Algorithmes de Newton en streaming

A. Godichon-Baggioni

Algorithmes de gradient en streaming

Données en streaming

- Au temps $t+1$: $X_{t+1,1}, \ldots, X_{t+1,n_{t+1}}$ données i.i.d.

- Mise à jour des etimateurs :

$$m_{t+1} = m_t - \gamma_{t+1} \frac{1}{n_{t+1}} \sum_{i=1}^{n_{t+1}} \nabla g \left(X_{t+1,i}, m_t \right)$$

- Conditions sur les pas :

$$\sum_{t \geq 1} \gamma_t = +\infty \qquad \text{et} \qquad \sum_{t \geq 1} \frac{\gamma_t^2}{n_t} < +\infty$$

$\Longrightarrow m_t$ converge p.s vers m.

Version moyennée

- Estimateurs moyennés :

$$\overline{m}_t = \frac{1}{N_t} \sum_{k=0}^{t-1} n_{k+1} m_k$$

où $N_t = \sum_{k=1}^{t} n_k$.

- Mise à jour récursive :

$$\overline{m}_{t+1} = \frac{N_t}{N_{t+1}} \overline{m}_t + \frac{n_{t+1}}{N_{t+1}} m_t$$

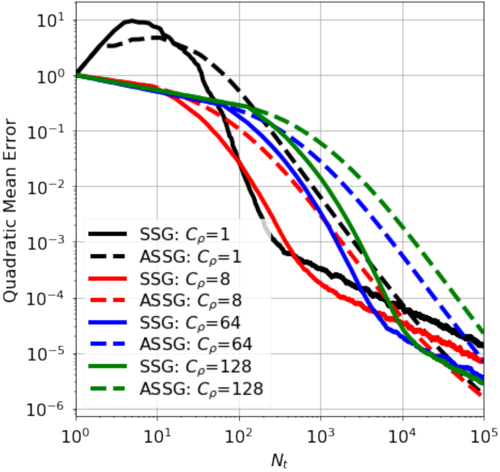
- Si $n_t = t$, on peut prendre

$$\overline{m}_{t+1} = \frac{t}{t+1} \overline{m}_t + \frac{1}{t+1} m_{t+1}.$$

- Convergence : sous certaines hypothèses

$$\sqrt{N_t} \left(\overline{m}_t - m \right) \xrightarrow[t \to +\infty]{\mathcal{L}} \mathcal{N}\left(0, H^{-1} \Sigma H^{-1}\right).$$

Exemple



Quadratic Mean Error
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
N_t
SSG: C_ρ=1
ASSG: C_ρ=1
SSG: C_ρ=8
ASSG: C_ρ=8
SSG: C_ρ=64
ASSG: C_ρ=64
SSG: C_ρ=128
ASSG: C_ρ=128

Algorithmes de Newton en streaming

Algorithmes de Newton en streaming

- On fixe $n_t = n$.
- On suppose

$$\nabla^2 G(m) = \mathbb{E}\left[\alpha\left(X,m\right)\Phi\left(X,m\right)\Phi\left(X,m\right)^T\right]$$

- Algorithme de Newton en streaming :

$$m_{t+1} = m_t - \frac{1}{t+1}\overline{H}_t^{-1}\nabla g\left(\mathbf{X}_{t+1}, m_t\right)$$

avec $\nabla g\left(\mathbf{X}_{t+1}, m_t\right) = \frac{1}{n}\sum_{i=1}^{n}\nabla g\left(X_{t+1,i}, m_t\right)$.

- Estimateur de la Hessienne :

$$\overline{H}_t = \frac{1}{N_t+1}\left(H_0 + \sum_{k=1}^{t}\sum_{i=1}^{n}\alpha_{k,i}\Phi_{k,i}\Phi_{k,i}^T\right)$$

avec $\alpha_{k,i} = \alpha\left(X_{k,i}, m_{k-1}\right)$ et $\Phi_{k,i} = \Phi\left(X_{k,i}, m_{k-1}\right)$.

Mise à jour de $\overline{H}_t^{-1}$

- On note $H_t = (N_t + 1)\,\overline{H}_t$ et

$$H_{t+1} = H_t + \sum_{i=1}^{n} \alpha_{t+1,i}\Phi_{t+1,i}\Phi_{t+1,i}^T$$

- Mise à jour de H_t^{-1}.

$$H_{t,i} = H_{t,i-1} + \alpha_{t+1,i}\Phi_{t+1,i}\Phi_{t+1,i}^T$$

  avec $H_{t,0} = H_t$ et $H_{t,n} = H_{t+1}$.

- Formule de Riccati, pour tout $i = 1, \ldots, n$,

$$H_{t,i}^{-1} = H_{t,i}^{-1} - \alpha_{t+1,i}\frac{1}{\left(1 + \alpha_{t+1,i}\Phi_{t+1,i}^T H_{t,i-1}^{-1}\Phi_{t+1,i}\right)} H_{t,i-1}^{-1}\Phi_{t+1,i}\Phi_{t+1,i}^T H_{t,i-1}^{-1}$$

RÉSULTATS DE CONVERGENCE ET TEMPS DE CALCULS

Sous certaines hypothèses, on a

$$\sqrt{N_t}\,(m_t - m) \xrightarrow[t\to+\infty]{\mathcal{L}} \mathcal{N}\left(0, H^{-1}\Sigma H^{-1}\right).$$

Temps de calculs : on a N_t/n itérations avec

- Mise à jour du gradient : $O(nd)$ opérations
- Mise à jour de la Hessienne : $O(nd^2)$ opérations
- Mise à jours de m_t : $O(d^2)$ opérations.

Cout total :

$$\underbrace{O\left(N_t d\right)}_{\text{mise à jour du gradient}} + \underbrace{O\left(N_t d^2\right)}_{\text{mise à jour de la Hessienne}} + \underbrace{O\left(\frac{N_t}{n} d^2\right)}_{\text{mise à jour de } m_t}$$

Algorithme de Newton avec $O(N_t d)$ opérations

- Idée : prendre $n = d$.
- Temps de calculs pour les mises à jour de m_t : $O(N_t d)$.
- Temps de calculs pour les mises à jour des gradients $O(N_t d)$.
- Objectif : réduire le temps de calcul pour la mise à jour de la Hessienne.

Nouvel estimateur de la Hessienne

- Nouvel estimateur :

$$H_{t+1} = H_t + \sum_{i=1}^{n} Z_{t+1,i}\alpha_{t+1,i}\Phi_{t+1,i}\Phi_{t+1,i}^T$$

avec $Z_{t+1,i} \sim \mathcal{B}(p)$ et $p \in (0,1)$ et $\overline{H}_t = \frac{1}{1+\sum_{k=1}^{t}\sum_{i=1}^{n} Z_{t,i}} H_t$.

- Mise à jour de H_t^{-1} : si $Z_{t+1,i} = 1$,

$$H_{t,i} = H_{t,i-1} + \alpha_{t+1,i}\Phi_{t+1,i}\Phi_{t+1,i}^T$$

avec $H_{t,0} = H_t$ et $H_{t,n} = H_{t+1}$.

- Formule de Riccati, pour tout $i = 1, \ldots, n$, si $Z_{t+1,i} = 1$

$$H_{t,i}^{-1} = H_{t,i}^{-1} - \alpha_{t+1,i}\frac{1}{\left(1 + \alpha_{t+1,i}\Phi_{t+1,i}^T H_{t,i-1}^{-1}\Phi_{t+1,i}\right)} H_{t,i-1}^{-1}\Phi_{t+1,i}\Phi_{t+1,i}^T H_{t,i-1}^{-1}$$

Temps de calcul et convergence

- Temps de calcul pour les mises à jours de la Hessienne : $O\left(N_t d^2 p\right)$ (en moyenne)
- Prendre $p = d^{-1}$.
- Temps de calcul total : $O\left(N_t d\right)$ (en moyenne).
- Convergence :

$$\sqrt{N_t}\left(m_t - m\right) \xrightarrow[t \to +\infty]{\mathcal{L}} \mathcal{N}\left(0, H^{-1}\Sigma H^{-1}\right)$$

- Pondération :

$$H_{t+1} = H_t + \log(t+1)^w \sum_{i=1}^{n} Z_{t+1,i}\alpha_{t+1,i}\Phi_{t+1,i}\Phi_{t+1,i}^T$$

et $\overline{H}_t = \frac{1}{1+\sum_{k=0}^{t}\log(k+1)^w \sum_{i=1}^{n} Z_{t,i}} H_t$.

Algorithmes de Newton moyennés pondérés

Algorithmes de Newton moyenné

- L'algorithme

$$m_{t+1} = m_t - \gamma_{t+1} \overline{H}_t^{-1} \nabla g \left(X_{t+1}, m_t \right)$$
$$\overline{m}_{t+1} = m_t + \frac{\ln(t+1)^w}{\sum_{k=0}^{t} \log(k+1)^w} \left(m_{t+1} - \overline{m}_t \right)$$

avec $\gamma_{t+1} = c_\gamma t^{-\alpha}$ et $c_\gamma > 0$ et $\alpha \in (1/2, 1)$.

- Attention ! Dans $\overline{H}_t$, on peut utiliser m_t ou $\overline{m}_t$.
- Convergence :

$$\sqrt{t} \left(\overline{m}_t - m \right) \xrightarrow[t \to +\infty]{\mathcal{L}} \mathcal{N} \left(0, H^{-1} \Sigma H^{-1} \right).$$

MODÈLE LINÉAIRE

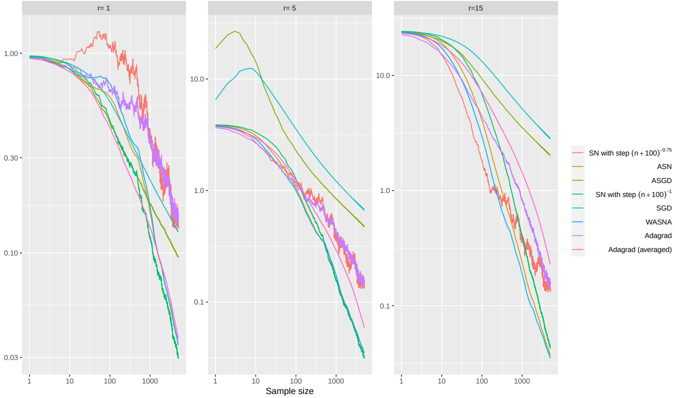


FIGURE – Evolution de l'erreur quadratique moyenne pour différentes initialisations : $m_0 = m + rU$, où U suit une loi uniforme sur la sphère $\mathbb{R}^d$ avec $r = 1$ (à gauche), $r = 2$ (au milieu) et $r = 5$ (à droite).

Algorithmes de Newton moyenné en streaming

- Algorithme de Newton moyenné en streaming :

$$
\begin{aligned}
m_{t+1} &= m_t - \frac{1}{t+1}\overline{H}_t^{-1}\nabla g\left(\mathbf{X}_{t+1}, m_t\right) \\
\overline{m}_{t+1} &= m_t + \frac{\ln(t+1)^w}{\sum_{k=0}^{t}\log(k+1)^w}\left(m_{t+1} - \overline{m}_t\right)
\end{aligned}
$$

avec $\nabla g\left(\mathbf{X}_{t+1}, m_t\right) = \frac{1}{n}\sum_{i=1}^{n}\nabla g\left(X_{t+1,i}, m_t\right)$.

- Estimateur de la Hessienne :

$$
\overline{H}_t = \frac{1}{1+\sum_{k=1}^{t}\log(k+1)^{w'}\sum_{i=1}^{n}Z_{k,i}}\left(H_0 + \sum_{k=1}^{t}\log(k+1)^{w'}\sum_{i=1}^{n}Z_{k,i}\alpha_{k,i}\Phi_{k,i}\Phi_{k,i}^T\right)
$$

avec $\alpha_{k,i} = \alpha\left(X_{k,i}, m_{k-1}\right)$ et $\Phi_{k,i} = \Phi\left(X_{k,i}, m_{k-1}\right)$.

MODÈLE LINÉAIRE

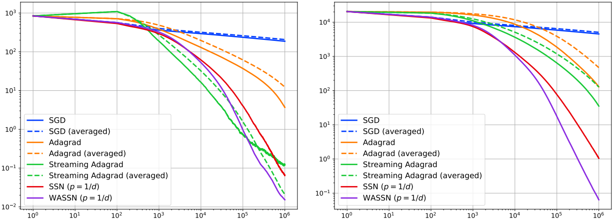


FIGURE – Evolution de l'erreur quadratique moyenne pour différentes initialisations : $m_0 = m + rU$, où U suit une loi uniforme sur la sphère $\mathbb{R}^d$ avec $r = 1$ (à gauche) et $r = 5$ (à droite).

MODÈLE LINÉAIRE

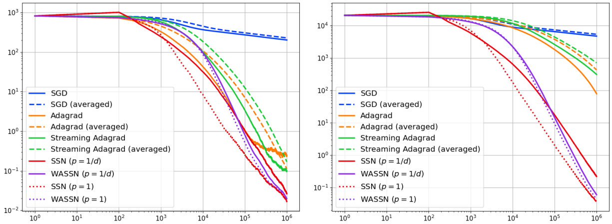


FIGURE – Evolution de l'erreur quadratique moyenne pour différentes initialisations : $m_0 = m + rU$, où U suit une loi uniforme sur la sphère $\mathbb{R}^d$ avec $r = 1$ (à gauche) et $r = 5$ (à droite).

Régression logistique

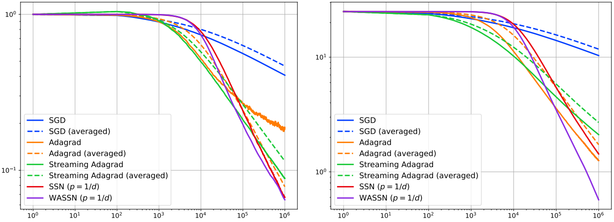


FIGURE – Evolution de l'erreur quadratique moyenne pour différentes initialisations : $m_0 = m + rU$, où U suit une loi uniforme sur la sphère $\mathbb{R}^d$ avec $r = 1$ (à gauche) et $r = 5$ (à droite).

Régression logistique

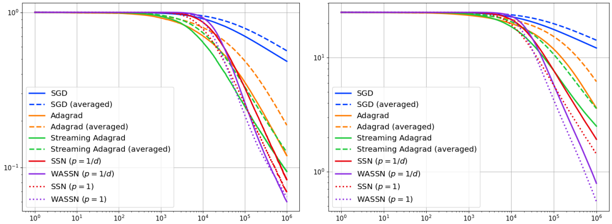


Figure – Evolution de l'erreur quadratique moyenne pour différentes initialisations : $m_0 = m + rU$, où U suit une loi uniforme sur la sphère $\mathbb{R}^d$ avec $r = 1$ (à gauche) et $r = 5$ (à droite).