Stochastic Approximation from Finance to Data Science

Gilles Pagès

LPSM-Sorbonne-Université

(Labo. Proba., Stat. et Modélisation)

M2 Probabilités & Finance

Novembre 2020

1. Optimization (deterministic, the origins)
2. Examples from Finance
   - Implicitation
   - Minimization
3. Learning procedures
   - Abstract Learning
   - Supervised Learning
   - Unsupervised Learning (clustering)
4. Stochastic algorithms/Approximation
   - From Robbins-Monro to Robbins-Siegmund
   - Stochastic Gradient Descent (*SGD*) and pseudo-*SGD*
5. Examples revisited by *SFD*
   - Numerical probability
   - Learning (supervised and unsupervised)
6. Application to Neural Networks and deep learning
   - Linear neural network
   - One hidden layer feedforward perceptron
   - Toward deep learning
   - Multilayer feedforward perceptron and Backpropagation

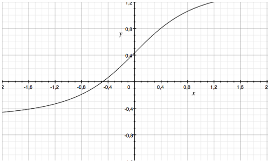
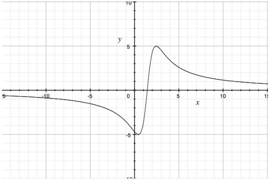
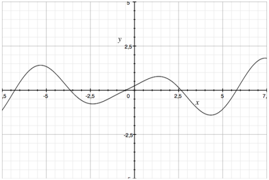
Table of Contents

1. Optimization (deterministic, the origins)
2. Examples from Finance
   - Implicitation
   - Minimization
3. Learning procedures
   - Abstract Learning
   - Supervised Learning
   - Unsupervised Learning (clustering)
4. Stochastic algorithms/Approximation
   - From Robbins-Monro to Robbins-Siegmund
   - Stochastic Gradient Descent (*SGD*) and pseudo-*SGD*
5. Examples revisited by *SFD*
   - Numerical probability
   - Learning (supervised and unsupervised)
6. Application to Neural Networks and deep learning
   - Linear neural network
   - One hidden layer feedforward perceptron
   - Toward deep learning
   - Multilayer feedforward perceptron and Backpropagation

- Zero search: One aims at finding a zero θ^* of a function $h : \mathbb{R}^d \to \mathbb{R}^d$. In view of generic notations in stochastic approximation, we will denote

$$h(\theta),\ \theta \in \mathbb{R}^d$$

rather than $h(x)$.

($d = 1$ is mandatory just for graphs).

- Various methods (I):
  - Local recursive zero search (standard): θ_0 be fixed and let $\gamma > 0$ be small enough. Set

$$\theta_{n+1} = \theta_n - \gamma h(\theta_n), \quad n \geq 0$$

- Various methods (II):
  - Local recursive zero search. if h is $\mathcal{C}^1$ (Newton-Raphson "false position" algoritm)

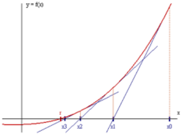
$$\theta_{n+1} = \theta_n - [J_h(\theta_n)]^{-1} h(\theta_n), \quad n \geq 0,$$

where $J_h(\theta)$ denotes the Jacobian of h at θ.

Idea: The tangent hyperplane is the best approximation of h (by an affine function)

$$h(\theta) \simeq h(\theta_n) + J_h(\theta_n)(\theta - \theta_n)$$

so θ_{n+1} is solution to $h(\theta_n) + J_h(\theta_n)(\theta - \theta_n) = 0$.



Very fast but also very unstable, especially when $J_h(\theta^*)$ is "small".

  - Yet another local recursive zero search if h $\mathcal{C}^1$ (Levenberg-Marquardt algorithm): Let $\lambda_n > 0$, $n \geq 1$,

$$\theta_{n+1} = \theta_n - \left[J_h(\theta_n) + \lambda_{n+1} I_d\right]^{-1} h(\theta_n), \quad n \geq 0.$$

turns out to be more stable... by an appropriate choice of λ_n.

- Various methods (III):
  - Global recursive zero search:

    – Idea: make the step decrease (not too fast) to "enlarge" in an adaptive way the convergence area of the algorithm...

    – Let γ_n, $n \geq 1$ satisfy

    $$\sum_{n\geq 1} \gamma_n = +\infty \text{ and } \sum_{n\geq 1} \gamma_n^2 < +\infty.$$

    – Set

    $$\theta_{n+1} = \theta_n - \gamma_{n+1} h(\theta_n), \; n \geq 0.$$

- To be continued...

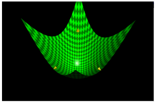
- BUT WARNING! All these methods require

h can be computed at a reasonable cost.

Minimizing a (potential function)

- Gradient descent (GD):
  Let $V : \mathbb{R}^d \to \mathbb{R}_+$, $\mathcal{C}^1$ with $\lim_{|x|\to+\infty} V(x) = +\infty$ so that $\mathrm{argmin}_{\mathbb{R}^d} V \neq \varnothing$.

$$\boxed{\text{How to compute } \mathrm{argmin}\ \&\ \min_{\mathbb{R}^d} V???}$$



  - If moreover V is convex, then

$$\mathrm{argmin}_{\mathbb{R}^d} V = \{\nabla V = 0\} \quad \text{(is a convex set)}$$

  – Solution: set $h = \nabla V$,

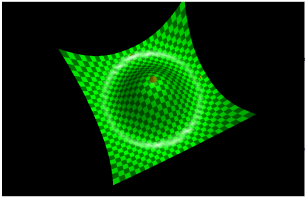
  – If ∇V Lipschitz, then (exercise)

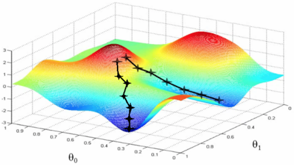
$$\theta_n \to \theta^* \in \{\nabla V = 0\} = \mathrm{argmin}_{\mathbb{R}^d} V \quad \text{as} \quad n \to +\infty.$$

- If V is not convex it often happens that

$$\operatorname{argmin} V \subsetneq \{\nabla V = 0\}.$$

Still set $h = \nabla V$ (what else?)





- Pseudo-gradient (back to zero search!):

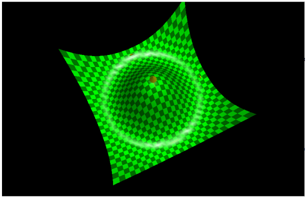
  The function h is often given (model) and (hopefully) there exists a Lyapunov function V s.t. $(h|\nabla V) \geq 0$ and

$$\{h = 0\} \simeq \{(h|\nabla V) = 0\} \qquad (\subset \text{ is ok!}).$$

  If $(d = 2)$, $\mathcal{H}(V)(x) = \begin{pmatrix} -\partial_{x_2} V \\ \partial_{x_1} V \end{pmatrix}$ (Hamiltonian of $\nabla V(x)$) and

$$h(x) = \lambda \nabla V(x) + \mu \mathcal{H}(V)(x)$$

  then, the above conditions are satisfied and $|h|^2$ has V-linear growth so that $\theta_n \to C(0;1)$ (if $\theta_0 \neq 0$) but does not converge "pointwise".



  However, on this example, $V(\theta_n) \to \mathrm{argmin}\, V$

- It may happen that $\{h = 0\} \neq \{(h|\nabla V) = 0\} \neq \{\nabla V = 0\} \neq \mathrm{argmin}\, V$!!.

Table of Contents

1. Optimization (deterministic, the origins)
2. Examples from Finance
   - Implicitation
   - Minimization
3. Learning procedures
   - Abstract Learning
   - Supervised Learning
   - Unsupervised Learning (clustering)
4. Stochastic algorithms/Approximation
   - From Robbins-Monro to Robbins-Siegmund
   - Stochastic Gradient Descent (*SGD*) and pseudo-*SGD*
5. Examples revisited by *SFD*
   - Numerical probability
   - Learning (supervised and unsupervised)
6. Application to Neural Networks and deep learning
   - Linear neural network
   - One hidden layer feedforward perceptron
   - Toward deep learning
   - Multilayer feedforward perceptron and Backpropagation

Table of Contents

1 Optimization (deterministic, the origins)
2 Examples from Finance
- Implicitation
- Minimization

3 Learning procedures
- Abstract Learning
- Supervised Learning
- Unsupervised Learning (clustering)

4 Stochastic algorithms/Approximation
- From Robbins-Monro to Robbins-Siegmund
- Stochastic Gradient Descent (*SGD*) and pseudo-*SGD*

5 Examples revisited by *SFD*
- Numerical probability
- Learning (supervised and unsupervised)

6 Application to Neural Networks and deep learning
- Linear neural network
- One hidden layer feedforward perceptron
- Toward deep learning
- Multilayer feedforward perceptron and Backpropagation

Implicitation: Implied Volatility

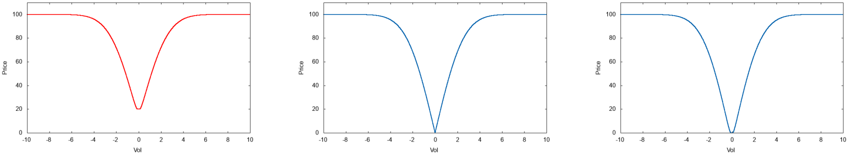
- Black-Scholes model: traded asset $X_t = x_0 e^{(r-\frac{\sigma^2}{2})t+\sigma W_t}$, x_0, volatility $\sigma > 0$, interest rate r, W standard Brownian motion.
- Call payoff $(X_T - K)_+ = \max(X_T - K, 0)$ with strike price K and maturity T.
- Mark-to-Market quoted price: $\mathrm{Call}_{M2Mkt} \in (0, x_0)$.
- Black-Scholes price at time 0

$$
\begin{aligned}
\mathrm{Call}_{BS}(x_0, K, r, \sigma, T) &= e^{-rT}\mathbb{E}\,(X_T - K)_+ \\
&= x_0\Phi_0(d_1) - Ke^{-rT}\Phi_0(d_2) \\
d_1 = \frac{\log(\frac{x_0}{K}) + (r + \frac{\sigma^2}{2})T}{\sigma\sqrt{T}}, &\quad d_2 = d_1 - \sigma\sqrt{T}.
\end{aligned}
$$

- Implicitation of the volatility: solve in σ the inverse problem

$$
\mathrm{Call}_{BS}(\ldots, \sigma, \ldots) - \mathrm{Call}_{M2Mkt} = 0.
$$

- Graphs of $\sigma \mapsto Call_{BS}(\sigma)$, $\sigma \in \mathbb{R}$: In-, At- and Out- the money.



- The function is even in σ and the equation has two opposite solutions.
- As $\sigma < 0$ is meaningless, one considers on the whole real line $\mathbb{R}$,

$$\sigma \longmapsto Call_{BS}(\sigma^+)$$

where $\sigma^+ = \max(\sigma, 0)$.

- It becomes a non-decreasing function.

- Algo_1:

$$\sigma_{n+1} = \sigma_n - \gamma_{n+1} \underbrace{\left(\mathrm{Call}_{BS}\big(x_0, K, r, {\sigma_n}^+, T\big) - \mathrm{Call}_{M2Mkt}\right)}_{=:h(\sigma_n)}, \; \sigma_0 > 0$$

with $\gamma_n = \gamma > 0$ or decreasing assumption.

- Algo_2 (Newton's zero search)

    - The Vega:

$$\mathrm{Vega}_{BS}(\sigma) = \frac{\partial}{\partial \sigma}\mathrm{Call}_{BS}(\sigma) = x_0 \,\mathrm{sign}(\sigma)\sqrt{T}\,\frac{e^{-\frac{d_1(\sigma)^2}{2}}}{\sqrt{2\pi}}$$

    - Implicit volatility search reads (works as long as $\sigma_n > 0\dots$):

$$\sigma_{n+1} = \sigma_n - \underbrace{\frac{1}{\mathrm{Vega}_{BS}(\sigma_n)}}_{=h'(\sigma_n)} \Big(\underbrace{\mathrm{Call}_{BS}\big(x_0, K, r, \sigma_n, T\big) - \mathrm{Call}_{M2Mkt}}_{=:h(\sigma_n)}\Big), \; \sigma_0 > 0.$$

[This is the actual algorithm with a "good choice" of σ_0 avoiding the negative side and ensuring a fast convergence ([1]).]

[1] S. Manaster, G. Koehler (1982). The calculation of Implied Variance from the Black–Scholes Model: A Note, *The Journal of Finance*, **37**(1):227–230

Implicitation: Implied Correlation I

- 2-dim (correlated) Black-Scholes model:

$$X_t^i = x_0^i e^{(r-\frac{\sigma_i^2}{2})t+\sigma_i W_t^i},\ x_0^i,\ \sigma_i > 0, i = 1, 2$$

  with $\langle W^1, W^2\rangle_t = \rho t$.

- Best-of-Call Payoff:

$$\big(\max(X_T^1, X_T^2) - K\big)_+$$

- Premium at time 0

$$\text{Best-of-Call}_{BS}(\dots, \rho, \dots) = e^{-rT}\mathbb{E}\,\big(\max(X_T^1, X_T^2) - K\big)_+.$$

- Organized markets on such options are market of the correlation ρ.
- The volatilities σ_i, $i = 1, 2$, are known from vanilla option markets on X^1 and X^2.

$$\boxed{\text{How to "extract" the correlation } \rho?}$$

- Deterministic algo(s):

$$\rho_{n+1} = \rho_n - \gamma_{n+1}\big(\underbrace{\text{Best-of-Call}_{BS}(\rho_n) - \text{Best-of-Call}_{M2Mkt}}_{=:h(\rho_n)}\big).$$

or the Levenberg-Marquard variant of Newton's zero search algorithm

$$\rho_{n+1} = \rho_n - \frac{\text{Best-of-Call}_{BS}(\rho_n) - \text{Best-of-Call}_{M2Mkt}}{\partial_\rho \text{Best-of-Call}_{BS}(\rho_n) + \lambda_n}.$$

  - Except that we have no (simple) closed form for the B-S price and its ρ-derivative.
  - The correlation $\rho \in [-1, 1]$. Projections are possible but. . . .

- What to do?

Table of Contents

1. Optimization (deterministic, the origins)
2. Examples from Finance
   - Implicitation
   - Minimization
3. Learning procedures
   - Abstract Learning
   - Supervised Learning
   - Unsupervised Learning (clustering)
4. Stochastic algorithms/Approximation
   - From Robbins-Monro to Robbins-Siegmund
   - Stochastic Gradient Descent (*SGD*) and pseudo-*SGD*
5. Examples revisited by *SFD*
   - Numerical probability
   - Learning (supervised and unsupervised)
6. Application to Neural Networks and deep learning
   - Linear neural network
   - One hidden layer feedforward perceptron
   - Toward deep learning
   - Multilayer feedforward perceptron and Backpropagation

- Let $X = \varphi(Z)$, $Z : (\Omega, \mathcal{A}, \mathbb{P}) \to \mathbb{R}^q$ be an integrable random variable representative of a loss and let $\alpha \in (0,1)$, $\alpha \simeq 1$.

$$\text{Value-at-Risk}_\alpha(X) = \alpha\text{-quantile} = \inf\big\{\xi : \mathbb{P}(X \le \xi) \ge \alpha\big\}.$$

- For simplicity, assume X has a density $f_X > 0$ on $\mathbb{R}$. Then $\xi_\alpha = \mathrm{VaR}_\alpha(X)$ is the unique solution to

$$\mathbb{P}\big(X \le \xi_\alpha\big) = \alpha \Longleftrightarrow \mathbb{P}\big(X > \xi_\alpha\big) = 1 - \alpha.$$

- The conditional Value-at-Risk is defined by

$$\mathrm{CVaR}_\alpha(X) = \mathbb{E}\big(X \,|\, X \ge \mathrm{VaR}_\alpha(X)\big).$$

- Rockafellar-Uryasev Potential ([2]):

$$V(\xi) = \xi + \frac{1}{1-\alpha}\mathbb{E}\,(X - \xi)_+, \quad \xi \in \mathbb{R}.$$

[2] R.T. Rockafellar, S. Uryasev (2000). Optimization of Conditional Value-At-Risk, *The Journal of Risk*, **2**(3):21–41. www.ise.ufl.edu/uryasev.

- The function V is convex and $\lim_{|\xi|\to+\infty} V(\xi) = +\infty$ since

$$V(\xi) \geq \xi \quad \text{so that} \quad \lim_{\xi\to+\infty} V(\xi) = +\infty$$

and

$$\begin{aligned} V(\xi) &\geq \xi + \frac{1}{1-\alpha}\big(\mathbb{E}\,X - \xi\big)_+ \qquad \text{by Jensen's inequality} \\ &\geq \xi + \frac{1}{1-\alpha}(\mathbb{E}\,X - \xi) \\ &= -\frac{\alpha}{1-\alpha}\xi + \frac{1}{1-\alpha}\mathbb{E}\,X \to +\infty \qquad \text{as} \quad \xi \to -\infty. \end{aligned}$$

- By exchanging differentiation and $\mathbb{E}$, we get

$$V'(\xi) = 1 - \frac{1}{1-\alpha}\mathbb{P}(X > \xi).$$

- $V'(\xi) = 0$ iff $\mathbb{P}(X > \xi) = 1 - \alpha$ iff $\xi = \xi_\alpha$.
- Moreover

$$V(\xi_\alpha) = \frac{\xi_\alpha \mathbb{P}(X > \xi_\alpha) + \mathbb{E}\,(X - \xi_\alpha)_+}{\mathbb{P}(X > \xi_\alpha)} = \frac{\mathbb{E}\,X\mathbf{1}_{\{X > \xi_\alpha\}}}{\mathbb{P}(X \geq \xi_\alpha)}$$
$$= \mathbb{E}\big(X \,|\, X \geq \mathrm{VaR}_\alpha(X)\big) = \mathrm{CVaR}_\alpha(X).$$

- (GD) pour la $\mathrm{VaR}_\alpha(X)$: $h(\xi) = V'(\xi)$. Let $\xi_0 \in \mathbb{R}$,

$$\xi_{n+1} = \xi_n - \gamma_{n+1}\Big(1 - \frac{1}{1-\alpha}\big(1 - F_X(\xi_n)\big)\Big)$$
$$= \xi_n - \frac{\gamma_{n+1}}{1-\alpha}\big(F_X(\xi_n) - \alpha\big), \quad n \geq 0.$$

- Newton/Levenberg-Marquardt algo: $\xi_0 \in \mathbb{R}$,

$$\xi_{n+1} = \xi_n - \frac{F_X(\xi_n) - \alpha}{f_X(\xi_n) + \lambda_n(?)}, \quad n \geq 0.$$

- Why not ! But $X = \varphi(Z)$ (the whole portfolio of a CIB Bank!) $\Rightarrow q$ large and no closed form for the c.d.f. $F_X(\xi) = \mathbb{P}(X \leq \xi)$ of X.

Table of Contents

1. Optimization (deterministic, the origins)
2. Examples from Finance
   - Implicitation
   - Minimization
3. Learning procedures
   - Abstract Learning
   - Supervised Learning
   - Unsupervised Learning (clustering)
4. Stochastic algorithms/Approximation
   - From Robbins-Monro to Robbins-Siegmund
   - Stochastic Gradient Descent (*SGD*) and pseudo-*SGD*
5. Examples revisited by *SFD*
   - Numerical probability
   - Learning (supervised and unsupervised)
6. Application to Neural Networks and deep learning
   - Linear neural network
   - One hidden layer feedforward perceptron
   - Toward deep learning
   - Multilayer feedforward perceptron and Backpropagation

Table of Contents

1. Optimization (deterministic, the origins)
2. Examples from Finance
   - Implicitation
   - Minimization
3. Learning procedures
   - Abstract Learning
   - Supervised Learning
   - Unsupervised Learning (clustering)
4. Stochastic algorithms/Approximation
   - From Robbins-Monro to Robbins-Siegmund
   - Stochastic Gradient Descent (*SGD*) and pseudo-*SGD*
5. Examples revisited by *SFD*
   - Numerical probability
   - Learning (supervised and unsupervised)
6. Application to Neural Networks and deep learning
   - Linear neural network
   - One hidden layer feedforward perceptron
   - Toward deep learning
   - Multilayer feedforward perceptron and Backpropagation

Abstract Learning

- Huge dataset $(z_k)_{k=1:N}$ with of possibly high dimension d: $N \simeq 10^6$, even 10^9, and $d \simeq 10^3$.
  [Image, profile, text, ...]
- Set of parameters $\theta \in \Theta \subset \mathbb{R}^K$, K large (see later on).
- There exists a smooth local loss function/local predictor

$$v(\theta, z).$$

- Global loss function: $V(\theta) = \dfrac{1}{N} \sum_{k=1}^{N} v(\theta, z_k)$

  with gradient $\nabla V(\theta) = \dfrac{1}{N} \sum_{k=1}^{N} \nabla_\theta v(\theta, z_k)$.

- Solving the minimization problem

$$\min_{\theta\in\Theta} V(\theta).$$

- Suggests a (GD) i.e. $h = \nabla V$ [or others... if $\nabla^2_\theta v(\theta, z)$ exists]:

$$\begin{aligned}\theta_{n+1} &= \theta_n - \gamma_{n+1}\nabla V(\theta_n)\\ &= \theta_n - \frac{\gamma_{n+1}}{N}\sum_{k=1}^{N}\nabla_\theta v(\theta, z_k),\ n \geq 0,\end{aligned}$$

with the step sequence satisfying the (DS) assumption.