INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN

Simon DELECOURT

Synthèse d'article

MINIMIZING FINITE SUMS WITH THE STOCHASTIC AVERAGE GRADIENT

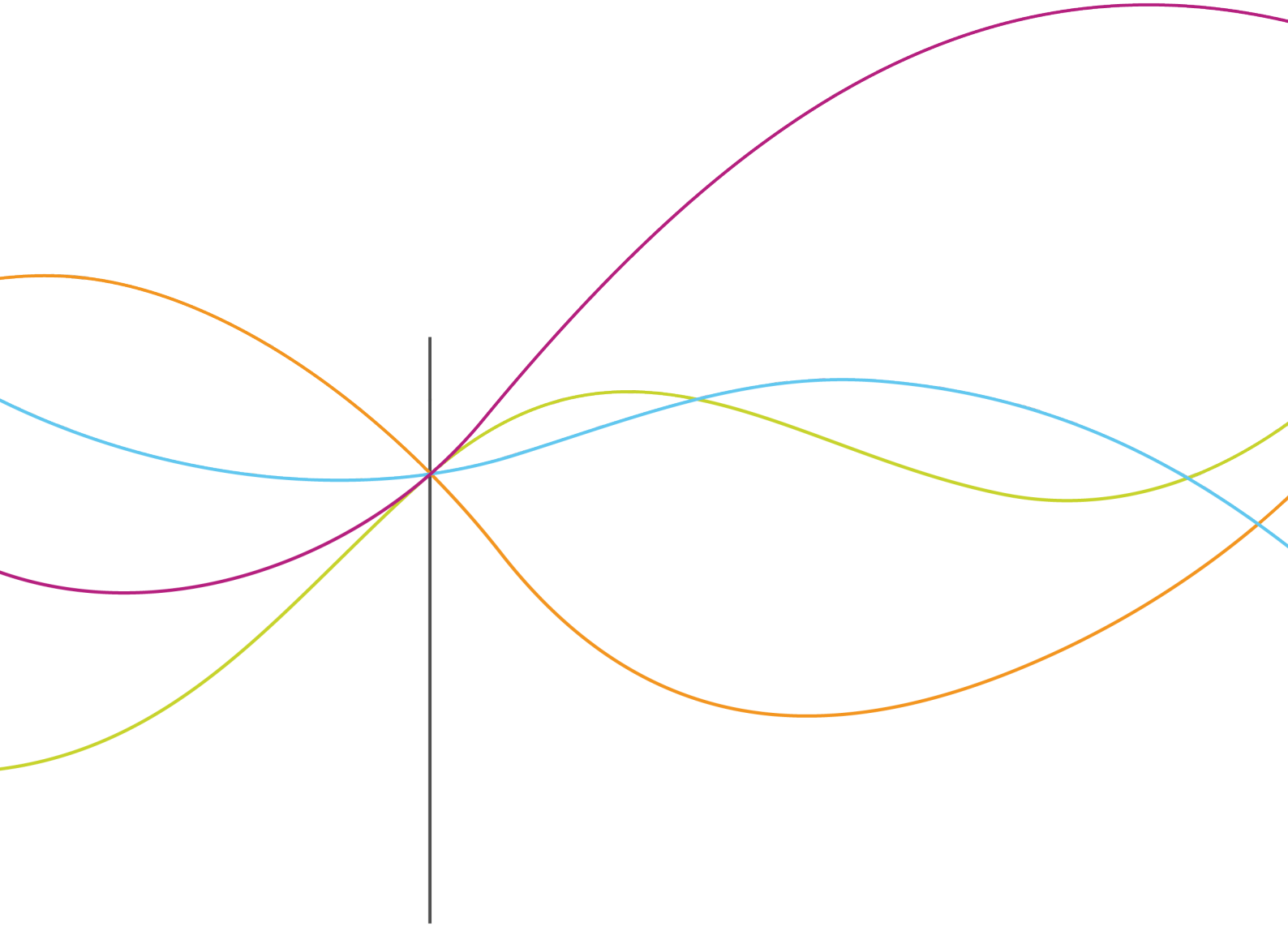

Table des matières

1 Introduction

L'article étudié est *"Minimizing Finite Sums with the Stochastic Average"* par Mark Schmidt, Nicolas Le Roux et Francis Bach et a été publié en 2017 [1]. L'article est organisé de la manière suivante :

- **Introduction** - Dans cette partie le problème est introduit. Les méthodes existantes pour résoudre ce problème sont présentées. Enfin leur méthode est présentée ainsi que son intérêt par rapport aux autres méthodes.
- **Related Works** - Les méthodes qui vont dans le même sens que celle proposé par les auteurs sont expliqués ainsi que leurs inconvénients.
- **Convergence Analysis** - La convergence dans leur méthode est détaillée
- **Implementation details** - L'algorithme général est présenté ainsi que différentes variantes afin d'améliorer les résultats dans certaines conditions.
- **Experimental Results** - L'algorithme proposé est appliqué sur différents *datasets* de *benchmark*. L'effet de la variation des différents hyper-paramètres est observé empiriquement.

2 Présentation du problème et méthode proposée

Le premier constat que font les auteurs est que beaucoup de problème d'optimisation, notamment avec la popularité du Machine Learning, nécessite de trouver le minimum d'une somme de fonction. Chaque terme de la somme représente le coût entre la valeur prédite par le modèle et la vraie donnée. Ainsi l'objectif est donc de résoudre le problème d'optimisation suivant :

$$\underset{x \in R^p}{\text{minimize}}\ g(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \tag{1}$$

Où chaque f_i est une fonction continue et convexe.

Pour résoudre ce type de problème, il existe la méthode de Full Gradient :

$$x^{k+1} = x^k - \frac{\alpha_k}{n} \sum_{i=1}^{n} f_i'(x^k) \tag{2}$$

Avec cette méthode à l'itération k, dans les conditions standard, g convexe et avec x^* la solution optimale on a : $g(x^k) - g(x^*) = O(1/k)$. Cependant lorsque l'on a beaucoup de données FG est très couteux, en effet le coût de chaque itération est proportionnel au nombre de données n.

Une autre méthode plus récente, *Stochastic Gradient* vise à résoudre le problème de minimisation en tirant aléatoirement à chaque itération un index i_k dans $1, ..., n$. On itère alors de la manière suivante :

$$x^{k+1} = x^k - \alpha_k f_{ik}'(x^k) \tag{3}$$

Avec cette méthode, dans certaines conditions et pour g convexe, on obtient un taux de convergence de $E[g(x^k)] - g(x^*) = O(1/\sqrt{(k)}$ ce qui moins bon que FG mais pour un coût très faible ce qui rend cette méthode applicable pour de plus grande base de données.

La contribution des auteurs est alors de proposer un nouvel algorithme, *Stochastic Average Gradient (SAG)* qui a le même coût à chaque itération que SG et un taux de convergence identique à FG. A l'itération k, x^k mis à jour selon l'équation suivante :

$$x^{k+1} = x^k - \frac{\alpha_k}{n} \sum_{i=1}^{n} y_i^k \tag{4}$$

$$y_i^k = \begin{cases} f_i'(x^k) \text{ si } i = i_k, \\ y_i^{k-1} \text{ sinon} \end{cases} \tag{5}$$

L'idée est de garder en mémoire les valeurs des gradients de l'itération précédente et de mettre à jour qu'une seule de ces valeurs.

3 Analyse de la convergence

Les auteurs prouvent alors le **théorème** suivant : Avec un pas constant de $\alpha_k = \frac{1}{16L}$ avec L constante de Lipschitz des f_i', les itérations de SAG vérifient pour $k \geq 1$ et g μ-fortement convexe :

$$E\left[g(x^k)\right] - g(x^*) \leq \left(1 - min\left\{\frac{\mu}{16L}, \frac{1}{8n}\right\}\right)^k C_0 \tag{6}$$

Le taux de convergence est donc en $O(\rho^k)$. Ils montrent également que dans le cas où g est juste convexe alors le taux de convergence est de O(n/k). Cela est donc moins bien que pour la méthode FG. Mais la comparaison est biaisée en effet chaque itération de SAG est n fois moins coûteuse que FG, de plus le pas choisi et l'initialisation x^0 ont leur importance. Pour mettre cela en perspective, voici un tableau récapitulatif qui compare 1 itération de FG contre n iterations de SAG (ce qui représente le même temps de calcul) pour un cas 1 bien conditionné et un cas 2 mal conditionné :

Algorithme	Step Size	Taux théorique	Taux Cas 1	Taux Cas 2
FG	$\frac{1}{n}$	$(1 + \frac{\mu}{L})^2$	0.9998	1.000
FG	$\frac{2}{\mu+L}$	$(1 + \frac{\mu}{L+\mu})^2$	0.9996	1.000
SAG (n iters)	$\frac{1}{16L}$	$(1 - min\{\frac{\mu}{16L}, \frac{1}{8n}\})^n$	0.8825	0.9938

On peut voir alors que SAG a une convergence plus rapide que FG.

4 Détails d'implémentation

Dans cette partie les auteurs montrent que leur méthode peut également s'appliquer dans différents cas particuliers et tirer partie de certaine particularité. Je ne vais aborder ici que quelques cas, veuillez vous référer au papier [1] pour la liste complète de ces cas.

4.1 *Line Search* quand L est inconnu

En général la constante de Lipschitz est inconnue. Les auteurs expliquent une façon de l'approximer par *Line Search*. Pour cela, il suffit de prendre une initialisation L^0 et de doubler cette valeur quand l'inégalité suivante n'est pas vérifiée :

$$f_{i_k}\left(x^k - \frac{1}{L^k} f_{i_k}'(x^k)\right) \leq f_{i_k}(x^k) - \frac{1}{2L^k}\|f_{i_k}'(x^k)\|^2 \tag{7}$$

En effet cette inégalité doit être vérifiée pour que L soit une constante de Lipschitz. Il est à remarquer que ce test est indépendant de n, tout comme la méthode SAG enfin d'en garder les avantages.

5 Résultats empiriques

Les taux théoriques de convergence indiquent qu'il est préférable d'utiliser une méthode SG lorsque qu'on ne peut se permettre d'itérer plus d'une fois sur les données, car il a trop de données. Tandis que si on peut itérer plus fois sur les données alors une méthode FG peut être utilisée. L'hypothèse des auteurs est que SAG serait plus avantageuse que les méthodes SG et FG lorsqu'on le se situe entre les 2 cas mentionnés ci-dessus.

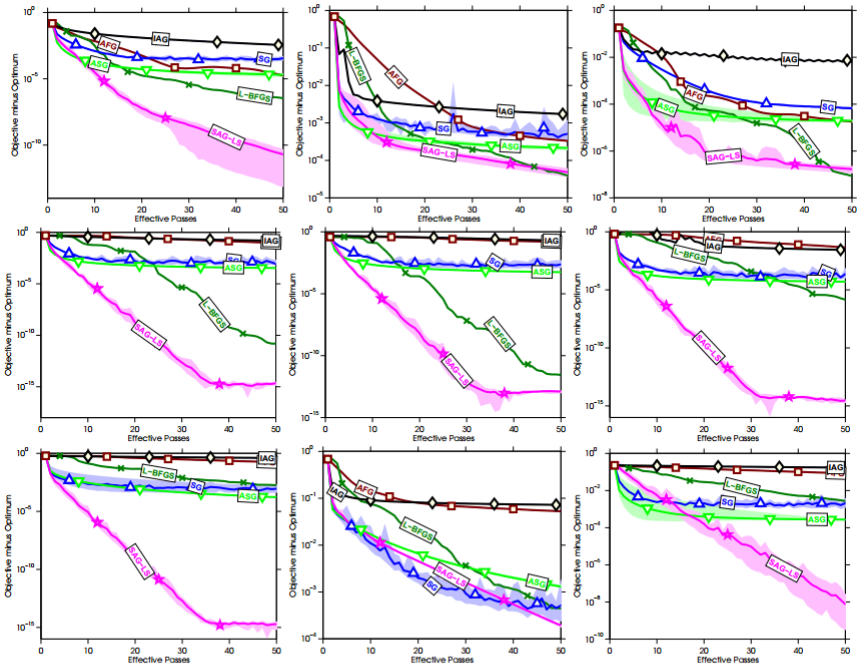

Figure 1: Comparison of different FG and SG optimization strategies. The top row gives results on the *quantum* (left), *protein* (center) and *covertype* (right) datasets. The middle row gives results on the *rcv1* (left), *news* (center) and *spam* (right) datasets. The bottom row gives results on the *rcv1Full* (left), *sido* (center), and *alpha* (right) datasets. This figure is best viewed in colour.

FIGURE 1 – Résultats obtenus dans le papier. Les différentes figures représentent différentes base de données. La courbe rose est SAG, les courbes marrons et vert foncées sont des méthodes FG, le reste étant des méthodes SG.

Ces courbes nous montrent que de manière générale SAG semblent avoir un taux de convergence empirique bien meilleur que les autres méthodes. Si on regarde de plus près les courbes on remarque qu'il y a 2 base de données où sont intérêt est mitigé par rapport aux autres méthodes. Ces 2 bases de données sont protein (145 751 points et 74 variables) et sido (12 678 et 4 932 variables). De plus SAG semble particulièrement bon pour alpha (500 000 points et 4 932 variables, news (19 996 points pour 1 355 191 variables) et spam (92 189 points pour 823 470 variables). Même s'il est difficile de confirmer l'hypothèse des auteurs, il semblerait que pour des grandes bases de données SAG est effectivement le meilleur choix.

6 Conclusion

Cet article est très complet et apporte plusieurs enseignements. Tout d'abord, l'intérêt de faire un bon état de l'art tout en posant les limites de chaque méthode. Puis à partir de ces constats, le papier apporte une nouvelle idée, simple, mais prouve de manière théorique son efficacité. Les auteurs montrent également que dans plusieurs cas particuliers leur méthode peut s'ajuster. Finalement, l'application de leur méthode pour des bases de données de benchmarks finit de prouver empiriquement que leur travail théorique est valide. Cependant comme le montre d'autres expériences qu'ils ont menés, le choix des hyper-paramètres dans les différentes comme le pas de descente ou la valeur de la constance de Lispchitz ont un impact non-négligeables sur les performances des méthodes et viennent nuancer leur résultat.

Références

[1] Minimizing finite sums with the stochastic average gradient Mark Schmidt, Nicolas Le Roux, Francis Bach, Math. Program. (2017) 162 : 83. https://doi.org/10.1007/s10107-016-1030-6