Online Learning in Adversarial Lipschitz Environments

Odalric Maillard, Rémi Munos

Online Learning in Adversarial Lipschitz Environments

Odalric-Ambrym Maillard and Rémi Munos

SequeL Project, INRIA Lille - Nord Europe, France
{odalric.maillard, remi.munos}@inria.fr

Abstract. We consider the problem of online learning in an adversarial environment when the reward functions chosen by the adversary are assumed to be Lipschitz. This setting extends previous works on linear and convex online learning. We provide a class of algorithms with cumulative regret upper bounded by $\tilde{O}(\sqrt{dT\ln(\lambda)})$ where d is the dimension of the search space, T the time horizon, and λ the Lipschitz constant. Efficient numerical implementations using particle methods are discussed. Applications include online supervised learning problems for both full and partial (bandit) information settings, for a large class of non-linear regressors/classifiers, such as neural networks.

Introduction

The adversarial online learning problem is defined as a repeated game between an agent (the learner) and an opponent, where at each round t, simultaneously the agent chooses an action (or decision, or arm, or state) $\theta_t \in \Theta$ (where Θ is a subset of $\mathbb{R}^d$) and the opponent chooses a reward function $f_t : \Theta \mapsto [0,1]$. The agent receives the reward $f_t(\theta_t)$. In this paper we will consider different assumptions about the amount of information received by the agent at each round. In the *full information* case, the full reward function f_t is revealed to the agent after each round, whereas in the case of *bandit information* only the reward corresponding to its own choice $f_t(\theta_t)$ is provided.

The goal of the agent is to allocate its actions $(\theta_t)_{1\le t\le T}$ in order to maximize the sum of obtained rewards $F_T \stackrel{\text{def}}{=} \sum_{t=1}^T f_t(\theta_t)$ up to time T and its performance is assessed in terms of the best constant strategy $\theta \in \Theta$ on the same reward functions, i.e. $F_T(\theta) \stackrel{\text{def}}{=} \sum_{t=1}^T f_t(\theta)$. Defining the **cumulative regret**:

$$R_T(\theta) \stackrel{\text{def}}{=} F_T(\theta) - F_T,$$

with respect to (w.r.t.) a strategy θ, the agent aims at minimizing $R_T(\theta)$ for all $\theta \in \Theta$.

In this paper we consider the case when the functions f_t are Lipschitz w.r.t. the decision variable θ (with Lipschitz constant upper bounded by λ).

Previous results. Several works on adversarial online learning include the case of finite action spaces (the so-called *learning from experts* [1] and the *multi-armed bandit problem* [2, 3]), countably infinite action spaces [4], and the case of continuous action spaces, where many works have considered strong assumptions on the reward functions, i.e. linearity or convexity.

In the *online linear optimization* (see e.g. [5–7] in the adversarial case and [8, 9] in the stochastic case) where the functions f_t are linear, the resulting upper- and lower-bounds on the regret are of order (up to logarithmic factors) $\sqrt{dT}$ in the case of full information and $d^{3/2}\sqrt{T}$ in the case of bandit information [6] (and in good cases $d\sqrt{T}$ [5]). In *online convex optimization* f_t is assumed to be convex [10] or σ-strongly convex [11], and the resulting upper bounds are of order $C\sqrt{T}$ and $C^2\sigma^{-1}\ln(T)$ (where C is a bound on the gradient of the functions, which implicitly depends on the space dimension). Other extensions have been considered in [12, 13] and a minimax lower bound analysis in the full information case in [14]. These results hold in bandit information settings where either the value or the gradient of the function is revealed.

To our knowledge, the weaker Lipschitz assumption that we consider here has not been studied in the adversarial optimization literature. However, in the stochastic bandit setting (where noisy evaluations of a fixed function are revealed), the Lipschitz assumption has been previously considered in [15, 16], see the discussion in Section 2.3.

Motivations: In many applications (such as the problem of matching ads to web-page contents on the Internet) it is important to be able to consider both large action spaces and general reward functions. The continuous space problem appears naturally in online learning, where a decision point is a classifier in a parametric space of dimension d. Since many non-linear non-convex classifiers/regressors have shown success (such as neural-networks, support vector machines, matching pursuits), we wish to extend the results of online learning to those non-linear non-convex cases. In this paper we consider a Lipschitz assumption (illustrated in the case of neural network architectures) which is much weaker than linearity or convexity.

What we do: We start in Section 1 by describing a general continuous version of the Exponentially Weighted Forecaster and state (Theorem 1) an upper bound on the cumulative regret of $O(\sqrt{dT\ln(d\lambda T)})$ under a non-trivial geometrical property of the action space. The algorithm requires, as a sub-routine, being able to sample actions according to continuous distributions, which may be impossible to do perfectly well in general.

To address the issue of sampling, we may use different sampling techniques, such as uniform grids, random or quasi-random grids, or use adaptive methods such as Monte-Carlo Markov chains (MCMC) or Population Monte-Carlo (PMC).

However, since any sampling technique introduces a sampling bias (compared to an ideal sampling from the continuous distribution), this also impacts the resulting performance of the method in terms of regret. This shows a tradeoff

between regret and numerical complexity, which is illustrated by numerical experiments in Section 1.3 where PMC techniques are compared to sampling from uniform grids.

Then in Section 2 we describe several applications to learning problems. In the full information setting (when the desired outputs are revealed after each round), the case of regression is described in Section 2.1 and the case of classification in Section 2.2. Then Section 2.3 considers a classification problem in a bandit setting (i.e. when only the information of whether the prediction is correct or not is revealed). In the later case, we show that the expected number of mistakes does not exceed that of the best classifier by more than $O(\sqrt{dTK\ln(d\lambda T)})$, where K is the number of labels. We detail a possible PMC implementation in this case.

We believe that the work reported in this paper provides arguments that the use of MCMC, PMC, and other adaptive sampling techniques is a promising direction for designing numerically efficient algorithms for online learning in adversarial Lipschitz environments.

1 Adversarial learning with full information

We consider a search space $\Theta \subset \mathbb{R}^d$ equipped with the Lebesgue measure μ. We write $\mu(\Theta) = \int_\Theta 1$. We assume that all reward functions f_t have values in $[0,1]$ and are Lipschitz w.r.t. some norm $||\cdot||$ (e.g. L_1, L_2, or L_∞) with a Lipschitz constant upper bounded by $\lambda > 0$, i.e. for all $t \geq 1$ and $\theta_1, \theta_2 \in \Theta$,

$$|f_t(\theta_1) - f_t(\theta_2)| \leq \lambda ||\theta_1 - \theta_2||.$$

1.1 The ALF algorithm

We consider the natural extension of the EWF (Exponentially Weighted Forecaster) algorithm [17, 18, 1] to the continuous action setting. Since this in an Adversarial Lipschitz Full-information environment, we call it ALF algorithm. The algorithm is described in Figure 1.

At each time step, the forecaster samples θ_t from a probability distribution $p_t \stackrel{\text{def}}{=} \frac{w_t}{\int_\Theta w_t}$ with w_t being the weight function defined according to the previously observed reward functions $(f_s)_{s<t}$. The function f_t is then revealed and the weight function is updated. We have $w_{t+1}(\theta) = \exp(\eta F_t(\theta))$, and η is a parameter of the algorithm.

Geometric considerations: The performance of the algorithm depends on the geometry of the space $\Theta \subset \mathbb{R}^d$ (relatively to the chosen norm), and since we want to derive bounds as a function of the dimension d, we now define classes of domains ($(\Theta_d)_{d\geq 0}$ indexed by their dimension) with similar geometrical properties.

Definition 1. *For the class of domains $(\Theta_d)_{d\geq 1}$, we define $\kappa(d) > 1$:*

$$\kappa(d) \stackrel{\text{def}}{=} \sup_{\theta\in\Theta_d, r>0} \frac{\min\left[\mu\big(B(\theta,r)\big), \mu(\Theta_d)\right]}{\mu\big(B(\theta,r)\cap\Theta_d\big)} \tag{1}$$

Initialization: Set $w_1(\theta) = 1$ for all $\theta \in \Theta$.

For each round $t = 1, 2, \ldots, T$

(1) Simultaneously the adversary chooses the reward function $f_t : \Theta \mapsto [0, 1]$, and the learner chooses $\theta_t \overset{iid}{\sim} p_t$, where $p_t(\theta) \overset{\text{def}}{=} \dfrac{w_t(\theta)}{\int_\Theta w_t(\theta) d\theta}$,

(2) The learner incurs the reward $f_t(\theta_t)$,

(3) The reward function f_t is revealed to the learner. The weight function w_t is updated as:

$$w_{t+1}(\theta) \overset{\text{def}}{=} w_t(\theta) e^{\eta f_t(\theta)}, \text{ for all } \theta \in \Theta$$

Fig. 1. Adversarial Lipschitz learning algorithm in a Full-information setting (ALF algorithm)

Assumption A1 There exists $\kappa > 0$ such that $\kappa(d) \leq \kappa^d$, for all $d \geq 1$, and there exists $\kappa' > 0$ and $\alpha \geq 0$ such that $\mu(B(\theta, r)\big) \geq (r/(\kappa' d^\alpha))^d$ for all $r > 0$, $d \geq 1$, and $\theta \in \mathbb{R}^d$.

The first part of this assumption says that $\kappa(d)$ scales at most exponentially with the dimension. This is reasonable if we consider domains with similar geometries (i.e. whenever the "angles" of the domains do not go to zero when the dimension d increases). The second part of the assumption about the volume of d-balls is a property of the norms and holds naturally for any usual norm: for example, $\kappa' = 1/2$, $\alpha = 0$ for L_∞, and $\kappa' = \sqrt{\pi}/(\sqrt{2}e)$, $\alpha = 3/2$ for any norm L_p, $p \geq 1$, since for L_p norms, $\mu(B(\theta, r)) \geq (2r)^d/d!$ and from Stirling formula, $d! \sim \sqrt{2\pi d}(d/e)^d$, thus $\mu(B(\theta, r)) \geq \big(r/(\frac{\sqrt{2\pi}}{2e} d^{3/2})\big)^d$.

Remark 1. Notice that Assumption A1 makes explicit the required geometry of the domain in order to derive tight regret bounds.

We now provide upper-bounds for the ALF algorithm on the worst expected regret (i.e. $\sup_{\theta \in \Theta} \mathbb{E} R_T(\theta)$) and high probability bounds on the worst regret $\sup_{\theta \in \Theta} R_T(\theta)$.

Theorem 1. *(ALF algorithm) Under Assumption A1, for any $\eta \leq 1$, the expected (w.r.t. the internal randomization of the algorithm) cumulative regret of the ALF algorithm is bounded as:*

$$\sup_{\theta \in \Theta} \mathbb{E} R_T(\theta) \leq T\eta + \frac{1}{\eta}\Big[d \ln(c d^\alpha \eta \lambda T) + \ln(\mu(\Theta))\Big], \tag{2}$$

whenever $(d^\alpha \eta \lambda T)^d \mu(\Theta) \geq 1$, where $c \overset{\text{def}}{=} 2\kappa \max(\kappa', 1)$ is a constant (which depends on the geometry of Θ and the considered norm). Under the same assumptions, with probability $1 - \beta$,

$$\sup_{\theta \in \Theta} R_T(\theta) \leq T\eta + \frac{1}{\eta}\Big[d \ln(c d^\alpha \eta \lambda T) + \ln(\mu(\Theta))\Big] + \sqrt{2T \ln(\beta^{-1})}. \tag{3}$$

We deduce that for the choice $\eta = \left(\frac{d}{T}\ln(cd^\alpha \lambda T)\right)^{1/2}$*, when* $\eta \leq 1$ *and assuming* $\mu(\Theta) = 1$*, we have:*

$$\sup_{\theta\in\Theta} \mathbb{E}R_T(\theta) \leq 2\sqrt{dT\ln(cd^\alpha\lambda T)},$$

and a similar bound holds in high probability.

The proof is given in Appendix A. Note that the parameter η of the algorithm depends very mildly on the (unknown) Lipschitz constant λ. Actually even if λ was totally unknown, the choice $\eta = \left(\frac{d}{T}\ln(cd^\alpha T)\right)^{1/2}$ would yield a bound $\sup_{\theta\in\Theta}\mathbb{E}R_T(\theta) = O(\sqrt{dT\ln(dT)}\ln\lambda)$ which is still logarithmic in λ (instead of linear in the case of the discretization) and enables to consider classes of functions for which λ may be large (and unknown).

Anytime algorithm. Like in the discrete version of EWF (see e.g. [19, 20, 1]) this algorithms may easily be extended to an anytime algorithm (i.e. providing similar performance even when the time horizon T is not known in advance) by considering a decreasing coefficient $\eta_t = \left(\frac{d}{2t}\ln(cd^\alpha\lambda t)\right)^{1/2}$ in the definition of the weight function w_t. We refer to [20] for a description of the methodology.

The issue of sampling. In order to implement the ALF algorithm detailed in Figure 1 one should be able to sample θ_t from the continuous distribution p_t. However it is in general impossible to sample perfectly from arbitrary continuous distributions p_t, thus we need to resort to approximate sampling techniques, such as based on uniform grids, random or quasi-random grids, or adaptive methods such as Monte-Carlo Markov Chain (MCMC) methods or population Monte-Carlo (PMC) methods. If we write p_t^N the distribution from which the samples are actually generated, where N stands for the computational resources (e.g. the numberof grid points if we use a grid) used to generate the samples, then the expected regret $\mathbb{E}R_T(\theta)$ will suffer an additional term of at most $\sum_{t=1}^T |\int_\Theta p_t f_t - \int_\Theta p_t^N f_t|$. This shows a tradeoff between the regret (low when N is large, i.e. p_t^N is close to p_t) and numerical complexity and memory requirement (which scales with N). In the next two sub-sections we discuss sampling techniques based on fixed grids and adaptive PMC methods, respectively.

1.2 Uniform grid over the unit hypercube

The simplest sampling technique consists in setting a uniform grid (say with N grid points) before the learning starts and sampling at each round one point of the grid. In that case the distribution has finite support and the sampling is easy. It is easy to see that this method will provide the same order of regret as the continuous version, when N is large enough.

Actually, in the case when the domain is the unit hypercube, it is easy to do the analysis of an Exponentially Weighted Forecaster (EWF) playing on the grid. Indeed, let $\Theta_N \stackrel{\text{def}}{=} \{\theta_1, \ldots, \theta_N\}$ be a uniform grid of resolution $h > 0$, i.e. such

that for any $\theta \in \Theta$, $\min_{1 \leq i \leq N} ||\theta - \theta_i|| \leq h$. This means that at each round t, we select the action $\theta_{I_t} \in \Theta_N$, where $I_t \overset{iid}{\sim} p_t^N$ with p_t^N the distribution on $\{1, \ldots, N\}$ defined by $p_t^N(i) \overset{\text{def}}{=} w_t(i) / \sum_{j=1}^N w_t(j)$, where the weights are defined as $w_t(i) \overset{\text{def}}{=} e^{\eta F_{t-1}(\theta_i)}$ for some appropriate constant $\eta = \sqrt{2 \ln N / T}$.

The usual analysis of EWF implies that the regret relatively to any point of the grid is upper bounded as: $\sup_{1 \leq i \leq N} \mathbb{E} R_T(\theta_i) \leq \sqrt{2T \ln N}$.

Now, since we consider the unit hypercube $\Theta = [0,1]^d$, and under the assumption that the functions f_t are λ-Lipschitz with respect to L_∞-norm, we have that $F_T(\theta) \leq \min_{1 \leq i \leq N} F_T(\theta_i) + \lambda T h$. We deduce that the expected regret relatively to any $\theta \in \Theta$ is bounded as $\sup_{\theta \in \Theta} \mathbb{E} R_T(\theta) \leq \sqrt{2T \ln N} + \lambda T h$.

Setting $N = h^{-d}$ with the optimal choice of h in the previous bound (up to a logarithmic term) $h = \frac{1}{\lambda} \sqrt{d/T}$ gives the upper bound on the regret: $\sup_{\theta \in \Theta} \mathbb{E} R_T = O(\sqrt{dT \ln(\lambda \sqrt{T})})$.

However this discretized EWF algorithm suffers from severe limitations from a practical point of view:

1. The choice of the best resolution h of the grid depends crucially on the knowledge of the Lipschitz constant λ and has an important impact on the regret bound. However, usually λ is not known exaclty (but an upper-bound may be available, e.g. in the case of neural networks discussed below). If we choose h irrespective of λ (e.g. $h = \sqrt{d/T}$) then the resulting bound on the regret will be of order $O(\lambda \sqrt{dT})$ which is much worst in terms of λ than its optimal order $\sqrt{\ln \lambda}$.
2. The number of grid points (which determines the memory requirement and the numerical complexity of the EWF algorithm) scales exponentially with the dimension d.

Notice that instead of using a uniform grid, one may resort to the use of random (or quasi-random) grids with a given number of points N, which would scale better in high dimensions. However all those method are non-adaptive in the sense that the position of the grid point do not adapt to the actual reward functions f_t observed through time. We would like to sample points according to an "adaptive discretization" that would allocate more points where the cumulative reward function F_t is high. In the next sub-section we consider the ALF algorithm where we use adaptive sampling techniques such as MCMC and PMC which are designed for sampling from (possibly high dimensional) continuous distributions.

1.3 A Population Monte-Carlo sampling technique

The idea of sampling techniques such as Metropolis-Hasting (MH) or other MCMC (Monte-Carlo Markov Chain) methods (see e.g. [21, 22]) is to build a Markov chain that has p_t as its equilibrium distribution, and starting from an initial distribution, iterates its transition kernel K times so as to approximate p_t. Note that the rate of convergence of the distribution towards p_t is exponential

with K (see e.g. [23]): $\delta(k) \leq (2\epsilon)^{k/\tau(\epsilon)}$, where $\delta(k)$ is the total variation distance between p_t and the distribution at step k, and $\tau(\epsilon) = \min\{k; \delta(k) \leq \epsilon\}$ is the so called mixing time of the Markov Chain ($\epsilon < 1/2$).

Thus sampling $\theta_t \sim p_t$ only requires being able to compute $w_t(\theta)$ at a finite number of points K (the number of transitions of the corresponding Markov chain needed to approximate the stationary distribution p_t). This is possible whenever the reward functions f_t can be stored by using a finite amount of information, which is the case in the applications to learning, described in the next section.

However, using MCMC at each time step to sample from a distribution p_t which is similar to the previous one p_{t-1} (since the cumulative functions F_t do not change much from one iteration to the next) is a waste of MC transitions. The exponential decay of $\delta(k)$ depends on the mixing time $\tau(\epsilon)$ which depends on both the target distribution and the transition kernel, and can be reduced when considering efficient methods based on interacting particles systems. The population Monte-Carlo (PMC) method (see e.g. [24]) approximates p_t by a population of N particles $(x_{t,k}^{1:N})$ which evolve (during $1 \leq k \leq K$ rounds) according to a transition/selection scheme:

- At round k, the **transition step** generates a successor population $\widetilde{x}_{t,k}^{1:N} \overset{iid}{\sim} g_{t,k}(x_{t,k-1}^{1:N}, \cdot)$ according to a transition kernel $g_{t,k}(\cdot,\cdot)$. Then likelihood ratios are defined as $w_{t,k}^{1:N} = \frac{p_t(\widetilde{x}_{t,k}^{1:N})}{g(x_{t,k-1}^{1:N}, \widetilde{x}_{t,k}^{1:N})}$,
- The **selection step** resamples N particles $x_{t,k}^i = \widetilde{x}_{t,k}^{I_i}$ for $1 \leq i \leq N$ where the selection indices $(I_i)_{1 \leq i \leq N}$ are drawn (with replacement) from the set $\{1 \dots N\}$ according to a multinomial distribution with parameters $(w_{t,k}^i)_{1 \leq i \leq N}$

At round K, one particle (out of N) is selected uniformly randomly, which defines the sample θ_t that is returned by the sampling technique. Some properties of this approch is that the proposed sample tends to an unbiased independent sample of p_t (when either N or $K \to \infty$). We do not provide additional details about this method here because this is not the object of this paper but we refer the interested reader to [24] for discussion about the choice of good kernels $g_{t,k}$ and automatic tuning methods of the parameter K and number of particles N, and to [**?**] for a theoretical analysis of the sampling bias.

When using this sampling techniques in the ALF algorithm, since the distribution p_{t+1} does not differ much from p_t, we can initialize the particles at round $t+1$ with the particles obtained at the previous round t at the last step of the PMC sampling: $x_{t+1,1}^i \overset{\text{def}}{=} x_{t,K}^i$, for $1 \leq i \leq N$. In the numerical experiments reported in the next sub-section, this enabled to reduce drastically the number of rounds K per time step (less than 5 in all experiments below).

1.4 Numerical experiments

For illustation, let us consider the problem defined by: $\Theta = [0,1]^d$, $f_t(\theta) = (1 - ||\theta - \theta_t||/\sqrt{d})^3$ where $\theta_t = t/T\,(1, \dots, 1)'$. The optimal θ^* (i.e. $\arg\max_\theta F_T(\theta)$) is

$1/2\,(1,\ldots,1)'$. Figure 2 plots the expected regret $\sup_{\theta\in\Theta}\mathbb{E}R_T(\theta)$ (with $T=100$, averaged over 10 experiments) as a function of the parameter N (number of sampling points/particles) for two sampling methods: the random grid mentioned in the end of Section 1.2 and the PMC method. We considered two values of the space dimension: $d=2$ and $d=20$. Note that the uniform discretization technique is not applicable in the case of dimension $d=20$ (because of the curse of dimensionality). We used $K=5$ steps and used a Gaussian centered kernel $g_{t,k}$ of variance $\sigma^2=0.1$ for the PMC method.

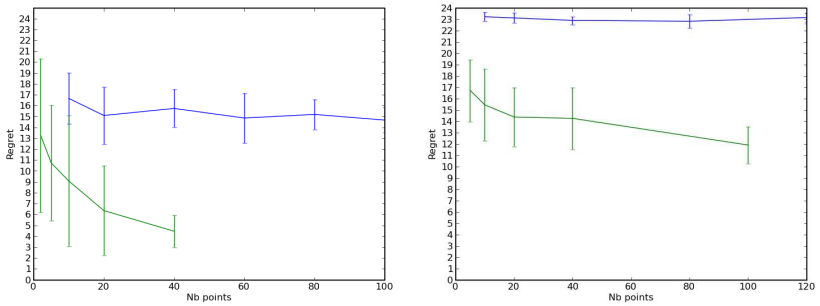

Fig. 2. Regret as a function of N, for dimensions $d=2$ (left figure) and 20 (right figure). In both figures, the top curve represents the grid sampling and the bottom curve the PMC sampling

Since the complexity of sampling from a PMC method with N particles and from a grid of N points is not the same, in order to compare the performance of the two methods both in terms of regret and runtime, we plot in Figure 3 the regret as a function of the CPU time required to do the sampling, for different values of N.

As expected, the PMC method is more efficient since its allocation of points (particles) depends on the cumulative rewards F_t (it thus may be considered as an adaptive algorithm).

2 Applications to learning problems

2.1 Online regression

Consider an online adversarial regression problem defined as follows: at each round t, an opponent selects a couple (x_t, y_t) where $x_t \in \mathcal{X}$ and $y_t \in \mathcal{Y} \subset \mathbb{R}$, and shows the input x_t to the learner. The learner selects a regression function $g_t \in \mathcal{G}$ and predicts $\hat{y}_t = g_t(x_t)$. Then the output y_t is revealed and the learner incurs the reward (or equivalently a loss) $l(\hat{y}_t, y_t) \in [0,1]$.

Since the true output is revealed, it is possible to evaluate the reward of any $g \in \mathcal{G}$, which corresponds to the full information case.

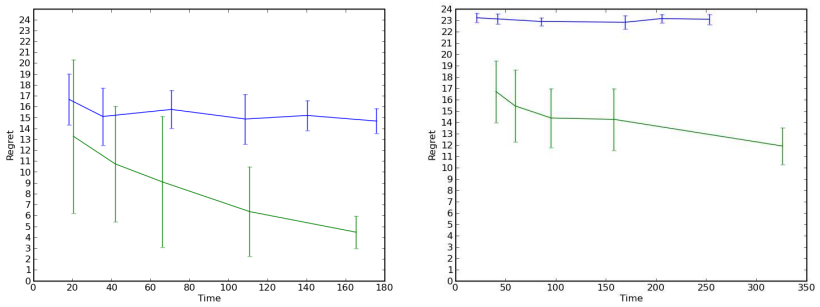

Fig. 3. Regret as a function of the CPU time used for sampling, for dimensions $d = 2$ (left figure) and 20 (right figure). Again, in both figures, the top curve represents the grid sampling and the bottom curve the PMC sampling.

Now, consider a parametric space $\mathcal{G} = \{g_\theta, \theta \in \Theta \subset \mathbb{R}^d\}$ of regression functions, and assume that the mapping $\theta \mapsto l(g_\theta(x), y)$ is Lipschitz w.r.t. θ with a uniform (over $x \in \mathcal{X}, y \in \mathcal{Y}$) Lipschitz constant $\lambda < \infty$. This happens for example when $\mathcal{X}$ and $\mathcal{Y}$ are compact domains, the regression $\theta \mapsto g_\theta$ is Lipschitz, and the loss function $(u, v) \mapsto l(u, v)$ is also Lipschitz w.r.t. its first variable (such as for e.g. L_1 or L_2 loss functions) on compact domains.

The online learning problem consists in selecting at each round t a parameter $\theta_t \in \Theta$ such as to optimize the accuracy of the prediction of y_t with $g_{\theta_t}(x_t)$. If we define $f_t(\theta) \stackrel{\text{def}}{=} l(g_\theta(x), y)$, then applying the ALF algorithm described previously (changing rewards into losses by using the transformation $u \mapsto 1 - u$), we obtain directly that the expected cumulative loss of the ALF algorithm is almost as small as that of the best regression function in $\mathcal{G}$, in the sense that:

$$\mathbb{E}\Big[\sum_{t=1}^{T} l_t\Big] - \inf_{g \in \mathcal{G}} \mathbb{E}\Big[\sum_{t=1}^{T} l(g(x_t), y_t)\Big] \leq 2\sqrt{dT \ln(d^\alpha \lambda T)},$$

where $l_t \stackrel{\text{def}}{=} l(g_{\theta_t}(x_t), y_t)$. To illustrate, consider a feedforward neural network (NN) [25] with parameter space Θ (the set of weights of the network) and one hidden layer. Let n and m be the number of input (respectively hidden) neurons. Thus if $x \in \mathcal{X} \subset \mathbb{R}^n$ is the input of the NN, a possible NN architecture would produce the output: $g_\theta(x) \stackrel{\text{def}}{=} \theta^o \cdot \sigma(x)$ with $\sigma(x) \in \mathbb{R}^m$ and $\sigma(x)_l \stackrel{\text{def}}{=} \sigma(\theta^i_l \cdot x)$ (where σ is the sigmoid function) is the output of the l-th hidden neuron. Here $\theta = (\theta^i, \theta^o) \in \Theta \subset \mathbb{R}^d$ the set of (input, output) weights (thus here $d = n \times m + m$).

The Lipschitz constant of the mapping $\theta \mapsto g_\theta(x)$ is upper bounded by $\sup_{x \in \mathcal{X}, \theta \in \Theta} ||x||_\infty ||\theta||_\infty$, thus assuming that the domains $\mathcal{X}$, $\mathcal{Y}$, and Θ are compacts, the assumption that $\theta \mapsto l(g_\theta(x), y)$ is uniformly (over $\mathcal{X}, \mathcal{Y}$) Lipschitz w.r.t. θ holds e.g. for L_1 or L_2 loss functions, and the previous result applies.

Now, as discussed above about the practical aspects of the ALF algorithm, in this online regression problem, the knowledge of the past input-output pairs $(x_s, y_s)_{s<t}$ enables to compute the weight $w_t(\theta) = \exp(\eta \sum_{s=1}^{t-1} l(g_\theta(x_s), y_s))$ for any $\theta \in \Theta$, and thus enables to use a PMC algorithm to sample θ_t from the distribution p_t. Up to our knowledge, we believe this is a first regret bound analysis of online learning for non-linear NN regression, in an adversarial setting.

2.2 Online classification

Now consider the problem of online classification (i.e. when the set of labels $\mathcal{Y}$ is finite). Here we can no longer make the assumption that the classifier's prediction $g_\theta(x) \in \mathcal{Y}$ is Lipschitz w.r.t. the parameter θ (and neither that the loss function $l(y, y') = \mathbb{I}_{\{y=y'\}}$ is Lipschitz w.r.t. its first variable). One way to circumvent this problem is to consider a class $\mathcal{G} = \{g_\theta, \theta \in \Theta\}$ of stochastic classifiers, so that $g_\theta(y|x)$ represents the probability of predicting label y given input x. The ALF algorithm would apply as follows: at round t, the algorithms chooses $\theta_t \in \Theta$ and samples the prediction $\hat{y}_t$ from the distribution $g_{\theta_t}(\cdot|x_t)$.

When the label y_t is revealed, the loss function $f_t(\theta) \stackrel{\text{def}}{=} g_\theta(y_t|x_t)$ for all classifiers g_θ may be computed. Thus assuming that the mapping $\theta \mapsto g_\theta(y|x)$ is Lipschitz w.r.t. θ with uniform (over $\mathcal{X}\times\mathcal{Y}$) Lipschitz constant λ, then Theorem 1 applies, and we have that

$$\underbrace{\sup_{g\in\mathcal{G}} \mathbb{E}\left\{\sum_{t=1}^{T} g(y_t|x_t)\right\}}_{\substack{\text{Exp. nb. of correct} \\ \text{predictions of best classifier}}} - \underbrace{\mathbb{E}\left\{\sum_{t=1}^{T} g_{\theta_t}(y_t|x_t)\right\}}_{\substack{\text{Exp. nb. of correct} \\ \text{predictions of ALF algo.}}} \leq 2\sqrt{dT \ln(cd^\alpha \lambda T)}$$

which says that the expected number of good predictions of the ALF algorithm is almost as good as that of the best classifier in $\mathcal{G}$. An example of such parametric regression setting is the case of neural networks (parameterized by θ) where the activation of the output neurons (one for each label y of $\mathcal{Y}$), up to some renormalization, define the probability distribution $g_\theta(y|x)$.

2.3 Online classification with bandit information

In the previous section, the information revealed by the opponent enables to compute the reward (or loss) function $f_t(\theta)$ for all $\theta \in \Theta$. In the bandit information case considered now only the reward $f_t(\theta_t)$ of the selected action is revealed. Under our Lipschitz assumption on the functions, the knowledge of f_t at a point θ_t reveals very few information about f_t elsewhere. Thus we cannot expect to derive tight regret bounds in general. However we can obtain interesting bounds in the case when the reward function f_t may actually be coded by a finite amount of information. We illustrate this setting on the online classification problem described in Section 2.2 but with the difference that the true label $y_t \in \mathcal{Y} = \{1, \ldots, K\}$ is not revealed at each round: the only available information

is $Z_t \stackrel{\text{def}}{=} \mathbb{I}_{\{\hat{y}_t = y_t\}}$, i.e. whether the prediction $\hat{y}_t$ is correct or not. An example of applications is the problem of web advertisement systems, where the user's click is the only received feedback.

Initialization: Set $w_1(\theta) = 1$ for all $\theta \in \Theta$.

For each round $t = 1, 2, \ldots, T$

(1) The adversary chooses $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ and shows x_t to the learner,

(2) The learner chooses $\theta_t \sim p_t$, where $p_t(\theta) \stackrel{\text{def}}{=} \dfrac{w_t(\theta)}{\int_\Theta w_t(\theta) d\theta}$, and predicts $\hat{y}_t \sim q_{t,\theta_t}$, where $q_{t,\theta}(y) \stackrel{\text{def}}{=} (1-\gamma) g_{\theta_t}(y|x_t) + \frac{\gamma}{K}$,

(3) The learner sees the (bandit) information $Z_t \stackrel{\text{def}}{=} \mathbb{I}_{\hat{y}_t = y_t}$, from which he defines $\tilde{f}_t(\theta) \stackrel{\text{def}}{=} \frac{g_\theta(\hat{y}_t|x_t)}{q_t(\hat{y}_t)} Z_t$, where $q_t(y) \stackrel{\text{def}}{=} \int_\Theta p_t(\theta) q_{t,\theta}(y) d\theta$, for any $y \in \mathcal{Y}$.

(4) The weight function w_t is updated according to $w_{t+1}(\theta) = w_t(\theta) e^{\eta \tilde{f}_t(\theta)}$, for all $\theta \in \Theta$.

Fig. 4. The Adversarial Lipschitz Bandit Classifier (ALBC algo)

Again, we consider a parametric family of stochastic classifiers $\mathcal{G} = \{g_\theta, \theta \in \Theta\}$, where $g_\theta(y|x)$ corresponds to the probability of selecting $y \in \mathcal{Y}$ given the input x. Now, in each round, a classifier g_{θ_t} is selected (by sampling $\theta_t \sim p_t$) and a prediction $\hat{y}_t$ is made. However, in this bandit setting, the feedback information $Z_t = \mathbb{I}_{\{\hat{y}_t = y_t\}}$ does not enable to evaluate the performance $f_t(\theta) \stackrel{\text{def}}{=} g_\theta(y_t|x_t)$ of any classifiers g_θ, $\theta \in \Theta$. Instead, we randomize the prediction by considering a mixture distribution between g_{θ_t} and the uniform distribution: $\hat{y}_t \sim q_{t,\theta_t}$, where $q_{t,\theta}$ is the distribution over the labels $\mathcal{Y}$ defined by $q_{t,\theta}(y) \stackrel{\text{def}}{=} (1-\gamma) g_\theta(y|x_t) + \frac{\gamma}{K}$.

This idea is close to the Exp4 algorithm in [3]. Given the information Z_t, we build an estimate $\tilde{f}_t(\theta)$ of the performance $f_t(\theta)$ of any classifiers g_θ: $\tilde{f}_t(\theta) \stackrel{\text{def}}{=} \frac{g_\theta(\hat{y}_t|x_t)}{q_t(\hat{y}_t)} Z_t$, where $q_t(y) \stackrel{\text{def}}{=} \mathbb{E}_{\theta \sim p_t}[q_{t,\theta}(y)]$, for any $y \in \mathcal{Y}$. This estimate is unbiased since:

$$\begin{aligned}
\mathbb{E}_{\theta_t, \hat{y}_t} \tilde{f}_t(\theta) &= \int_\Theta p_t(\theta') \sum_{y \in \mathcal{Y}} \frac{q_{t,\theta'}(y) g_\theta(y|x_t)}{q_t(y)} \mathbb{I}_{\{y = y_t\}} d\theta' \\
&= \int_\Theta p_t(\theta') \frac{q_{t,\theta'}(y) g_\theta(y_t|x_t)}{q_t(y_t)} d\theta' = g_\theta(y_t|x_t) = f_t(\theta)
\end{aligned}$$

Figure 4 describes this Adversarial Lipschitz Bandit Classifier (ALBC) algorithm. The next result assesses the expected performance of the ALBC algorithm $\sum_{t=1}^T \mathbb{I}_{\{\hat{y}_t = y_t\}}$ in comparison with the expected performance of the best classifier

$g \in \mathcal{G}$, in terms of number of correct predictions. Define the regret:

$$R_T(\theta) \stackrel{\text{def}}{=} \sum_{t=1}^{T} g_\theta(y_t|x_t) - \mathbb{E}\Big[\sum_{t=1}^{T} \mathbb{I}_{\{\hat{y}_t = y_t\}}\Big].$$

The ALBC algorithm has a regret $\sup_{\theta\in\Theta} \mathbb{E}R_T(\theta) \leq 4\sqrt{KdT\ln(cd^\alpha \lambda T)}$ (the proof is omitted from this extended abstract but follows the same lines as the proof of ALF algorithm combined with EXP4 ideas). Notice that like in the multi-armed bandit problem, in this bandit setting, the regret suffers from an additional factor K per round (i.e. $\sqrt{T}$ is replaced by $\sqrt{KT}$ in the bound), compared to the full information case.

A practical algorithm. A practical implementation of the ALBC algorithm requires being able to sample θ_t from p_t. The key difference with the technique detailed in Section 1.3 is that in the ALBC algorithm, the functions $\tilde{f}_t(\theta)$ depend on $q_t(\hat{y}_t)$ which is not directly known. However a refined MCMC or PMC algorithm is possible: at round t, assume that we have kept in memory the information: $H_{<t} \stackrel{\text{def}}{=} \{(x_s, \hat{y}_s, Z_s, q_s(\hat{y}_s))_{s<t}\}$.

We now show that (1) this is possible, and (2) this is sufficient for sampling $\theta_t \sim p_t$. We prove (1) recursively by showing that from $H_{<t}$ we are able to calculate $q_t(\hat{y}_t)$ (the other pieces of information $x_t, \hat{y}_t$, and Z_t are available at the end of round t). Thus we only need to prove that from $H_{<t}$, we can sample $\theta_t \sim p_t$ and compute $q_t(\hat{y}_t)$. But since $q_t(\hat{y}_t)$ is the expectation of $q_{t,\theta}(\hat{y}_t)$ for $\theta \sim p_t$, we may consider a MCMC or PMC method where the same Markov chain (having p_t as stationnary distribution) or particle population serves for both sampling $\theta_t \sim p_t$ and estimating $q_t(\hat{y}_t)$. Finally, this is possible since the pointwise evaluation of w_t (thus of p_t up to a renormalization constant) only depends on information in $H_{<t}$.

3 Conclusion

We have considered the adversarial online learning framework in the case of Lipschitz functions. In the full information case, the bound shows the same rate $\sqrt{dT}$ as for linear functions. This enables to derive similar performance bounds for online regression and classification, thus extending previous results to non-linear parametric approximation, such as neural networks. Our main contribution was to consider a continuous extension of the EWF algorithm (ALF algorithm) for which we provide geometrical conditions for sound regret analysis, and discuss the use of different approximation schemes and especially the use of a PMC sampling method compared to non adaptive sampling methods. We provided experiments showing the benefit of using a PMC sampling method for minimizing regret under computational time constraint compared to naive random grid.

We applied this result to derive bounds for (full information) regression and classification online learning problems and (bandit information) K-classes classification problems where the revealed information is the correctness of the prediction. We derived a regret bound on the expected number of mistakes of order $\sqrt{dTK}$, and illustrate the case of a Neural Networks architecture.

A Proof of Theorem 1 (ALF algorithm)

We start by following the usual proof for exponentially weighted forecasting. Define $W_t \stackrel{\text{def}}{=} \int_\Theta w_t$. For any $t \in \{1, \ldots, T\}$, we have:

$$\frac{W_{t+1}}{W_t} = \frac{\int_\Theta \exp(\eta F_t)}{\int_\Theta \exp(\eta F_{t-1})} = \int_\Theta p_t(\theta) \exp(\eta f_t(\theta)).$$

Since $\exp(u) \leq 1 + u + u^2$ for $u \leq 1$, then, whenever $\eta \leq 1$, we have $\frac{W_{t+1}}{W_t} \leq 1 + \eta \int_\Theta p_t f_t + \eta^2 \int_\Theta p_t f_t^2$. Moreover, since $W_1 = \mu(\Theta)$, we get:

$$\ln(W_{T+1}) \leq \eta \sum_{t=1}^{T} \int_\Theta p_t f_t + T\eta^2 + \ln(\mu(\Theta)). \tag{4}$$

Let us write $h(\theta) \stackrel{\text{def}}{=} \exp(\eta F_T(\theta))$, and $h^* \stackrel{\text{def}}{=} \max_{x \in \Theta} h(\theta)$. We have that

$$\begin{aligned} |h(\theta_1) - h(\theta_2)| &\leq \eta |F_T(\theta_1) - F_T(\theta_2)| h^* \\ &\leq \eta \lambda T h^* ||\theta_1 - \theta_2||, \end{aligned} \tag{5}$$

since the function F_T is λT-Lipschitz. Let θ^* be any point of maximum of h, and define $\pi(\theta) \stackrel{\text{def}}{=} \max(0, 1 - \eta\lambda T ||\theta - \theta^*||)$. Then for all $\theta \in \Theta$,

$$h(\theta) \geq h^* \pi(\theta). \tag{6}$$

Indeed, this holds for any $\theta \notin B(\theta^*, 1/(\eta\lambda T))$ where $B(\theta, r)$ is the ball $\{x', ||x - x'|| \leq r\}$, since in that case, $\pi(\theta) = 0$. Now if there were some $\theta \in B(\theta^*, 1/(\eta\lambda T))$ such that $h(\theta) < h^* \pi(\theta)$, then we would have: $h(\theta^*) - h(\theta) > \eta\lambda T h^* ||x - x^*||$, which would contradict the Lipschitz property (5) of h.

Notice that π is a pyramid function with base $B(\theta^*, 1/(\eta\lambda T))$ and height 1. We now state a Lemma that will enable us to derive a lower bound on $\int_\Theta \pi$.

Lemma 1. *For any $\theta^* \in \Theta$, $r > 0$, let π be the function defined by $\pi(\theta) \stackrel{\text{def}}{=} \max(0, 1 - ||x - x^*||/r)$. Then:*

$$\int_\Theta \pi \geq \frac{1}{(d+1)\kappa(d)} \min\left[\mu\big(B(\theta^*, r)\big), \mu(\Theta)\right]$$

Proof.

$$\begin{aligned}\int_\Theta \pi &= \int_{\mathbb{R}^D} \mathbb{I}_{\theta\in\Theta\cap B(\theta^*,r)}(1-\frac{||\theta^*-\theta||}{r})\mu(d\theta)\\ &= \int_{\mathbb{R}^D} \mathbb{I}_{\theta\in\Theta\cap B(\theta^*,r)}\int_0^1 \mathbb{I}_{||\theta^*-\theta||\leq\alpha r}d\alpha\mu(d\theta)\\ &= \int_0^1\int_{\mathbb{R}^D} \mathbb{I}_{\theta\in\Theta\cap B(\theta^*,\alpha r)}\mu(d\theta)d\alpha\\ &= \int_0^1 \mu(\Theta\cap B(\theta^*,\alpha r))d\alpha\end{aligned}$$

Now, using the definition of $\kappa(d)$ from (1),

$$\int_\Theta \pi \geq \int_0^1 \frac{1}{\kappa(d)}\min[\alpha^d\mu(B(\theta^*,r)),\mu(\Theta)]d\alpha$$

We deduce that if $\mu(\Theta)\geq\mu(B(\theta^*,r))$ then $\int_\Theta\pi\geq\frac{\mu(B(\theta^*,r))}{(d+1)\kappa(d)}$. And otherwise, $\exists\alpha_0<1$ such that $\mu(\Theta)=\alpha_0^d\mu(B(\theta^*,r))$ thus we have $\int_\Theta\pi\geq\frac{\mu(\Theta)}{\kappa(d)}(1-\alpha_0+\frac{\alpha_0}{d+1})\geq\frac{\mu(\Theta)}{(d+1)\kappa(d)}$ and the Lemma is proved.

We apply this Lemma with the π function and $r=1/\eta\lambda T$ to obtain:

$$\int_\Theta \pi \geq \frac{1}{(d+1)\kappa(d)}\min\big[\mu\big(B(\theta^*,\frac{1}{\eta\lambda T})\big),\mu(\Theta)\big]$$

Now using (6) together with the previous bound combined with Assumption A1 (i.e. $\kappa(d)\leq\kappa^d$ and $\mu\big(B(\theta^*,r)\big)\geq(r/(\kappa' d^\alpha)^d)$, we derive the lower bound:

$$\int_\Theta h \geq h^*\min\big[\frac{1}{(cd^\alpha\eta\lambda T)^d},\frac{\mu(\Theta)}{c^d}\big].$$

where we set $c=2\kappa\max(\kappa',1)$.

From its definition, $W_{T+1}=\int_\Theta h$, thus

$$\ln(W_{T+1})\geq\eta\max_{\theta\in\Theta}F_T(\theta)-\ln\left(\max\big[(cd^\alpha\eta\lambda T)^d,\frac{c^d}{\mu(\Theta)}\big]\right),$$

which, together with (4) yields:

$$\sup_{\theta\in\Theta}F_T(\theta)-\sum_{t=1}^T\int_\Theta p_t f_t\leq T\eta+\frac{1}{\eta}\max\big[d\ln(cd^\alpha\eta\lambda T)+\ln(\mu(\Theta)),d\ln c\big].$$

Since $\int_\Theta p_t f_t=\mathbb{E}_t[f_t(\theta_t)]$, where $\mathbb{E}_t$ denotes the expectation w.r.t. the choice of $\theta_t\sim p_t$, we deduce that the expected regret (w.r.t. the internal randomization of the learner) of any $\theta\in\Theta$ is bounded according to:

$$\mathbb{E}R_T(\theta)\leq T\eta+\frac{1}{\eta}(d\ln(cd^\alpha\eta\lambda T)+\ln(\mu(\Theta))),$$

whenever $d\ln(d^\alpha \eta\lambda T) \geq -\ln(\mu(\Theta))$.

Now, for the high probability result, if we introduce $Y_t = \int_\Theta p_t f_t - f_t(\theta_t)$ and $\mathcal{F}_{<t}$ the σ-algebra generated by the past random decisions, then $\mathbb{E}[Y_t|\mathcal{F}_{<t}] = 0$, thus $Y_1, ..., Y_T$ is a martingale difference sequence, and since $f_t \in [0,1]$, $|Y_t| \leq 1$ a.s., using Hoeffding-Azuma's inequality (see e.g. [26]), we obtain that with probability at least $1-\beta$,

$$\sum_{t=1}^{T} \int_\Theta p_t f_t \leq F_T + \sqrt{2T\ln(\beta^{-1})},$$

which enables to deduce (3).

References

1. Cesa-Bianchi, N., Lugosi, G.: Prediction, Learning, and Games. Cambridge University Press, New York, NY, USA (2006)
2. Auer, P., Cesa-bianchi, N., Freund, Y., Schapire, R.E.: Gambling in a rigged casino: The adversarial multi-armed bandit problem. In: In Proceedings of the 36th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press (1995) 322–331
3. Auer, P., Cesa-bianchi, N., Freund, Y., Schapire, R.E.: The nonstochastic multi-armed bandit problem. SIAM Journal on Computing **32** (2002) 2002
4. Poland, J.: Nonstochastic bandits: Countable decision set, unbounded costs and reactive environments. Theor. Comput. Sci. **397**(1-3) (2008) 77–93
5. Dani, V., Hayes, T., Kakade, S.: The price of bandit information for online optimization. In Platt, J., Koller, D., Singer, Y., Roweis, S., eds.: Advances in Neural Information Processing Systems 20. MIT Press, Cambridge, MA (2008) 345–352
6. Abernethy, J., Hazan, E., Rakhlin, A.: Competing in the dark: An efficient algorithm for bandit linear optimization. In Servedio, R.A., Zhang, T., eds.: COLT, Omnipress (2008) 263–274
7. Cesa-Bianchi, N., Lugosi, G.: Combinatorial bandits. In: Conference on Computational Learning Theory. (2009)
8. Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research (2002) 397–422
9. Dani, V., Hayes, T.P., Kakade, S.M.: Stochastic linear optimization under bandit feedback. In: In submission. (2008)
10. Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent. In: ICML. (2003) 928–936
11. Hazan, E., Agarwal, A., Kale, S.: Logarithmic regret algorithms for online convex optimization. In: In COLT. (2006) 499–513
12. Bartlett, P., Hazan, E., Rakhlin, A.: Adaptive online gradient descent. In Platt, J.C., Koller, D., Singer, Y., Roweis, S.T., eds.: NIPS, MIT Press (2007)
13. Shalev-Shwartz, S.: Online Learning: Theory, Algorithms, and Applications. PhD thesis (July 2007)
14. Abernethy, J.D., Bartlett, P., Rakhlin, A., Tewari, A.: Optimal strategies and minimax lower bounds for online convex games. Technical Report UCB/EECS-2008-19, EECS Department, University of California, Berkeley (Feb 2008)

15. Kleinberg, R., Slivkins, A., Upfal, E.: Multi-armed bandit problems in metric spaces. In: Proceedings of the 40th ACM Symposium on Theory of Computing. (2008) 681–690
16. Bubeck, S., Munos, R., Stoltz, G., Szepesvári, C.: Online optimization of X-armed bandits. In: NIPS. (2008)
17. Littlestone, N., Warmuth, M.: The weighted majority algorithm. Information and Computation **108** (1994) 212–261
18. Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D.P., Shapire, R., Warmuth, M.: How to use expert advice. Journal of the ACM **44**(3) (1997) 427–485
19. Auer, P., Cesa-bianchi, N., Gentile, C.: Adaptive and self-confident on-line learning algorithms. Journal of Computer and System Sciences **64** (2000) 2002
20. Stoltz, G.: Incomplete information and internal regret in prediction of individual sequences. PhD thesis (2005)
21. Gilks, W., Richardson, S., Spiegelhalter, D.: Markov Chain Monte Carlo in Practice. Chapman Hall/CRC, Boca Raton, FL (1996)
22. Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.: An introduction to mcmc for machine learning. Journal of Machine Learning Research **50** (2003) 5–43
23. Levin, D.A., Peres, Y., Wilmer, E.L.: Markov Chains and Mixing Times. American Mathematical Society (2008)
24. Douc, R., Guillin, A., Marin, J., Robert, C.: Minimum variance importance sampling via population monte carlo. Esaim P&S **11** (2007)
25. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer (August 2006)
26. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer-Verlag, New York (1996)