Algorithmes de Majorisation-Minimisation et Factorisation Structurée de Matrices

Julien MAIRAL[1]

[1]Inria, Lear Team, Laboratoire Jean Kuntzmann, CNRS, Univ. Grenoble Alpes.
655, avenue de l'Europe, 38330 Montbonnot, France.
julien.mairal@inria.fr

Résumé – Les algorithmes de majorisation-minimisation consistent à minimiser de façon itérative une majorante de la fonction objectif. De par sa simplicité et ses nombreuses applications, ce principe a obtenu des succès importants en statistiques et traitement du signal. Dans ce travail, nous souhaitons faire passer ce principe à l'échelle. Nous proposons un schéma d'optimisation stochastique qui est capable de traiter des jeux de données de grande taille, voire de taille infinie. Nous montrons que ce schéma converge presque sûrement vers des points stationnaires pour une large classe de problèmes non-convexes, et admet des taux de convergence rapides pour des problèmes convexes. Nous développons plusieurs algorithmes efficaces à partir de ce schéma d'optimisation générique. Tout d'abord, nous proposons un nouvel algorithme proximal stochastique. Ensuite, nous montrons que notre approche est efficace pour des problèmes de factorisation structurée de grandes matrices.

Abstract – Majorization-minimization algorithms consist of iteratively minimizing a majorizing surrogate of an objective function. Because of its simplicity and its wide applicability, this principle has been quite successful in statistics and in signal processing. In this paper, we intend to make this principle scalable. We introduce a stochastic majorization-minimization scheme which is able to deal with large-scale or possibly infinite data sets. We show that our scheme almost surely converges to stationary points for a large class of non-convex problems and enjoys fast convergence rates for convex problems. We develop several efficient algorithms based on our framework. First, we propose a new stochastic proximal gradient method. Second, we demonstrate the effectiveness of our approach for solving large-scale non-convex structured matrix factorization problems.

1 Introduction

En optimisation mathématique, le principe de majorisation-minimisation [6] consiste à minimiser à chaque itération une fonction de substitution qui majore la fonction objectif, en faisant ainsi décroitre cette dernière. De nombreuses procédures existantes utilisent le principe de majorisation-minimisation, soit directement, soit indirectement. Par exemple, les algorithmes d'espérance-maximisation (EM) (voir [2, 11]) sont fondés sur des fonctions de substitution construites avec l'inégalité de Jensen pour un modèle de vraisemblance. D'autres approches peuvent aussi être interprétées du point de vue de majorisation-minimisation, comme la programmation DC pour minimiser la différence de fonctions convexes [5], ou certains algorithmes de gradient proximaux [1, 13, 3].

Dans ce papier, nous proposons un algorithme de majorisation-minimisation stochastique, qui est adapté aux problèmes à large échelle que l'on rencontre en apprentissage statistique et en traitement du signal. Plus précisément, nous nous intéressons à la minimisation d'un coût pouvant s'écrire sous la forme d'une espérance par rapport à la distribution des données. Pour de telles fonctions objectif, des méthodes d'apprentissage en ligne fondées sur des approximations stochastiques se sont montrées particulièrement efficaces, et ont fait l'objet de nombreux travaux [4, 12].

Notre schéma, que nous avons introduit à l'origine dans [8], suit cette direction de recherche. Il consiste à construire de façon itérative une fonction substitut de la fonction objectif en n'autorisant l'observation d'un seul échantillon des données à chaque itération ; cet échantillon est utilisé pour mettre à jour la fonction substitut, qui est elle-même minimisée pour obtenir une nouvelle estimée de la solution. Ce principe est fortement relié aux algorithmes EM stochastiques [2, 11] et à des méthodes d'apprentissage de dictionnaires en ligne (voir [9]). En comparaison de ces travaux, nous nous intéressons à des problèmes d'optimisation plus généraux.

2 Le principe MM

Dans ce papier, nous souhaitons minimiser une fonction continue $f : \mathbb{R}^p \to \mathbb{R}$:

$$\min_{\theta \in \Theta} f(\theta), \tag{1}$$

où $\Theta \subseteq \mathbb{R}^p$ est un ensemble convexe. Le principe de majorisation-minimisation consiste à calculer une majorante g_n de f à l'itération n et de mettre à jour l'estimée courante par

$$\theta_n \in \arg\min_{\theta \in \Theta} g_n(\theta).$$

Ce principe très simple est illustré en Figure 1.

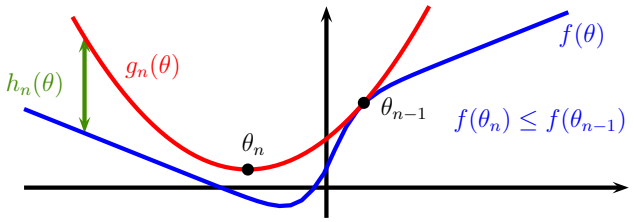

FIGURE 1 – Illustration du principe de majorisation-minimisation. La fonction substitut g_n majore f et est tangente à f au point θ_{n-1}. La nouvelle estimée correspond au minimum de g_n. Nous appelerons h_n l'erreur d'approximation.

3 Un schéma stochastique

Dans un grand nombre d'applications en apprentissage statistique, la fonction f à minimiser prend la forme d'une espérance

$$\min_{\theta\in\Theta} \Big[f(\theta) \triangleq \mathbb{E}_{\mathbf{x}}[\ell(\mathbf{x},\theta)] \Big], \tag{2}$$

où $\mathbf{x}$ représente un échantillon d'un certain ensemble $\mathcal{X}$ tiré d'après une distribution inconnue ; ℓ est une fonction de perte continue. Comme souvent dans la littérature [12], nous supposons que toutes les espérances sont bien définies et de valeur finie ; nous supposons aussi que f est bornée inférieurement.

Notre approche pour minimiser (2) est présentée dans l'algorithme 1. A chaque itération, nous tirons un point $\mathbf{x}_n$, en supposons que ces points forment un échantillon i.i.d. de la distribution des données. Nous choisissons ensuite une fonction de substitution g_n pour la fonction $f_n : \theta \mapsto \ell(\mathbf{x}_n,\theta)$. Celle-ci est utilisée pour mettre à jour une autre fonction auxiliaire $\bar{g}_n$, qui est une moyenne pondérée des fonctions g_k pour $k \leq n$. La raison d'être de la fonction $\bar{g}_n$ est de se comporter asymptotiquement comme une fonction majorante de substitution de l'espérance f. $\bar{g}_n$ est construite avec une suite de poids $(w_n)_{n\geq 1}$ qui seront discutés plus tard. Enfin, les "options 2 et 3" implémentent une stratégie de moyennage des estimées, qui est classique en optimisation stochastique [12].

Algorithm 1 Majorisation-minmisation stochastique

input $\theta_0 \in \Theta$ (éstimée initiale) ; N (nombre d'itérations) ; $(w_n)_{n\geq 1}$, séquence de poids dans $(0,1]$;
1: initialisation : $\bar{g}_0 : \theta \mapsto \frac{\rho}{2}\|\theta-\theta_0\|_2^2$; $\bar{\theta}_0 = \theta_0$; $\hat{\theta}_0 = \theta_0$;
2: **for** $n = 1,\dots,N$ **do**
3: tirage d'un échantillon $\mathbf{x}_n$; $f_n : \theta \mapsto \ell(\mathbf{x}_n,\theta)$;
4: choix d'une fonction substitut g_n de f_n en θ_{n-1} ;
5: mise à jour : $\bar{g}_n = (1-w_n)\bar{g}_{n-1} + w_n g_n$;
6: mise à jour : $\theta_n \in \arg\min_{\theta\in\Theta} \bar{g}_n(\theta)$;
7: pour option 2, $\hat{\theta}_n \triangleq (1-w_{n+1})\hat{\theta}_{n-1} + w_{n+1}\theta_n$;
8: pour option 3, $\bar{\theta}_n \triangleq \frac{(1-w_{n+1})\bar{\theta}_{n-1}+w_{n+1}\theta_n}{\sum_{k=1}^{n+1} w_k}$;
9: **end for**
output (option 1) : θ_N (pas de moyennage) ;
output (option 2) : $\bar{\theta}_N$ (premier type de moyennage) ;
output (option 3) : $\hat{\theta}_N$ (second type de moyennage).

3.1 Choix de g_n et algorithme proximal

Intuitivement, le succès du schéma de majorisation-minimisation présenté en Figure 1 dépend de la qualité d'approximation fournie par la fonction substitut g_n. Dans ce paper, cette qualité d'approximation est garantie par la définition suivante, qui impose à l'erreur d'approximation d'être régulière.

Definition 3.1 (Fonctions substitut de premier ordre). *Soit κ un point de Θ. Nous appelons fonctions substitut de premier ordre l'ensemble $\mathcal{S}_{L,\rho}(f,\kappa)$ de fonctions g qui sont (i) ρ-fortement convexes ; (ii) majorantes $g \geq f$; (iii) tangentes à f : $g(\kappa) = f(\kappa)$; avec une erreur d'approximation $h = g - f$ différentiable et dont le gradient ∇h est L-Lipschitz continu.*

Cet ensemble $\mathcal{S}_{L,\rho}(f,\kappa)$ a été introduit dans le cadre d'algorithmes de majorisation-minimisation dans [7]. L'exemple de fonction substitut de premier ordre qui nous intéresse ici s'applique aux fonctions composites, mais d'autres exemples sont présentés dans [7]. Considérons ainsi une fonction f qui peut s'écrire comme la somme de deux fonctions $f_1 + f_2$, avec f_1 differentiable, ∇f_1 L-Lipschitz continu et f_2 convexe. Alors, la fonction suivante est dans $\mathcal{S}_{L-\mu,L}(f,\kappa)$, où μ est le paramètre de forte convexité de f (éventuellement égal à 0) :

$$g : \theta \mapsto f_1(\kappa) + \nabla f_1(\kappa)^\top(\theta-\kappa) + \frac{L}{2}\|\theta-\kappa\|_2^2 + f_2(\theta). \tag{3}$$

Minimiser g correspond à effectuer une étape d'algorithme de gradient proximal [1, 13] :

$$\theta \leftarrow \arg\min_{\theta} \frac{1}{2}\left\|\kappa - \frac{1}{L}\nabla f_1(\kappa) - \theta\right\|_2^2 + \frac{1}{L} f_2(\theta).$$

Dans le cadre de l'algorithme stochastique que nous proposons, nous pouvons utiliser ce type de fonction substitut pour minimiser une espérance régularisée par une fonction ψ convexe non nécessairement différentiable et une fonction de perte $\theta \mapsto \ell(\mathbf{x},\theta)$ qui satisfait les hypothèses de la fonction f_1 pour tout $\mathbf{x}$:

$$\min_{\theta\in\Theta} \mathbb{E}_{\mathbf{x}}[\ell(\mathbf{x},\theta)] + \psi(\theta), \tag{4}$$

L'algorithme 1 produit ainsi une séquence de mises à jour

$$\theta_n \leftarrow \arg\min_{\theta\in\Theta} \sum_{i=1}^{n} w_n^i \left[\nabla f_i(\theta_{i-1})^\top\theta + \tfrac{L}{2}\|\theta-\theta_{i-1}\|_2^2 + \psi(\theta)\right]. \tag{SMM}$$

où la séquence de poids $(w_n)_{n\geq 1}$ est choisie telle que $w_1 = 1$, et les quantités w_n^i sont définies récursivement par $w_n^i \triangleq (1-w_n)w_n^{i-1}$ for $i < n$ et $w_n^n \triangleq w_n$. Cette séquence de mises à jour est reliée à l'algorithme FOBOS [4] :

$$\theta_n \leftarrow \arg\min_{\theta\in\Theta} \nabla f_n(\theta_{n-1})^\top\theta + \tfrac{1}{2\eta_n}\|\theta-\theta_{n-1}\|_2^2 + \psi(\theta), \tag{FOBOS}$$

ainsi qu'au moyennage dual régularisé de [14] :

$$\theta_n \leftarrow \arg\min_{\theta\in\Theta} \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\theta_{i-1})^\top\theta + \tfrac{1}{2\eta_n}\|\theta\|_2^2 + \psi(\theta). \tag{RDA}$$

En comparaison, notre schéma implique une moyenne pondérée des gradients et des estimées obtenus précédemment.

3.2 Analyse de convergence

Plusieurs résultats de convergence peuvent être obtenus. Dans le cas convexe, nous nous intéressons au taux de convergence de la séquence $\mathbb{E}[f(\bar{\theta}_n) - f^\star]$, où $f^\star$ est la valeur minimale de la fonction objectif. Dans le cas non-convexe, nous étudions la convergence de l'algorithme vers des points stationnaires. Toutes les preuves sont disponibles dans l'appendice de [8].

Cas convexe. Tout d'abord, nous étudions le cas de fonctions convexes $f_n : \theta \mapsto \ell(\theta, \mathbf{x}_n)$, et nous faisons l'hypothèse suivante **(A)** : pour tout θ dans Θ, les fonctions f_n sont R-Lipschitz continues. Dans ce cas, nous avons la proposition suivante.

Proposition 3.1 (**Taux de convergence).**
*Lorsque les fonctions f_n sont convexes et sous hypothèse **(A)**, et lorsque les fonctions g_n sont dans $\mathcal{S}_{L,L}(f, \kappa)$, nous avons pour tout $n \geq 1$,*

$$\mathbb{E}[f(\bar{\theta}_{n-1}) - f^\star] \leq \frac{L\|\theta^\star - \theta_0\|_2^2 + \frac{R^2}{L}\sum_{k=1}^n w_k^2}{2\sum_{k=1}^n w_k}, \qquad (5)$$

où $\bar{\theta}_{n-1}$ est défini dans l'algorithme 1, et $\theta^\star$ minimise f sur Θ.

Nous pouvons alors en déduire le corollaire suivant :

Corollary 3.1 (**Taux de convergence à horizon infini).**
Sous les mêmes hypothèse que dans la proposition 3.1 et en choisissant des poids $w_n = \gamma/\sqrt{n}$, alors pour tout $n \geq 2$,

$$\mathbb{E}[f(\bar{\theta}_{n-1}) - f^\star] \leq \frac{L\|\theta^\star - \theta_0\|_2^2}{2\gamma\sqrt{n}} + \frac{R^2\gamma(1+\log(n))}{2L\sqrt{n}}.$$

Notons que que la borne $O(1/\sqrt{n})$ ne peut pas être améliorée en général sans faire d'hypothèse supplémentaire sur la fonction objectif [12]. Le taux de convergence est donc "optimal" au terme logarithmique près. Notons aussi que notre analyse suggère d'utiliser des poids de la forme $O(1/\sqrt{n})$. En pratique, nous avons observé qu'un choix judicieux était une séquence de poids de la forme $w_n = \sqrt{n_0+1}/\sqrt{n_0+n}$, où n_0 est ajusté sur un sous-échantillon de l'ensemble d'apprentissage.

Si nous faisons maintenant l'hypothèse **(B)** que les fonctions f_i sont μ-fortement convexes, nous obtenons un taux de convergence optimal en $O(1/n)$:

Proposition 3.2 (**Taux de convergence à horizon infini).**
*Sous les hypothèses **(A)** et **(B)**, avec des fonctions g_n choisies dans $\mathcal{S}_{L,L-\mu}(f, \kappa)$, nous avons pour tout $n \geq 1$,*

$$\mathbb{E}[f(\hat{\theta}_{n-1}) - f^\star] \leq \max\left(\frac{2R^2}{\mu}, \rho\|\theta^\star - \theta_0\|_2^2\right)\frac{1}{\beta n + 1},$$

avec $\beta \triangleq \frac{\mu}{L}$,$w_n \triangleq \frac{1+\beta}{1+\beta n}$, et $\hat{\theta}_n$ est défini dans l'algorithme 1.

Cas non-convexe. Les résultats de convergence pour les problèmes non-convexe sont faibles par nature et difficile à obtenir dans le cas de l'optimisation stochastique. Les analyses classiques se concentrent donc sur l'analyse de convergence vers des points stationnaires. Dans le cas présent, nous faisons l'hypothèse que toutes les fonctions $f_n : \theta \mapsto \ell(\mathbf{x}_n, \theta)$ admettent des dérivées directionnelles $\nabla f_n(\theta, \theta' - \theta)$ en tout point θ et direction $\theta' - \theta$. Une condition nécessaire classique d'optimalité pour un point $\theta^\star$ est d'avoir des dérivées directionnelles positives en toute direction faisable.

Notre résultat principal de convergence est alors le suivant :

Proposition 3.3 (**Convergence presque sûre).**
Supposons que Θ et le support des données $\mathcal{X}$ soient compacts, que les fonctions f_n soient uniformément bornées et R-Lipschitz continues, que la séquence de poids w_n soit décroissante et telle que $w_1 = 1$, $\sum_{n\geq 1} w_n = +\infty$, et $\sum_{n\geq 1} w_n^2\sqrt{n} < +\infty$; alors $(f(\theta_n))_{n\geq 0}$ converge presque sûrement et

$$\liminf_{n\to+\infty} \inf_{\theta\in\Theta} \frac{\nabla \bar{f}_n(\theta_n, \theta - \theta_n)}{\|\theta - \theta_n\|_2} \geq 0,$$

où la fonction $\bar{f}_n$, définie récursivement par $\bar{f}_n = (1-w_n)\bar{f}_{n-1} + w_n f_n$, converge uniformément vers f.

La condition de dérivée directionnelle positive caractérisant les points stationnaires du problème, la garantie asymptotique obtenue est donc satisfaisante.

4 Factorisation de matrice

Nous illustrons maintenant la méthode pour un problème de factorisation structurée de matrice. Nous considérons une grande collection de signaux $(\mathbf{x}_i)_{i=1}^N$ in $\mathbb{R}^m$, et nous voulons trouver un dictionnaire $\mathbf{D}$ dans $\mathbb{R}^{m\times K}$ qui peut représenter ces signaux de façon parcimonieuse (voir [9]). La qualité d'un dictionnaire $\mathbf{D}$ peut être mesurée grâce à une fonction de perte $\ell(\mathbf{x}, \mathbf{D}) \triangleq \min_{\boldsymbol{\alpha}\in\mathbb{R}^K} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda_1\|\boldsymbol{\alpha}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\alpha}\|_2^2$.

Nous cherchons alors un dictionnaire qui minimise la perte sur l'ensemble d'apprentissage :

$$\min_{\mathbf{D}\in\mathbb{R}^{m\times K}} \mathbb{E}_{\mathbf{x}}\left[\ell(\mathbf{x}, \mathbf{D})\right] + \varphi(\mathbf{D}), \qquad (6)$$

où φ est une fonction de régularisation. Le lien avec la factorisation de matrice est visible si l'on remarque qu'après avoir résolu (6), chaque signal $\mathbf{x}_i$ peut être approché par un produit $\mathbf{D}\boldsymbol{\alpha}_i$. En d'autres termes, la matrice $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ est factorisée par un produit $\mathbf{DA}$, la matrice $\mathbf{A}$ étant parcimonieuse.

Une régularisation classique consiste à forcer les colonnes de $\mathbf{D}$ à avoir une norme ℓ_2 plus petite que l'unité. Dans ce cas, la fonction φ encode un ensemble de contraintes et vaut $+\infty$ si les contraintes ne sont pas satisfaites et zéro sinon. Il est alors possible d'utiliser la fonction substitut $g_n : \mathbf{D} \mapsto \frac{1}{2}\|\mathbf{x}_n - \mathbf{D}\boldsymbol{\alpha}_n^\star\|_2^2 + \lambda_1\|\boldsymbol{\alpha}_n^\star\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\alpha}_n^\star\|_2^2$ avec

$$\boldsymbol{\alpha}_n^\star = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x}_n - \mathbf{D}_{n-1}\boldsymbol{\alpha}\|_2^2 + \lambda_1\|\boldsymbol{\alpha}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\alpha}\|_2^2.$$

L'algorithme 1 devient alors la méthode d'apprentissage en ligne de dictionnaires de [10]. Lorsque φ est une fonction de régularisation plus générale, nous pouvons à la place utiliser la

fonction de substitution proximale présentée dans l'équation (4), et qui s'écrit ici comme la fonction g_n qui à $\mathbf{D}$ associe

$$f_n(\mathbf{D}) + \mathrm{Tr}(\nabla f_n(\mathbf{D}_{n\text{-}1})^\top (\mathbf{D} - \mathbf{D}_{n\text{-}1})) + \frac{L}{2}\|\mathbf{D} - \mathbf{D}_{n\text{-}1}\|_2^2 + \varphi(\mathbf{D}). \tag{7}$$

Il suffit alors de remarquer que $\nabla f_n(\mathbf{D}_{n\text{-}1}) = (\mathbf{D}_{n\text{-}1}\boldsymbol{\alpha}_n^\star - \mathbf{x}_n)\boldsymbol{\alpha}_n^{\star\top}$, et nous obtenons un algorithme concret pour trouver un point stationnaire de (6). Une difficulté consiste à trouver une bonne constante L, ce qui peut nécessiter une recherche linéaire.

Nous illustrons maintenant la méthode avec un exemple pratique, obtenu sur un ensemble de $N = 400\,000$ imagettes $\mathbf{x}_n$ de taille $m = 20 \times 20$ pixels extraites d'images naturelles blanchies. Nous visualisons quelques éléments d'un dictionnaire $\mathbf{D}$ contenant 256 éléments, obtenu avec le logiciel SPAMS [10]. Ces éléments sont presque parcimonieux mais contiennent du bruit résiduel. Nous appliquons alors notre méthode en utilisant les fonctions g_n définies en (7), lorsque φ est une pénalité induisant de la parcimonie structurée.

Plus précisément, nous choisissons une fonction de régularisation φ qui encourage des pixels voisins à être mis à zéro simultanément. Pour ce faire, nous définissons la collection $\mathcal{G}$ de groupes de pixels correspondant à des voisinages carrés de taille 4×4, et alors $\varphi(\mathbf{D}) \triangleq \gamma_1 \sum_{j=1}^K \sum_{g \in \mathcal{G}} \max_{k \in g} |\mathbf{d}_j[k]| + \gamma_2 \|\mathbf{D}\|_{\mathrm{F}}^2$, où $\mathbf{d}_j$ est la colonne j de $\mathbf{D}$. Ce genre de pénalité induit l'effet escompté pour des raisons détaillées dans [9]. Par ailleurs, nous savons calculer efficacement l'opérateur proximal de φ, ce qui rend notre méthode applicable.

Apprendre le dictionnaire de $K = 256$ éléments nous a pris quelques minutes sur un ordinateur portable en effectuant une passe sur les données d'entraînement et en choisissant les paramètres comme détaillé dans [8]. Notons que lorsque γ_2 est assez grand, les itérées $\mathbf{D}_n$ restent nécessairement dans un domaine compact, et ainsi nous pouvons appliquer notre analyse de convergence présentée précédemment.

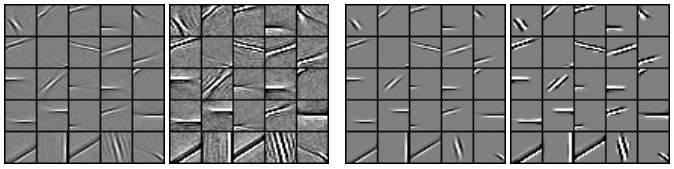

FIGURE 2 – A gauche : Visualisation de 25 éléments d'un dictionnaire plus large obtenu par le logiciel SPAMS [10] ; la deuxième vue amplifie les petits coefficients. A droite : la même vue pour le dictionnaire obtenu avec la fonction de régularisation φ structurée après initialisation avec le dictionnaire présenté à gauche. Les élements du dictionnaires contiennent des zones contiguës égales à zéro.

5 Conclusion

Dans ce travail, nous avons introduit un algorithme de majorisation-minimisation stochastique qui permet de traiter des jeux de données de très grande taille. Nous avons montré que l'algorithme a des garanties théoriques solides et une valeur pratique dans le contexte de la factorisation de matrice. Par ailleurs, il est intéressant de noter que d'autres variantes ont été proposées, telles que des algorithmes incrémentaux ou de descente de coordonnées par blocs [7].

Remerciements

Ce travail a été réalisé grâce à un financement du projet Gargantua (Mastodons – CNRS) et de l'Agence Nationale de la Recherche (projet MACARON ANR-14-CE23-0003-01).

Références

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1) :183–202, 2009.

[2] O. Cappé and E. Moulines. On-line expectation–maximization algorithm for latent data models. *J. Roy. Stat. Soc. B*, 71(3) :593–613, 2009.

[3] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, 2010.

[4] J. Duchi and Y. Singer. Efficient online and batch learning using forward backward splitting. *J. Mach. Learn. Res.*, 10 :2899–2934, 2009.

[5] R. Horst and N.V. Thoai. DC programming : overview. *J. Optim. Theory App.*, 103(1) :1–43, 1999.

[6] K. Lange. *Optimization*. Springer-Verlag, 2004.

[7] J. Mairal. Optimization with first-order surrogate functions. In *Proc. ICML*, 2013.

[8] J. Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Adv. NIPS*, 2013.

[9] J. Mairal, F. Bach, and J. Ponce. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3) :85–283, 2014.

[10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11 :19–60, 2010.

[11] R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, 89, 1998.

[12] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optimiz.*, 19(4) :1574–1609, 2009.

[13] Y. Nesterov. Gradient methods for minimizing composite objective functions. Technical report, CORE Discussion Paper, 2007.

[14] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.*, 11 :2543–2596, 2010.