Conditional quantile sequantial estimation for stochastic codes

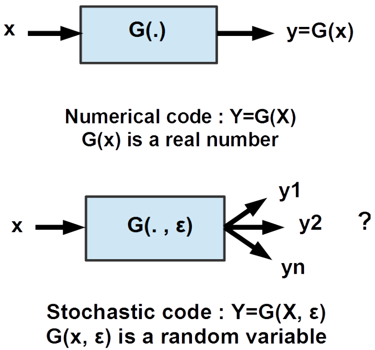
T. Labopin-Richard F. Gamboa A. Garivier

Institut de mathématiques de Toulouse

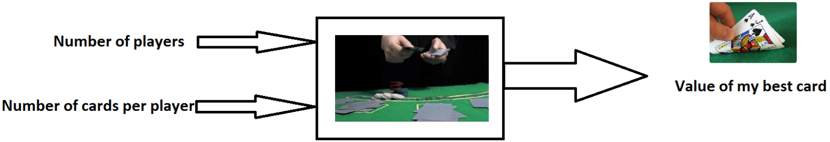
March 23, 2016

Table of contents

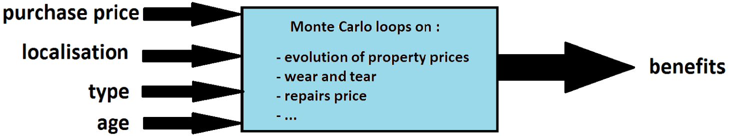
What is a stochastic code ?



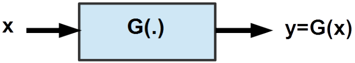
Example 1 :


Number of players
Number of cards per player
Value of my best card

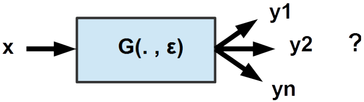
Example : property investment



What is a stochastic code ?

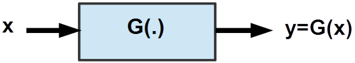


Numerical code : Y=G(X)
G(x) is a real number

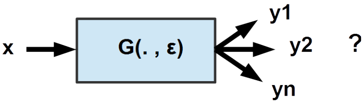


Stochastic code : Y=G(X, ε)
G(x, ε) is a random variable

What is a stochastic code ?



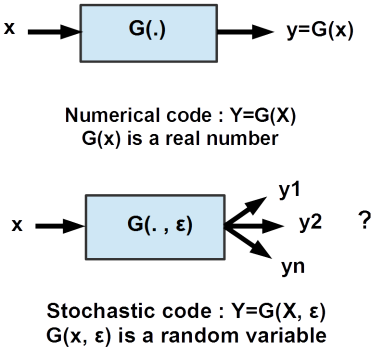
Numerical code : Y=G(X)
G(x) is a real number



Stochastic code : Y=G(X, ε)
G(x, ε) is a random variable

Notation : $X \in \mathbb{X} \subset \mathbb{R}^d$.

What is a stochastic code ?



Notation : $X \in \mathbb{X} \subset \mathbb{R}^d$.

Goal : Estimate the quantile of level $\alpha \in]0,1[$ of the law $\mathcal{L}\left(G(x,\epsilon)\right)$ using as few as possible calls to the code.

To estimate the quantile of a law Z, one can

1) Build a sample $(Z_1, \ldots, Z_n)$ of Z.
2) Use a quantile estimator as :
   a) The empirical quantile $Z_{(\lfloor n\alpha \rfloor + 1)}$.
      $\Rightarrow$ consistant and normaly Gaussian.
   b) The Robbins-Monro stochastic algorithm

$$\begin{cases} \theta_0 \in \mathbb{R} \\ \theta_{n+1} = \theta_n - \dfrac{1}{n^\gamma} \left(\mathbf{1}_{Z_{n+1} \leq \theta_n} - \alpha \right). \end{cases}$$

$\Rightarrow$ consistant and normaly Gaussian if $\gamma \in]1/2, 1]$.

Application to our problem

We aim at estimate the quantile of the law $\mathcal{L}\left(G(x,\epsilon)\right)$, we could then

1) Provide several times the same input x to the stochastic code, to build a sample of the target law.
2) Use one of the previous estimator.

Problem

We aim at estimating the conditional quantile for every input x. If $\mathbb{X}$ is uncountable or if each call to the code is expensive, the is not a solution.

Type of strategy

1) A budget N of calls to the code is fixed.
2) We sample $(X_1, \ldots, X_N)$ from the input law X.
3) We observe the corresponding responses $(Y_1, \ldots, Y_N)$.
4) We apply an algorithm wich allows, for every x and using only the previous observations, to estimate the conditional quantile.

Algorithm

$$\begin{cases} \theta_0 & \in \mathbb{R} \\ \theta_{n+1} & = \theta_n - \frac{1}{n^\gamma}\left(\mathbf{1}_{Z_{n+1}\leq\theta_n} - \alpha\right) \end{cases}$$

Algorithm

$$\begin{cases} \theta_0 & \in \mathbb{R} \\ \theta_{n+1} & = \theta_n - \frac{1}{n^\gamma}\left(\mathbf{1}_{Y_{n+1}\leq\theta_n} - \alpha\right) \end{cases}$$

Algorithm

$$\begin{cases} \theta_0(x) \in \mathbb{R} \\ \theta_{n+1}(x) = \theta_n(x) - \dfrac{1}{n^\gamma}\left(\mathbf{1}_{Y_{n+1} \leq \theta_n(x)} - \alpha\right) \end{cases}$$

Algorithm

$$\begin{cases} \theta_0(x) \in \mathbb{R} \\ \theta_{n+1}(x) = \theta_n(x) - \dfrac{1}{n^\gamma}\left(\mathbf{1}_{Y_{n+1}\leq\theta_n(x)} - \alpha\right)\mathbf{1}_{X_{n+1}\in kNN_{n+1}(x)} \end{cases}$$

où $k_n = \lfloor n^\beta \rfloor$.

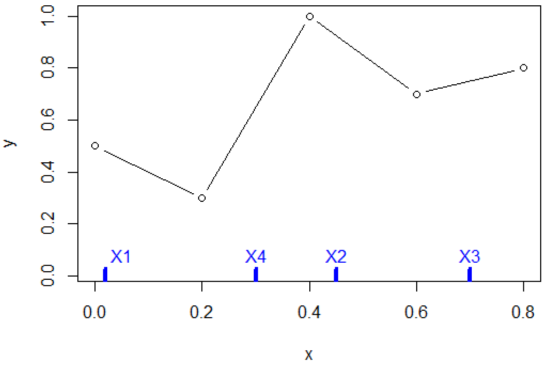
Algorithm

$$\begin{cases} \theta_0(x) \in \mathbb{R} \\ \theta_{n+1}(x) = \theta_n(x) - \dfrac{1}{n^\gamma}\left(\mathbf{1}_{Y_{n+1} \leq \theta_n(x)} - \alpha\right)\mathbf{1}_{X_{n+1} \in kNN_{n+1}(x)} \end{cases}$$

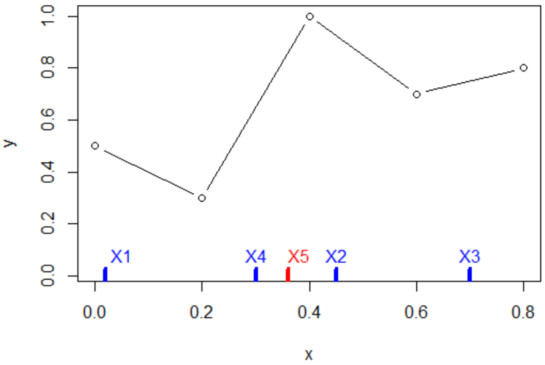
où $k_n = \lfloor n^\beta \rfloor$.

- For which parameters (γ, β) is the algorithm convergent ?
  $\Rightarrow$ Compromise needed.
- Could we prove some non-asymptotic results ?
- Are there optimal parameters ?

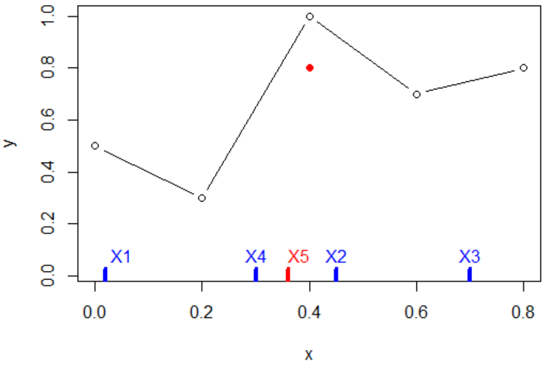
Example



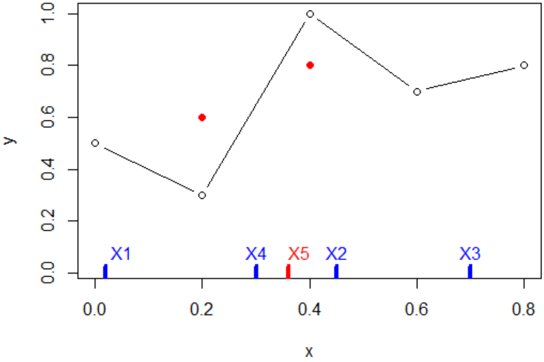
Example



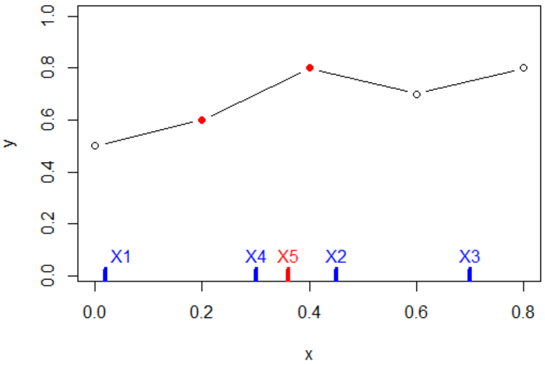
Example



Example



Example



Continuity assumption

Notations :

- $\mathcal{B}_x$ the set of the balls of $\mathbb{R}^d$ centered in x. $B \in \mathcal{B}_X$ has a radius r_B.
- For $B \in \mathcal{B}_x$, F_Y^B is the cumulative distribution function of the law $\mathcal{L}(g(X,\epsilon)|X \in B)$.
- F_{Y^x} is the cumulative distribution function of the law $\mathcal{L}(g(x,\epsilon))$.

Assumption A1 For all $x \in \mathrm{Supp}(X)$, there exists a constant $M(x)$ such that

$$\forall B \in \mathcal{B}_x,\ \forall t \in \mathbb{R},\ |F_{Y^B}(t) - F_{Y^x}(t)| \leq M(x) r_B\ .$$

Technical asumptions

Assumption A2 The input law X has a density which is lower bounded by a constant $C_{inputs} > 0$.

$\Rightarrow$ *Useful to deal with* $\mathbb{E}(||X - x||_{(k_n,n)})$ *or* $\mathbb{P}(X \in kNN_n(x))$.

Technical asumptions

Assumption A2 The input law X has a density which is lower bounded by a constant $C_{inputs} > 0$.

$\Rightarrow$ *Useful to deal with* $\mathbb{E}(||X - x||_{(k_n,n)})$ *or* $\mathbb{P}(X \in kNN_n(x))$.

Assumption A3 The code function g is at value in the compact set $[L_Y, U_Y]$.

$$\Rightarrow \ \forall x,\ \theta_n(x) \text{ is almost-surely bounded.}$$

Technical asumptions

Assumption A2 The input law X has a density which is lower bounded by a constant $C_{inputs} > 0$.

$\Rightarrow$ *Useful to deal with* $\mathbb{E}(||X - x||_{(k_n,n)})$ *or* $\mathbb{P}(X \in kNN_n(x))$.

Assumption A3 The code function g is at value in the compact set $[L_Y, U_Y]$.

$$\Rightarrow \ \forall x, \ \theta_n(x) \text{ is almost-surely bounded.}$$

Assumption A4 For every $x \in \text{Supp}(X)$, the law of $g(x, \epsilon)$ has a density which is lower bounded by a constant $C_g(x) > 0$.

$\Rightarrow$ *There exists a constant* $C_2(x, \alpha)$ *such that :*

$$\forall \theta_n(x), \ [F_{Y^x}(\theta_n(x)) - F_{Y^x}(\theta^*(x))]\,[\theta_n(x) - \theta^*(x)] \geq C_2(x, \alpha)\,[\theta_n(x) - \theta^*(x)]^2 .$$

Results

Theorem : almost-sure convergence

Let x be a fixed input. Under assumptions **A1** and **A2**, the algorithm in x is almost surely convergent whenever $\frac{1}{2} < \gamma < \beta < 1$.

Results

Theorem : almost-sure convergence

Let x be a fixed input. Under assumptions **A1** and **A2**, the algorithm in x is almost surely convergent whenever $\frac{1}{2} < \gamma < \beta < 1$.

Comments on parameters :

- $1/2 < \gamma \leq 1 \Rightarrow$ classical assumption for Robbins-Monro algorithm.

Results

Theorem : almost-sure convergence

Let x be a fixed input. Under assumptions **A1** and **A2**, the algorithm in x is almost surely convergent whenever $\frac{1}{2} < \gamma < \beta < 1$.

Comments on parameters :

- $1/2 < \gamma \leq 1 \Rightarrow$ classical assumption for Robbins-Monro algorithm.
- $0 < \beta \Rightarrow$ the number a neighbors goes to $+\infty$ and then, $||X - x||_{(k_n,n)} \to 0$.

Results

Theorem : almost-sure convergence

Let x be a fixed input. Under assumptions **A1** and **A2**, the algorithm in x is almost surely convergent whenever $\frac{1}{2} < \gamma < \beta < 1$.

Comments on parameters :

- $1/2 < \gamma \leq 1 \Rightarrow$ classical assumption for Robbins-Monro algorithm.
- $0 < \beta \Rightarrow$ the number a neighbors goes to $+\infty$ and then, $||X - x||_{(k_n,n)} \to 0$.
- $\beta < 1 \Rightarrow$ technical assumption.

Results

Theorem : almost-sure convergence

Let x be a fixed input. Under assumptions **A1** and **A2**, the algorithm in x is almost surely convergent whenever $\frac{1}{2} < \gamma < \beta < 1$.

Comments on parameters :

- $1/2 < \gamma \leq 1 \Rightarrow$ classical assumption for Robbins-Monro algorithm.
- $0 < \beta \Rightarrow$ the number a neighbors goes to $+\infty$ and then, $||X - x||_{(k_n,n)} \to 0$.
- $\beta < 1 \Rightarrow$ technical assumption.
- $\gamma < \beta \Rightarrow$ *effective learning rate*.

Classical algorithm :

- An update at each step
- $\sum_n \gamma_n = +\infty \Rightarrow \gamma \in \left]\frac{1}{2}, 1\right]$.

Our algrithm :

- An update every $\frac{k_n}{n} \sim n^{\beta-1}$ steps
- At step n : $N = \sum_{k \leq n} k^{\beta-1} \sim n^{\beta}$ updates
- At time $t = n^{\frac{1}{\beta}}$: $N = n$ updates
- *Effective* learning rate : $\gamma_{k_n} = \frac{1}{\left(n^{\frac{1}{\beta}}\right)^{\gamma}}$.
- $\sum_n \gamma_{k_n} = +\infty \Rightarrow \gamma < \beta$.

Rate of convergence of the MSE

Theorem

Under hypothesis ***A1***, ***A2***, ***A3*** *and* ***A4***, *the MSE* $a_n(x)$ *satisfies :*
$\forall(\gamma,\beta,\epsilon)$ *such that* $0<\gamma\leq\beta<1$ *and* $1>\epsilon>1-\beta$, $\forall n\geq N_0$,

$$a_n(x)\leq \exp\left(-2C_2(x,\alpha)\sum_{j=N_0+1}^{n} j^{-\epsilon-\gamma}\right)C_1$$

$$+\sum_{k=N_0+1}^{n}\exp\left(-2C_2(x,\alpha)\sum_{j=k+1}^{n} j^{-\epsilon-\gamma}\right)d_k + C_1\exp\left(-\frac{3n^{1-\epsilon}}{8}\right)$$

where

$$d_n = C_1\exp\left(-\frac{3n^{1-\epsilon}}{8}\right)+2\sqrt{C_1}M(x)C_3(d)\gamma_n\left(\frac{k_n}{n}\right)^{\frac{1}{d}+1}+\gamma_n^2\frac{k_n}{n}.$$

Compromise between the two errors

- The bias error gives the term
$$\exp\left(-2C_2(x,\alpha)(x)\sum_{k=N_0+1}^{n}\frac{1}{k^{\epsilon+\gamma}}\right).$$
This term decreases to 0 if and only if $\gamma+\epsilon<1$ which implies $\beta>\gamma$. Then β has to be chosen not too small.

- The online learning error gives the term
$$\gamma_n\left(k_n/n\right)^{1/d+1}=n^{(1-\beta)(1+1/d)+\gamma}.$$
We then need that β is as small as possible compared to 1. Then β has to be chosen not too big.

Corollary : optimal parameters

Under the same assumptions than in Theorem 1, the mean square error decreases faster when parameters are $\gamma = \frac{1}{1+d}$ and $\beta = \gamma + \eta_\beta$ where $\eta_\beta > 0$ is as small as possible. Moreover, with these parameters, there exists a constant $C_9(x, \alpha, d)$ such that $\forall n \geq N_4(x, \alpha, d)$,

$$a_n(x) \leq \frac{C_9(x, \alpha, d)}{n^{\frac{1}{1+d} - \eta}}$$

where $\eta = \frac{\eta_\epsilon}{2} + \eta_\beta$ and $\eta_\epsilon = 1 - \beta - \epsilon$.

Dimension 1 - almost-sure convergence

Two models with $X \sim \mathcal{U}([-1,1])$, $\epsilon \sim \mathcal{U}([-0.5,0.5])$ and $x = 0$:

$$g(X,\epsilon) = X^2 + \epsilon \text{ et } g(X,\epsilon) = |X| + \epsilon$$

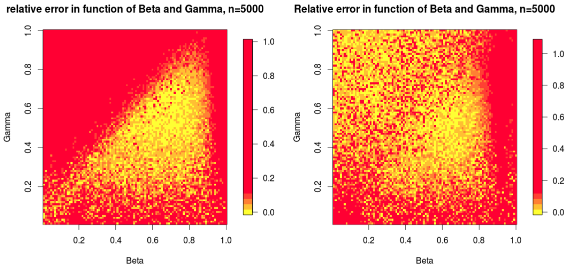


FIGURE – Almost-sure convergence in function of β et γ.

Mean Square Error

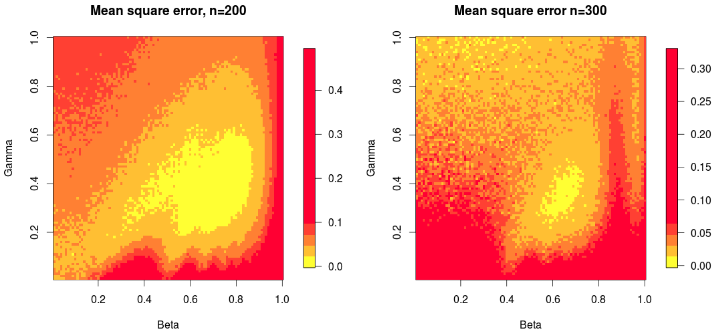


FIGURE – Convergence of the mean square error in function of β et γ.

Dimensions 2 and 3

Two models with $g(X, \epsilon) = ||X||^2 + \epsilon$ for $X \sim \mathcal{U}([-1, 1]^d)$, $\epsilon \sim \mathcal{U}([-0.5, 0.5])$ and $x = 0_{\mathbb{R}^d}$.

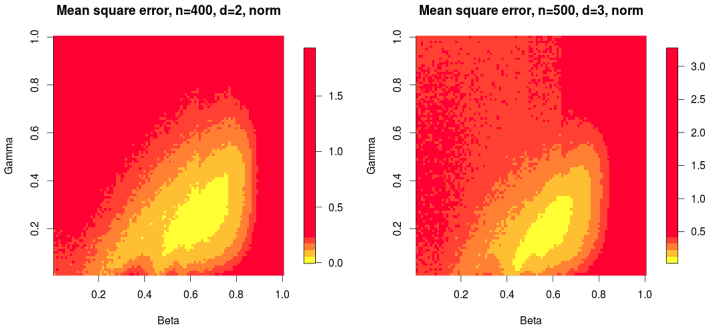


FIGURE – Mean Square Error in function of β and γ.

Conclusion et perspectives

Conclusion :

- We introduced an algorithm to estimate the conditional quantile of the output law of a stochastic code.
- We give the best parameters to tune the algorithme (to reach the best rate of convergence of the MSE).
- Numerical simulations show that our algorithm is powerfull to solve the problem.

Perspectives :

- What is happening if we relax the compact support assumptions ?
- Could we find lower bound for the Mean Square Error ?
- Apply this algorithm to real data.

Thanks for your attention.