

Sur la complexité et la rapidité d'algorithmes pour la minimisation de la variation totale sous contraintes

Pierre WEISS¹, Laure BLANC-FÉRAUD¹, Gilles AUBERT²

¹ARIANA, projet commun CNRS/INRIA/UNSA
INRIA Sophia Antipolis, 2004, route des Lucioles
BP93, 06902, Sophia Antipolis Cedex, France

²Laboratoire J.A.Dieudonné, UMR CNRS 6621
Université de Nice Sophia-Antipolis, Parc Valrose
06108 Nice Cedex 2, France

pweiss@sophia.inria.fr, blancf@sophia.inria.fr, gaubert@unice.fr

Résumé – Dans ce travail nous proposons des solutions pour minimiser la variation totale sous des contraintes convexes. Les algorithmes proposés sont inspirés d'un résultat récent de Yurii Nesterov [4]. Suivant la régularité de la contrainte, nous proposons d'utiliser des techniques reposant soit sur la dualité, soit sur une régularisation différentiable de la variation totale. Les méthodes proposées sont rapides et leur complexité est analysée. Nous montrons la supériorité de ces algorithmes par rapport à d'autres approches classiques. Des résultats sur la restauration d'images compressées sont présentés.

Abstract – In this work, we propose new algorithms to minimize total variation under some convex constraints. The proposed algorithms are inspired by a recent advance in convex optimization [4]. If the constraint is strongly convex we propose techniques based on Fenchel-Moreau duality. Otherwise, we minimize a smooth approximation of the total variation. The proposed methods are fast, their complexity is analysed. We compare their efficiency with some classical schemes. We show their superiority both theoretically and experimentally.

Depuis l'introduction de la variation totale en traitement d'images par Rudin, Osher et Fatemi [1], l'intérêt pour cet outil n'a cessé de croître. Il est bien adapté à la modélisation des images constantes par morceaux. Il sert aujourd'hui à de nombreuses tâches telles que le débruitage, la déconvolution, la reconstruction, l'agrandissement ou encore la décomposition d'images en partie texturée et géométrique.

Dans ce travail, nous proposons des algorithmes rapides pour résoudre ces problèmes, par minimisation de la variation totale sous contraintes. Les algorithmes proposés sont inspirés d'un résultat récent de Yurii Nesterov. Suivant la régularité de la contrainte, nous proposons d'utiliser des techniques reposant soit sur la dualité, soit sur une régularisation différentiable de la variation totale. Les méthodes proposées sont rapides et leur complexité est analysée. Nous montrons la supériorité de cette approche par rapport à d'autres méthodes classiques. Des résultats sur la restauration d'images compressées sont présentés.

1 Définitions

La norme euclidienne de \mathbb{R}^n sera notée $|\cdot|_2$:

$$|u|_2 = \sqrt{\sum_{i=1}^n |u_i|^2}$$

Une fonction J est dite L -Lipschitz différentiable sur un

ensemble K si et seulement si elle est différentiable et s'il existe L tel que :

$$|\nabla J(u^1) - \nabla J(u^2)|_2 \leq L|u^1 - u^2| \quad \forall (u^1, u^2) \in K^2$$

Le diamètre euclidien d'un ensemble K est défini par :

$$D = \sup_{(u^1, u^2) \in K^2} (|u^1 - u^2|_2)$$

La norme d'un opérateur linéaire $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$ est définie par :

$$\|A\|_2 = \max_{\{u \in \mathbb{R}^n, |u|_2 \leq 1\}} (|Au|_2)$$

Le projecteur euclidien sur un ensemble $K \subset \mathbb{R}^n$ est défini par :

$$\Pi_K(x) = \arg \min_{u \in K} (|u - x|_2)$$

La transformée de Fenchel d'une fonction $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ est définie par :

$$F^*(v) = \sup_{u \in \mathbb{R}^n} (\langle u, v \rangle - F(u))$$

où $\langle \cdot, \cdot \rangle$ représente le produit scalaire canonique.

Pour que les résultats énoncés par la suite soient corrects, les discrétisations des opérateurs div et ∇ doivent vérifier pour tout q et u :

$$\langle q, \nabla u \rangle = \langle -div(q), u \rangle$$

2 Complexité d'un problème

Les résultats énoncés dans cette partie, proviennent essentiellement du livre [3]. Nous considérons le problème suivant :

$$\inf_{u \in K} J(u) \quad (1)$$

où $J : K \subset \mathbb{R}^n \rightarrow \mathbb{R}$ est une fonction convexe, et K est un ensemble convexe, compact. Il est à noter qu'on ne suppose pas J strictement convexe, ni différentiable. Ce problème admet un ensemble convexe de solutions U qui n'est pas forcément réduit à un singleton. Nous noterons \bar{J} le minimum de (1).

Dans la suite, nous nous focaliserons au cas particulier où $J(u) = |\nabla u|_1 = \sum_{i=1}^n |(\nabla u)_i|$ qui correspond au problème de la minimisation de la variation totale. Dans cette partie, pour introduire la notion de complexité, nous considérons le problème général (1).

Pour mesurer la rapidité de convergence d'un algorithme, on se sert de la notion de ϵ -solution. Une ϵ -solution est un élément u_ϵ de K tel que $J(u_\epsilon) - \bar{J} \leq \epsilon$.

Un algorithme itératif génère une suite $\{u^k\}$ de K sensée vérifier $\lim_{k \rightarrow \infty} J(u^k) - \bar{J} = 0$. Imaginons que pour trouver la solution, on s'autorise *uniquement* l'utilisation des valeurs $J(u^k)$ et d'un élément du sous-différentiel $\partial J(u^k)$. On peut montrer (cf. [3] pour une assertion plus précise) qu'aucune méthode ainsi construite, n'a une complexité meilleure que $O(\frac{1}{\epsilon^2})$. Ceci signifie que pour obtenir une ϵ -solution, il faut au pire, avec un algorithme optimal, $\frac{C}{\epsilon^2}$ itérations. C est une valeur finie dépendant de K et de J .

Dans le cas de la variation totale, deux algorithmes couramment utilisés atteignent cette complexité 'optimale'. Nous les décrivons dans la partie suivante avant de proposer des solutions encore plus rapides exploitant les propriétés de la norme l^1 .

3 Sous-gradient et gradient projeté

J représente désormais la variation totale. On s'intéresse à sa minimisation sur un ensemble convexe K . Pour ce problème, deux schémas couramment utilisés atteignent une complexité 'optimale' en $O(\frac{1}{\epsilon^2})$. Il s'agit du sous-gradient projeté à pas variable et du gradient projeté à pas constant après approximation différentiable de la variation totale.

L'algorithme du sous-gradient projeté s'écrit comme suit :

$$\begin{cases} u^0 \in K \\ u^{k+1} = \Pi_K(u^k - t_k \frac{\eta^k}{|\eta^k|_2}) \end{cases} \quad (2)$$

avec $\eta^k \in \partial J(u^k)$. Une façon optimale de choisir la suite $\{t_k\}$ est :

$$t_k = \frac{J(u^k) - \bar{J}}{|\eta^k|_2} \quad (3)$$

Cette suite est inutilisable en pratique car \bar{J} est inconnu, il faut donc l'estimer itérativement [5]. Une autre possibilité 'optimale', mais plus lente en pratique consiste à choisir $t_k = \frac{D}{\sqrt{k}}$, où D est le diamètre euclidien de K .

La principale difficulté de ce type d'algorithme est donc de définir la suite t_k . Une manière de contourner ce problème est d'utiliser une approximation différentiable de la variation totale. Au lieu de minimiser J , on minimise $J_\mu(u) = \sum_{i=1}^n \psi(|(\nabla u)_i|)$. Dans la suite, nous utilisons l'approximation ¹ :

$$\psi(x) = \begin{cases} |x| & \text{si } |x| \geq \mu \\ \frac{x^2}{2\mu} + \frac{\mu}{2} & \text{si } |x| < \mu \end{cases} \quad (4)$$

J_μ est alors Lipschitz différentiable, on peut donc utiliser un algorithme de gradient projeté à pas constant. Celui-ci s'écrit comme suit :

$$\begin{cases} u^0 \in K \\ u^{k+1} = \Pi_K(u^k - t \nabla J_\mu(u^k)) \end{cases} \quad (5)$$

En choisissant $t = \frac{2\mu}{\|\text{div}\|_2^2}$, on peut montrer que l'on obtient un taux de convergence optimal. Comme on a fait une approximation de J , on ne peut pas espérer converger vers la solution de (1). Par contre, on peut l'approcher autant qu'on le souhaite en diminuant μ . De plus, on peut montrer que pour obtenir une ϵ -solution au problème (1), en choisissant μ de l'ordre de ϵ , il ne faut pas plus de $\frac{C}{\epsilon^2}$ itérations (C est une constante dépendant linéairement du diamètre de K).

Le gradient projeté et le sous-gradient projeté ont donc des taux de convergence voisins et 'optimaux' si les paramètres sont bien choisis. Cependant ces taux sont faibles. En effet, le nombre d'itérations nécessaire pour être sûr d'avoir une précision $\epsilon = 10^{-3}$ est de l'ordre de 10^6 . Dans le cas des images, on ne peut pas se permettre un tel nombre d'itérations. On doit se contenter d'une solution approchée, bien qu'il y ait une preuve de convergence. Dans les deux sections suivantes, nous présentons un algorithme en $O(\frac{1}{\epsilon})$ et un algorithme en $O(\frac{1}{\sqrt{\epsilon}})$.

4 Approximation pour les problèmes contraints

Dans cette partie, nous présentons un algorithme en $O(\frac{1}{\epsilon})$ pour minimiser la variation totale sous contraintes.

L'idée est de transformer les problèmes non-différentiables en problèmes différentiables, plus simples à minimiser. L'erreur d'approximation peut être compensée par la rapidité des algorithmes de minimisation. Yurii Nesterov dans [4] propose une méthode rapide pour minimiser des fonctions Lipschitz différentiables sous contraintes convexes. En utilisant ces résultats, on peut montrer le théorème suivant :

Théorème 1. *L'algorithme suivant :*

¹meilleure que l'approximation plus courante $\psi(x) = \sqrt{x^2 + \mu^2}$

- 1 Définir $k = -1$, $G^{-1} = 0$, $x^0 \in K$,
- 2 Définir $k = k + 1$, calculer $\eta_k = \nabla J_\mu(x^k)$,
- 3 Définir $y^k = \Pi_K(x^k - \frac{\eta_k}{L})$,
- 4 Définir $G^k = G^{k-1} + \frac{k+1}{2}\eta^k$,
- 5 Définir $z_k = \Pi_K(-\frac{G^k}{L})$,
- 6 Définir $x^{k+1} = \frac{2}{k+3}z^k + \frac{k+1}{k+2}y^k$,
- 7 Revenir à 2 jusqu'à $k = N$.

assure que :

$$0 \leq J(y^k) - \bar{J} \leq \frac{\|div\|_2^2 2D}{\mu k^2} + \frac{n\mu}{2} \quad (6)$$

où D est le diamètre euclidien de l'ensemble K .

Le terme $\frac{n\mu}{2}$ dans l'équation (6) représente l'erreur d'approximation, tandis que le terme $\frac{\|div\|_2^2 2D}{\mu k^2}$ est l'erreur de convergence. En utilisant la formule (6) on montre :

Théorème 2. Pour obtenir une ϵ -solution au problème (1), il suffit de choisir $\mu = \frac{\epsilon}{n}$ et :

$$N = \lfloor \frac{2\|div\|_2^2}{\epsilon} \sqrt{nD} \rfloor$$

Ce résultat montre que cet algorithme a une complexité en $O(\frac{1}{\epsilon})$. Utiliser une approximation différentiable de la norme l^1 est donc une bonne manière d'exploiter sa structure.

4.1 Dualité pour les problèmes $BV - l^2$

Récemment plusieurs auteurs ont proposé d'utiliser des méthodes primales-duales pour résoudre les problèmes de variation totale [7, 8]. Ces méthodes permettent souvent d'obtenir de meilleurs résultats de convergence expérimentaux, mais aucune analyse théorique ne justifie leur intérêt par rapport aux méthodes primales que nous avons présentées. Dans cette partie, nous montrons que l'utilisation des méthodes duales permet d'obtenir des algorithmes en $O(\frac{1}{\sqrt{\epsilon}})$ et de gagner encore un ordre de convergence par rapport aux méthodes présentées précédemment.

Nous nous intéressons maintenant au problème lagrangien suivant :

$$\inf_{\{u \in \mathbb{R}^n\}} (J(u) + F(u)) \quad (7)$$

où J est la variation totale. Nous supposons F différentiable et fortement convexe. Ceci signifie qu'il existe $\sigma > 0$ tel que pour tout u^1 et u^2 :

$$|F(u^1) - F(u^2)| \geq \sigma |u^1 - u^2|_2 \quad (8)$$

Sous ces hypothèses, on peut montrer en utilisant les principes de la dualité [6] le résultat suivant :

Théorème 3. Le problème dual de (7) est défini par :

$$\inf_{q \in K} (F^*(-div(q))) \quad (9)$$

avec $K = \{q \in (\mathbb{R}^n \times \mathbb{R}^n), \|q\|_\infty \leq 1\}$.

L'application $H : q \rightarrow F^*(-div(q))$ est L Lipschitz différentiable, avec $L = \frac{\|div\|_2^2}{\sigma}$. La solution \bar{u} de (7) et la solution \bar{q} de (9) sont liées par :

$$\bar{u} = \nabla F^*(-div(\bar{q}))$$

De plus $\nabla \bar{u} = \bar{q} |\nabla \bar{u}|$. Utiliser une méthode duale revient donc à faire évoluer les lignes de niveau de l'image plutôt que ses niveaux de gris.

En combinant ce théorème avec les résultats de Yurii Nesterov [4], on peut montrer que :

Théorème 4. L'algorithme suivant :

- 1 Définir $k = -1$, $G^{-1} = 0$, $L = \frac{\|div\|_2^2}{\sigma}$, $x^0 \in K$,
- 2 Définir $k = k + 1$, calculer $\eta_k = \nabla H(x^k)$,
- 3 Définir $y^k = \Pi_K(x^k - \frac{\eta_k}{L})$,
- 4 Définir $G^k = G^{k-1} + \frac{k+1}{2}\eta^k$,
- 5 Définir $z_k = \Pi_K(-\frac{G^k}{L})$,
- 6 Définir $x^{k+1} = \frac{2}{k+3}z^k + \frac{k+1}{k+2}y^k$,
- 7 Revenir à 2 jusqu'à $k = N$,
- 8 Poser $u^N = \nabla F^*(-div(y^k))$.

assure que :

$$0 \leq J(u^N) - \bar{J} \leq \frac{\|div\|_2^2 4n}{\sigma N^2} \quad (10)$$

Cet algorithme a donc une complexité en $O(\frac{1}{\sqrt{\epsilon}})$. Il permet de résoudre le problème lagrangien, sous des hypothèses de forte convexité de F . Une classe importante de fonctions fortement convexes est celle des fonctions basées sur des normes l^2 de transformées linéaires inversibles de l'image. On peut montrer le résultat suivant :

Théorème 5. Si A est inversible, et $\lambda > 0$ alors la transformée de Fenchel de $F : u \rightarrow |\lambda A(u - f)|_2^2$ est :

$$F^*(v) = \frac{1}{\lambda} \langle A^{-1}f, v \rangle + \frac{1}{4\lambda^2} \|A^{-*}v\|$$

De plus F est fortement convexe, avec $\sigma = \frac{2\lambda^2}{\|A^{-1}\|_2^2}$.

4.2 Résultats

Les courbes de la figure (1) montrent la décroissance de la fonction coût en fonction du nombre d'itérations pour le problème de Rudin-Osher-Fatemi. Les approches indiquées 'primal' résolvent :

$$\inf_{u, \|u-f\|_2^2 \leq \alpha} (J(u))$$

Les algorithmes indiqués 'dual' résolvent :

$$\inf_u (J(u) + \lambda \|u - f\|_2^2)$$

Les paramètres α et λ ont été choisis de telle manière que les solutions des deux problèmes soient identiques. Le paramètre d'approximation μ vaut 0.005, ce qui assure que les solutions approchées et exactes soient identiques perceptuellement. Dans la suite nous utilisons les notations suivantes. *PSD* - Primal Projected Subgradient Descent. *PPG* - Primal Projected Gradient Descent. *PN* - Primal Nesterov. *DPG* - Dual Projected Gradient Descent. *DN* - Dual Nesterov.

La première remarque est que l'algorithme *DN* est le plus efficace. La méthode *DPG* est une méthode légèrement plus efficace que l'algorithme proposé par A. Chambolle [7]. Il faut 300 itérations à *DPG* pour atteindre la précision au bout de 100 itérations de *DN*. L'algorithme *PN* présente aussi de bonnes performances. Il est même plus efficace que l'algorithme *DPG* entre 100 et 200 itérations. Les algorithmes *PSG* et *PPG* (qui n'a pas été représenté par souci de lisibilité) sont nettement moins efficaces. Les résultats expérimentaux confirment bien les taux de convergence théoriques.

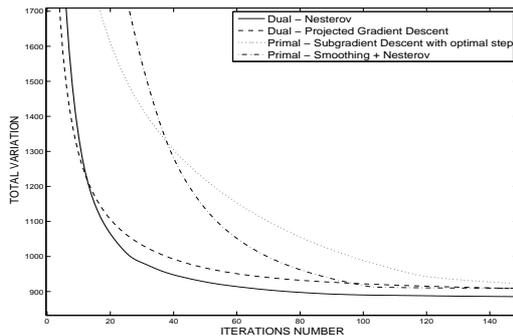


FIG. 1 – Comparaison des taux de convergence de différentes méthodes

Pour finir, nous présentons un résultat de décompression sur la figure (2). De nombreux algorithmes de compression sont basés sur des transformées en cosinus ou en ondelettes. Ils sont efficaces, mais ont pour défaut de créer des artefacts tels que des oscillations le long des contours. On peut montrer [2] qu'un moyen de restaurer ces images consiste à chercher la fonction à variation totale minimale dans l'ensemble des antécédents de l'image compressée. Cet ensemble est convexe, non lisse, on utilise donc une approximation de la variation totale.

Pour l'image en bas à gauche, nous utilisons une approximation grossière de la variation totale ($\mu = 0.01$). L'image est correctement restaurée, les détails sont préservés bien qu'il y ait une perte de contraste. Sur l'image en bas à droite, on utilise une approximation très fine ($\mu = 0.0001$). On voit un effet d'escalier typique de la variation totale. Pour certaines applications, il vaut donc mieux utiliser une approximation différentiable de la norme l^1 . Ceci augmente la rapidité des algorithmes (20 itérations sont nécessaires pour obtenir l'image en bas à gauche) et la qualité visuelle.

5 Conclusion

Nous avons proposé des algorithmes rapides pour minimiser la variation totale sous des contraintes convexes. Nous avons montré leur supériorité théorique et pratique par rapport à plusieurs approches classiques de premier ordre.

Précisons finalement que les algorithmes présentés se généralisent facilement à la minimisation de normes l^1



FIG. 2 – Reconstruction d'une image après quantification des coefficients d'ondelettes Daubechies 9-7. HG : original - HD : compressée - BG : Décompressée avec $\mu = 0.01$ - BD : Décompressée avec $\mu = 0.0001$

de transformées linéaires d'une image. Ces algorithmes peuvent donc être utilisés pour de très nombreuses applications.

Références

- [1] L. Rudin, S. Osher and E. Fatemi *Nonlinear Total Variation Based Noise Removal* Physica D, 1992.
- [2] Sylvain Durand and Jacques Froment *Reconstruction of wavelet coefficients using total variation minimization* SIAM Journal of Scientific Computing, Vol.24, 2003
- [3] A.S. Nemirovskii and D.B. Yudin *Problem Complexity and Method Efficiency in Optimization* Wiley, 1983
- [4] Yurii Nesterov *Smooth minimization of non-smooth functions* Mathematic Programming, Ser. A, Vol.103, 2005
- [5] K.C. Kiwiel *The efficiency of subgradient projection methods for convex optimization. Part I : General level methods and Part II : Implementations and extensions* SIAM J. Control Optim., Vol.34, 1996
- [6] I. Ekeland and R. Temam *Analyse convexe et problèmes variationnels* Dunod Gauthier-Villars 1974
- [7] A. Chambolle *An Algorithm for Total Variation Minimization and Applications* JMIIV, Vol. 20, 2004
- [8] Tony F. Chan, Gene H. Golub and Pep Mulet *A Non-linear Primal-Dual Method for Total Variation-Based Image Restoration* SIAM Journal of Scientific Computing, Vol. 20 (6), 1999