Sequential and reinforcement learning: Stochastic Optimization I

Summary

This session describes the important and nowadays framework of on-line learning and estimation. This kind of problem arises in bandit games (see below for details) and in optimization of big data problems that involve so massive data sets that the user cannot imagine exploiting the entire data with a batch algorithm. Instead, we are forced to use some subset of observations sequentially to produce both : tractable algorithms, fast predictions, efficient statistical estimators.

We will describe below the method of Stochastic Gradient Algorithm, with application in

- *Optimization of convex functions (regression, SVM, Lasso)*
- *Optimization with non convex functions : E.M. algorithm, Quantile estimation*

As well, we will introduce later a second family of method that relies on pure randomness exploration strategy with Markov chains. We will describe algorithms for solving the problem of simulations of posterior distribution through another family of stochastic methods :

- *M.H. M-C algorithms*
- *Adjusted Langevin algorithms*

At last, we will briefly describe an ultimate method for solving the minimization of a non convex function with multiple local traps, by using simulated annealing. This problem will be illustrated on the so-called travelling salesman problem.

1 Stochastic Gradient Descent (aka S.G.D.)

1.1 Gradient descent

1.1.1 Convex case

Let us consider a function $f : \mathbb{R}^p \longmapsto \mathbb{R}$, which is convex and $\mathcal{C}^2$ (for the sake of simplicity). We are interested in finding the minimum of f over $\mathbb{R}^p$. A natural method relies on a discretization of the so-called gradient descent :

$$\dot{x}_t = -\nabla f(x_t)dt. \tag{1}$$

It is easy to check that $E(t) = f(x_t)$ is a decreasing function all along the time :

$$E'(t) = -\|\nabla f(x_t)\|^2.$$

Assume now that $f(x^*) = \min_{\mathbb{R}^p} f = 0$, we can prove that

$$E(t) - E(0) = -\int_0^t \|\nabla f(x_s)\|^2 ds,$$

so that

$$\int_0^t \|\nabla f(x_s)\|^2 ds \leq E(0).$$

This last inequality immediately implies that

$$\lim_{t\longrightarrow+\infty} \nabla f(x_t) = 0.$$

The convexity of f permits to conclude the convergence of (1) :

$$\lim_{t\longrightarrow+\infty} x_t = x^*.$$

1.1.2 Strongly convex case

We can derive convergence rates if we add a slightly stronger hypothesis on f. Assume now that f is α-strongly convex : it means that $x \longmapsto f(x) - \alpha\|x - x^*\|^2$ is still convex, or equivalently that

$$D^2 f \geq \alpha Id,$$

in the following sense :

$$\forall x \in \mathbb{R}^p \qquad {}^t x D^2(f)(x) x \geq \alpha\|x\|^2.$$

A simple consequence is that

$$\forall x \in \mathbb{R}^p \qquad f(x) \geq \alpha \|x - x^*\|_2^2.$$

A typical example (involved in the linear models) :

$$f(x) = \|Ax - b\|^2 \qquad \text{with} \qquad A \in \mathcal{M}_{n,p}(\mathbb{R}),$$

such that ${}^tAA$ is an invertible matrix. We can be more precise about the behaviour of (1). We introduce $F(t) = \|x_t - x^*\|^2$ and compute

$$F'(t) = -\langle x_t - x^*, \nabla f(x_t) \rangle.$$

But any convex function with minimum x^* always satisfies

$$\langle \nabla f(x), x - x^* \rangle \geq f(x),$$

so that

$$F'(t) \leq -f(x_t) \leq -\alpha F(t).$$

We then conclude that

$$F(t) \leq F(0) e^{-\alpha t},$$

leading to an exponential convergence rate of x_t towards its target x^*.

1.2 Stochastic settings

We are now interested in the situation where the numerical scheme is discrete, given by an Euler explicit scheme (with step size $(\alpha_k)_{g \geq 1}$) :

$$x_{k+1} = x_k - \alpha_k d_k(x_k), \tag{2}$$

where $d_k(x_k)$ is a " descent direction" that should be made close to the gradient of f at point x_k. In some situation, it is not reasonnable to imagine using $d_k(x) = \nabla f(x)$ because of

- computationnal issues (computing ∇f may be costly)
- availability (the computation of ∇f is not possible).

However, it is possible to produce an *unbiased* estimate of $\nabla f(x_k)$ at time k while assuming the important hypothesis :

$$\forall k \in \mathbb{N} \quad \forall x \in \mathbb{R}^p \qquad \mathbb{E}[d_k(x)] = \nabla f(x) \qquad \text{and} \qquad Var(d_k(x)) \leq \sigma^2.$$

It means that now, we have access to an unbiased estimator of ∇f at each iteration of the algorithm, with bounded variance. The natural question raised by this framework is as follows. Does algorithm (2) still converges to x^*, if we tune suitably the step-size sequence $(\alpha_k)_{g \geq 1}$) ?

1.3 Examples of applications of SGD

1.3.1 f as an average of convex functions

A famous use of S.G.D. concerns the situation where f is given by

$$\forall x \in \mathbb{R}^p \qquad f(x) = \mathbb{E}_{U \sim P}[f(x, U)],$$

where U is a random variable of distributions P and for any u : $f(., u)$ is convex. If we can sample from the distribution P, then S.G.D. can be written as

$$x_{k+1} = x_k - \alpha_k \nabla_x f(x_k, U_k) \qquad \text{where} \qquad U_k \sim P.$$

Above, we implicitely assume that $\nabla_x f(x_k, U_k)$ is easy to handle, which is not the case for the whole gradient $\nabla_x f(x_k)$.

Another great advantage is that this approach permits generally to handle sequential arrivals of new observations X_n, with a very low cost of updates.

1.3.2 Linear regression

The loss L is given by

$$L(w) = \frac{1}{2} \sum_{i=1}^{n} [Y_i - \langle w, X_i \rangle]^2.$$

We then immediately adapt the on-line regression with stochastic gradient using the direction descent :

$$d_k(w) = [\langle w_k, X_{i_k} \rangle - Y_{i_k}].X_{i_k},$$

where (X_{i_k}, Y_{i_k}) is an observation uniformly picked in the training set. Note that $d_k(w)$ and X_{i_k} are vectors of size p. This stochastic step do not involve the inversion of the Fisher information matrix.

1.3.3 S.V.M. problem

The S.V.M. problem induces the minimization of a cost function that depends on a variable $w \in \mathbb{R}^{p+1}$. The S.V.M. is dedicated to a supervised classification problem with n observations (X_i, Y_i) where $Y_i \in \{\pm 1\}$. The observation X_i is sent in a p dimensional space through a kernel ϕ.

The loss function is defined as

$$L(w) = \lambda \|w\|_2^2 + \sum_{i=1}^{n} \max\{0; 1 - Y_i[\langle w, \phi(X_i)\rangle]\}.$$

This function is convex in w and minimizing this loss corresponds in trying to find the best separation hyperplane in a p dimensional space. If n is huge, handle the gradient of L may be complicated. However, we can notice that

$$L(w) = \lambda \|w\|_2^2 + \mathbb{E}_{(X,Y)\sim\mathbb{P}_n} \max\{0; 1 - Y[\langle w, \phi(X)\rangle]\}.$$

Hence, we can replacement at each step the gradient of L by an unbiased estimation given by

$$d_k(w) = \lambda w - Y_k \langle w, \phi(X_k)\rangle \mathbf{1}_{Y_k\langle w,\phi(X_k)\rangle<1} \qquad \text{where} \qquad (X_k, Y_k) \sim \mathbb{P}_n.$$

Here, we can see that we only need **one** sample of the training set to produce an estimator of the gradient at each step of the algorithm.

1.3.4 Lasso

The Lasso loss is

$$L(w) = \lambda \|w\|_1 + \frac{1}{2} \sum_{i=1}^{n} [Y_i - \langle w, \phi(X_i)\rangle]^2$$

We can trivially apply the S.G.D. on this loss using again a sampling of $\mathbb{P}_n$ the empirical measure. Nevertheless, this kind of method fails in producing a sparse reconstruction. Instead, we can also randomize on the variables $(w_1, \ldots, w_p)$ by picking one coordinate of w uniformly at random among the p variables.

In that case, the descent becomes :

- Pick one variable j_k uniformly in $\{1, \ldots, p\}$.
- $$\tilde{w}_{k+1}^j = w_k^j - \mathbf{1}_{j=j_k} \partial_{j_k} \left(\frac{1}{2} \sum_{i=1}^{n} [Y_i - \langle w, \phi(X_i)\rangle]^2 \right).$$
- Use the proximal step
$$w_{k+1} = s_\lambda[\tilde{w}_{k+1}].$$

1.3.5 Quantile

The quantile of a real distribution may be of interest while looking for confidence sets. Recall that the quantile q_α is defined as

$$\mathbb{P}(X < q_\alpha) = \int_{-\infty}^{q_\alpha} p(s)ds = 1 - \alpha.$$

Now, imagine you want to find q_α from sequential arrivals of observations $(X_n)_{n\in\mathbb{N}}$, you are thus looking for the minimizer of the loss

$$L(q) = [\mathbb{P}[X \leq q] - (1 - \alpha)]^2,$$

whose gradient is given by

$$L'(q) = p(q)[\mathbb{P}[X \leq q] - (1 - \alpha)] = p(q)\,[\mathbb{E}[\mathbf{1}_{X\leq q}] - (1 - \alpha)]$$

A natural extension of the initial gradient algorithm will produce the following descent choice :

$$d_k(q_k) = \mathbf{1}_{X_k \leq q_k} - (1 - \alpha),$$

that solely depends on the observation X_k at time k and the current estimation q_k.

1.4 Examples in Matlab

Each time, try to modify some parameters (step size, functions to be minimized, initialization, etc.)

1.4.1 Convex minimization

The baseline (toy) functions :

```
function y=E1(x)
y=x.^2/2;

function y=gradE1(x)
y=x;
```

The Matlab script :

```
% Script of the S.G.D.
clear all
close all
%Step Size power parameter
alpha=1;

%Maximal number of iterations
Nstop=1000;

%Initialisation of the gradient and of the S.G.D.
x(1)=randn; xx(1)=randn;


for i=1:Nstop-1
   x(i+1)=x(i)-(i+1)^(-alpha)*gradE1(x(i));
   xx(i+1)=xx(i)-(i+1)^(-alpha)*[gradE1(xx(i))+randn];
end

plot(x)
hold on
plot(xx,'r')

legend('EGD','SGD')
```

1.4.2 Non-Convex minimization

```
function y=E2(x)
y=5*sin(x)+x.^2/2;

function y=gradE2(x)
y=5*cos(x)+x;
```

The Matlab script :

```
x(1)=3;

%Initialisation of the S.G.D.
xx(1)=3;

for i=1:Nstop-1
   x(i+1)=x(i)-(i+1)^(-alpha)*gradE2(x(i));
   xx(i+1)=xx(i)-(i+1)^(-alpha)*[gradE2(xx(i))+randn];
end

figure

plot(x)
hold on
plot(xx,'r')

legend('EGD','SGD')
```

1.4.3 Linear regression with (possibly) giant dataset

The Matlab script :

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Linear Model
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
%Dimension of the problem:
p=10;
% Step size parameter
alpha=1/2;
% Unknown parameter
theta=(2*rand(p,1)-1)*10;
% Number of observations:
nobs=1000;
```

```
esti(:,1)=zeros(p,1);

for i=1:nobs
    %Current Error
    s(i)=norm(theta-esti(:,i));
    % Sequential arrival of an observation (X,Y)
    X=10*randn(p,1);
    Y=sum(X.*theta)+randn;
    esti(:,i+1)=esti(:,i)+(i+1)^(-alpha)*(Y-sum(esti(:,i).*X))*X/(100*p);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evolution of the error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot(s)
```

1.4.4 Quantile estimation

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sequential quantile estimation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
% Alpha
alpha=0.01;
% Step size parameter
nu=1/2;

% Number of observations:
nobs=10000;
q(1)=0;

for i=1:nobs
    X=randn;
    q(i+1)=q(i)-i^(-nu)*((X<q(i))-(1-alpha));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evolution of the quantile estimation for the std gauss
r.v.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
plot(q)
```

1.5 Theoretical guarantees

1.5.1 Convergence results

A first important ingredient for a good behaviour of these stochastic algorithms is about the choice of the step size parameter. This choice is not so obvious. It can be shown that two baseline important properties are as follows :

$$\sum_{n=1}^{\infty} \gamma_n = +\infty \qquad \text{and} \qquad \sum_{n=1}^{\infty} \gamma_n^{1+\epsilon} < +\infty,$$

for any $\epsilon > 0$. In practice, it is important to choose

$$\gamma_n \sim C n^{-\alpha} \qquad \text{with} \qquad \alpha \in (1/2, 1].$$

Such a choice permits to establish the following results :

THÉORÈME 1. — *Assume that f is stongly convex and ∇f is L Lipschitz, and that the noise at each step has a bounded variance. Then $\gamma_n = cn^{-\alpha}$ with $\alpha \in (0, 1)$ leads to*

$$\mathbb{E}[f(X_n)] - \min_{\mathbb{R}^p} f \lesssim \gamma_n.$$

Consequently, the smaller the step-size, the lower the upper bound. Unfortunately, we cannot choose $\alpha = 1$ without any additional assumption that relies on the strong convexity of f. It can be shown that if $\gamma_n = c/n$ with c large enough, then the result of Theorem 1 remains true. These results are difficult (especially when $\alpha < 1/2$ and can be found in the earliest work of Benaïm and Duflo).

1.5.2 Polyak averaging

A very good alternative to a very careful choice of the step-size parameter (that would depend on the strong convexity of f or not), is to use a step size

$\gamma_n = C/n^\alpha$ with $\alpha \in (1/2, 1)$ and an additional Cesaro averaging procedure.

$$\bar{X}_n := \frac{1}{n}\sum_{k=1}^{n} X_k.$$

In that situation, we can show the next result.

THÉORÈME 2. — *If f is strongly convex and $\gamma_n = cn^{-\alpha}$ with $\alpha \in [1/2, 1)$, then*

$$\mathbb{E}[f(\bar{X}_n)] - \min_{\mathbb{R}^p} f \lesssim \frac{1}{n}.$$

Hence, Polyak averaging has the good property to catch the best rate achievable with strong convexity, without setting a sharp constant in front of an hypothetic step size $\gamma_n = c/n$. In practice, a good setup is $\gamma_n \sim 1/\sqrt{n}$ coupled with Polyak averaging.