ldots, d.
\]
Then if $M_n = (M_n(1), \ldots, M_n(d))$ is defined by $M_n(s) = \frac{1}{n} \sum_{i=0}^{n-1} \Delta_i(s)$, we have that
\[
\lim_{n} \frac{1}{\sqrt{n}} M_n = N(0, M) \text{ in law},
\]
where $N(0, M)$ denotes a $d$-dimensional centered gaussian random vector with covariance matrix $M$.

**Theorem 3.2** We have that
\[
\lim_{n} \left( \frac{1}{n} c_n(x, h) - \lambda_h(x) \pi(x) 1_{\{h=h(x)\}} \right)_{x \in S_X, h \in H} = N(0, M) \text{ in law},
\]
where, for any $x, x^* \in S_X$, $h, h^* \in H$,
\[
M(x, h; x^*, h^*) = \left( \frac{V - 1}{V} \right) \lambda_h(x) \lambda_{h^*}(x^*) \pi(x) \pi(x^*) 1_{\{h(x)=h, h(x^*)=h^*\}},
\]
and
\[
\theta_h(x) := E\{R(x, y_h(x), Y, h)^2\} \text{ for } h = 1, \ldots, k; \, \theta_{k+1}(x) := E\{R(x, \hat{f}^1(x), Y, h)^2\}.
\]

**Proof:** Set
\[
\Delta_j(x, h) = c_{j+1}(x, h) - c_j(x, h) - \lambda_h(x) \pi(x) 1_{\{x \in U^j_{h}\}} \text{ for } x \in S_X, h \in H.
\]
After Remark 3.5, it is clear that is suffices to prove the following facts:

- **a)** $\lim_{n} \frac{1}{n} \sum_{j=1}^{n} E\{\Delta_j(x, h)\Delta_j(x^*, h^*)/\mathcal{F}_j\} = M(x, h; x^*, h^*)$ in probability, for any $x, x^* \in S_X$, $h, h^* \in H$.

- **b)** $\lim_{n} \sqrt{n} \left( \frac{1}{n} \sum_{j=0}^{n-1} 1_{\{x \in U^j_{h}\}} - 1_{\{h=h(x)\}} \right) = 0$ a.s. for any $x \in S_X, h \in H$.

By Theorem 3.1., for each $x \in S_X$, and for any $\omega$ on a set of total probability, there exists $n(\omega) \in \mathbb{N}$ such that for any $n \geq n(\omega)$ we have $h_n(x)(\omega) = h(x)$. It is then clear that, for $n \geq n(\omega)$, we have that
\[
\left| \frac{1}{n} \sum_{j=0}^{n-1} 1_{\{x \in U^j_{h}\}}(\omega) - 1_{\{h=h(x)\}} \right| \leq \frac{n(\omega)}{n},
\]
what clearly implies b).

We will then focus on a). It is easy to check that
\[
E\{\Delta_j(x, h)\Delta_j(x^*, h_*)/\mathcal{F}_j\} = E\{(c_{j+1}(x, h) - c_j(x, h))(c_{j+1}(x^*, h_*) - c_j(x^*, h_*))\} \\
- \lambda_h(x)\lambda_{h^*}(x^*)\pi(x)\pi(x^*)P(h_j(x) = h, h_j(x^*) = h^*).
\]

But
\[
E\{(c_{j+1}(x, h) - c_j(x, h))(c_{j+1}(x^*, h_*) - c_j(x^*, h_*))\} = \\
\frac{1}{V^2} \sum_{s=1}^V \sum_{t=1}^V E\{R(x, f_j(x), Y_s^{x,t}; h)R(x^*, f_j(x^*), Y_t^{x,t}; h^*) 1_{\{X_s^{x,t} = x, h_j(x) = h\}} 1_{\{X_t^{x,t} = x^*, h_j(x^*) = h^*\}}\}
\]

If \(s \neq t\), then
\[
E\{R(x, f_j(x), Y_s^{x,t}; h)R(x^*, f_j(x^*), Y_t^{x,t}; h^*) 1_{\{X_s^{x,t} = x, h_j(x) = h\}} 1_{\{X_t^{x,t} = x^*, h_j(x^*) = h^*\}}\} = \\
\lambda_h(x)\lambda_{h^*}(x^*)\pi(x)\pi(x^*)P(h_j(x) = h, h_j(x^*) = h^*).
\]

On the other hand, if \(s = t\), then
\[
E\{R(x, f_j(x), Y_s^{x,t}; h)R(x^*, f_j(x^*), Y_t^{x,t}; h^*) 1_{\{X_s^{x,t} = x, h_j(x) = h\}} 1_{\{X_t^{x,t} = x^*, h_j(x^*) = h^*\}}\} = \\
\theta_h(x)\pi(x)1_{\{x = x^*, h = h^*\}}P(h_j(x) = h).
\]

Therefore, we have that
\[
E\{(c_{j+1}(x, h) - c_j(x, h))(c_{j+1}(x^*, h_*) - c_j(x^*, h_*))\} = \\
\frac{(V - 1)}{V}\lambda_h(x)\lambda_{h^*}(x^*)\pi(x)\pi(x^*)P(h_j(x) = h, h_j(x^*) = h^*) \\
- \frac{1}{V}\theta_h(x)\pi(x)1_{\{x = x^*, h = h^*\}}P(h_j(x) = h),
\]

and applying Theorem 3.1, a) follows easily.\(\diamondsuit\)

**Remark 3.6:** Observe that for each pair \(x, x^*\), the limit covariance matrix is non-null except in the case \(h(x) = h, h(x^*) = h^*\). Indeed, instead of the whole credit matrix, we may consider the reduced mean credit vector \((\frac{1}{n}c_n(x, h(x)))_{x \in S_x}\) since no other term is relevant for the asymptotic behaviour of the algorithm.
Next, we give the asymptotic behaviour of the credit matrix when the NRM algorithm is used.

**Theorem 3.3** For the NRM algorithm, we have that

\[
\lim_{n} \frac{1}{n} \epsilon_n(x, h)_{x \in S_X, h \in H} = (\lambda_h(x)\pi(x)1_{\{h(x)\}})_{x \in S_X, h \in H}, \ a.s.
\]

**Proof:** For the sake of clarity, let us use a different notation for the principal ingredients of the algorithm in the NRM case. Let us now denote \( h_j(x) \) the analogous of \( h(x) \), \( g_j \) the analogous of \( f_j \) and \( g^j \) the analogous of \( f^j \). More precisely, we assume now that, at cycle \( j \), the available training sample is

\[
(X^*_i, Y^*_i)^{1 \leq i \leq T, 1 \leq s \leq j}.
\]

Hence, from the model we will choose \( \hat{g}^j \) such that

\[
\hat{g}^j = \arg\max_{f \in \mathcal{F}} \Gamma_{k+1;jT}(f)
\]

where

\[
\Gamma_{k+1;jT}(f) = \frac{1}{jT} \sum_{s=1}^{j} \sum_{i=1}^{T} R(X^*_i, f(X^*_i), Y^*_i, k + 1).
\]

Finally, \( \hat{h}_j(x) \) is now the most credible advisor among the \( k \) experts and \( \hat{g}^j \), the credit matrix is updated exactly as before (i.e., using only the validation sequence correspondingly to each cycle) and \( g_j \) denotes the predictor.

If we now set

\[
\hat{r}^j_{k+1}(x) = \sum_{y \in S_Y} R(x, \hat{g}^j(x), y, k + 1)p(y/x),
\]

it is clear that

\[
(\hat{r}^j_{k+1}(x))_{x \in S_X}
\]

has the same law as, in the MRLA,

\[
(r^j_{k+1}(x))_{x \in S_X}
\]

when a training sample of size \( jT \) is used, and therefore, by Lemma 3.1 ii), it converges in \( L^2 \), as \( j \) goes to infinity, to \( r_{k+1}(x) \).
From now on, the proof follows very closely the arguments used in Theorem 3.1 and may be easily reproduced by the reader.  

As a direct consequence of Theorem 3.1 and 3.2, we can finally compare the performance of MRLA and NRM. We will compare performances by means of the following performance ratio:

\[ \tau_j := \frac{E\{R(X, f_j(X), Y, h_j(X))\}}{E\{R(X, f_j(X), Y, h_j(X))\}}. \]

We have then

**Theorem 3.4** If in MRLA we use \( T \geq T_0 \), then

\[ \lim_j \tau_j = \frac{E\{\lambda_h(x)\}}{E\{\gamma_h(x)\}} = 1 - \frac{E\{(\gamma_{h+1}(x) - \gamma_{h+1}(X))1_{\{X \in D_{h+1}\}}\}}{E\{\gamma_h(x)\}}. \]

**Proof:** As seen before, we have that

\[
E\{R(X, f_j(X), Y, h_j(X))\} = \sum_{h=1}^{k} \sum_{x \in S_x} \sum_{y \in S_Y} R(x, y_h(x), y, h)P(h_j(x) = h)p(y/x)\pi(x) + \\
\sum_{x \in S_x} \sum_{f \in \text{natural}} \sum_{y \in S_Y} R(x, f(x), y, h)P(f^j = f)P(h_j(x) = k + 1)p(y/x)\pi(x) = \\
\sum_{h=1}^{k} \sum_{x \in S_x} \tau_h(x)\pi(x)P(h_j(x) = x) + \sum_{x \in S_x} \gamma_{h+1}(x)\pi(x)P(h_j(x) = k + 1) = \\
\sum_{h=1}^{k+1} \sum_{x \in S_x} \lambda_h(x)\pi(x)P(h_j(x) = h).
\]

By Theorem 3.1,

\[ \lim_j P\{h_j(x) = h\} = 1_{\{h = h(x)\}}, \]

and therefore,

\[ \lim_j E\{R(X, f_j(X), Y, h_j(X))\} = \sum_{h=1}^{k+1} \sum_{x \in S_x} \lambda_h(x)\pi(x)1_{\{h = h(x)\}} = \\
\]

29
\[
\sum_{x \in S_X} \lambda_h(x)\pi(x) = E\{\lambda_h(X)(X)\}.
\]

In a similar way, using Theorem 3.2, we deduce that
\[
\lim_j E\{R(X, f_j(X), Y, h_j(X))\} = E\{r_h(X)(X)\}.
\]

Finally, observe that
\[
E\{\lambda_h(X)(X)\} = E\{\lambda_h(X)(X)1_{X \notin D_{k+1}}\} + E\{\lambda_{k+1}(X)(X)1_{X \in D_{k+1}}\} =
E\{r_h(X)(X)1_{X \notin D_{k+1}}\} + E\{\gamma_{k+1}(X)(X)1_{X \in D_{k+1}}\},
\]
and the result follows. \(\Diamond\)

The following corollary illustrates on the applications of Theorem 3.4.

**Corollary 3.1**

i) Under the assumptions of Theorem 3.4, we have
\[
\lim_j \tau_j \geq 1 - \frac{(a(T)\pi(D_{k+1}))^{\frac{1}{2}}}{E\{r_h(X)\}}.
\]

ii) Set \(A = \min_{x \in D_{k+1}} r_{k+1}(x), B = \min_{x \notin D_{k+1}} r_{h}(x)\).
Then
\[
\lim_j \tau_j \geq 1 - \frac{(a(T)\pi(D_{k+1}))^{\frac{1}{2}}}{A\pi(D_{k+1}) + B(1 - \pi(D_{k+1}))}.
\]

**Proof:**
To prove i), observe that \(E\{r_h(X)(X)\}\) is positive and apply Cauchy-Schwarz inequality and Lemma 3.1 iv) in Theorem 3.4.
To prove ii), observe that
\[
E\{r_h(X)(X)\} \geq A\pi(D_{k+1}) + B(1 - \pi(D_{k+1})) \Diamond
\]

**Remark 3.7:** Corollary 3.1 says that if \(T\) is large, both algorithms have very similar performances. On the other hand, if \(T \geq T_0\) but \(T\) is not very
large, the ratio of performances is close to one if \( \pi(D_{k+1}) \) is small or if \( A \) or \( B \) are large. In simple words, this means that performance is almost the same if training samples are large, or experts are the most credible advisor for almost all inputs, or experts have a very good mean performance (even if they are not almost always chosen as the best advisor) or the model has a very good mean performance.

**Remark 3.8**: If an adaptive procedure allows to choose the best predictor of the model, of course there is no substantial difference on memory requirements in the MRLA and the NRM. In this case, NRM will perform a step-by-step computation of \( g' \) until the whole training sample of size \( jT \) has been used and it will the proceed to validation and updating of credits. Therefore, there is not difference on memory capacity requirement, but MRLA will give a series of intermediate predictors and credit matrices before the end of NRM.

## 4 Simulations

### 4.1 Experiments

We have simulated a Learning system using the following configuration. \( X \) is an Uniform 10-dimensional discrete variable taking values among 1, 2 and 3. \( Y \) is a one-dimensional binary variable. The model is an optimal CART tree constructed as in [?]. \( K = 2 \) experts are included in the system. Sample size for each iteration is \( n = 200 \). We run \( J = 300 \) iterations for each experiment. Each experiment is repeated 3 times. The true model and the experts differ according to the experiments. The used Pseudo Code is described in table 1. We have trained in parallel the NMR algorithm, based on the cumulated data sets at each iteration. Its credit matrix is incremented in the same manner as the MRL model. We have tested four different experiments differing each from other in the choice of the true model \( f \) and the experts. experiment 1:

\[
\begin{align*}
f(X) &= 1 \text{ if } sd(X) > .8 \\
h_1(X) &= 1 \text{ if } X_2 = 2 \\
h_2(X) &= 1 \text{ if } X_5 = 2
\end{align*}
\]
$X = (X_1, X_2, ..., X_p)$ predictors.

$n$ Sample size, $J$ number of iterations.
$h_1, h_2, ..., h_{K+1}$ advisors, $h_{K+1} = f$ model to estimate. $h_k : X \to \{0, 1\}$

**Initialization**

$h_0 \in (1, ..., k + 1) \ N \times 1$ randomly selected.

$C = 0$ credit matrix, $N \times (k + 1)$ matrix

for ($j$ in 1 to $J$)

$X_j$ new sample of size $n$.

$y_j = f(X_j)$

$X_V$ validation sample, $n \times p$ matrix, $y_V = f(X_V), V = (y_V, X_V)$

$V_I$ = subset of $V$ for which the model $f_j$ has the maximum credit.

$UV = (y_{UV}, X_{UV}) = V$ without the duplicated observations : $N \times (p + 1)$

$w(x)$ frequency of each observation of UV in V

$f_{j+1}$ = best model trained over $(y_j, X_j)$ , tested over $V_I$

$f_{j+1} = f_{j+1}(UV)$

for $x$ in 1 to $N$

$C_{x, h_0} = C_{x, h_0} + w(x)(1_{f_{j+1}(x) = y_{UV}[x]} - 1_{f_{j+1} = y_{UV}[x]})/n$

$h_0 = col\cdot max(C)$ , best advisor for each observation.

**Table 1:** Pseudo Code for restricted memory learning algorithm.

$$sd(X)$$ is the standard deviation of $X$. experiment 2 :

$$f(X) = 1 \text{ if } \langle X \cdot 1 \rangle > 20$$

$$h_1(X) = 1 \text{ if } X_2 = 2$$

$$h_2(X) = 1 \text{ if } X_5 = 2$$

experiment 3 :

$$f(X) = 1 \text{ if } \langle X \cdot 1 \rangle > 20$$

$$h_1(X) = 1 \text{ if } \langle X \cdot 1 \rangle > 15$$

$$h_2(X) = 1 \text{ if } \langle X \cdot 1 \rangle < 25$$
Experiment 4:

\[ f(X) = 1 \text{ if } \langle X \cdot 1 \rangle > 20 \]
\[ h_1(X) = 1 \text{ if } X_2 = 2 \]
\[ h_2(X) = 1 \text{ if } \langle X \cdot 1 \rangle > 21 \]

Experiment 1 is a hard task for CART. The others are easier but need trees with quite big number of leaves. The true model is the same in the experiments 2 and 3. Experts are chosen to be performant in the third experiment but not in experiment 2. In the fourth experiment one of the experts, the second, is chosen to be very close to the true model.

4.2 Results

Figures 1 to 4 present the results for the four experiments. Each figure represents the results of 1000 iterations runs of our algorithm. The upper curve represents the trend of the reward ratios between the NMRL and MRL algorithms (MRL /NMRL). The lower ones shows the mean advisors’ credits evolution, the mean being taken over all the possible observations. The left pannel is for MRL, and the right one for NMRL. Each curve represents an advisor. The higher ones which are linear correspond to the model in all of the cases.

The reward ratio oscillates round 1 in all the experiments, staying often lower than one. In the four experiments and for the two algorithms NMR and NMRL, the credit of the model increases linearly. What we observe corresponds to the result given in theorem 3.4. NMRL credits are always higher then MRL ones. Experts credit varies differently in the four experiments but stays often near zero, very low compared to the model, except for the fourth experiment. In the fourth experiment, the second’s advisor’s credit approaches the model, the other decreases slightly staying close to zero. Figure 5 (resp 6) shows advisors’ credit evolution for 16 randomly chosen observations for the NMRL (resp. MRL) algorithm. The Credits are not monotone and are locally constant. For 6 observations the 3 advisor’s credit has exactly the same behaviour within the two algorithms, for the others it is different.
Figure 1: first experiment

Figure 2: second experiment
Figure 3: third experiment

Figure 4: fourth experiment
NMRL: advisors' credit evolution for 16 randomly chosen observations

Figure 5: credit evolution for some observations in NMRL
MRL: advisors’ credit evolution for 16 randomly chosen observations

Figure 6: credit evolution for some observations in MRL
References


