TP1: Gradient Proximal

Pirashanth RATNAMOGAN , Othmane SAYEM

ratnamogan@ensta.fr, sayem@ensta.fr

1 Description de la méthode LASSO

La méthode LASSO consiste à prédire de manière linéaire une variable donnée b à partir de variables explicatives $u_1, u_2..., u_n$. Le but étant de faire une regression, une minimisation naive des moindres carrés a l'inconvénient de distribuer l'erreur sur toutes les variables explicatives, il est indispensable de rajouter un terme de "régularisation" au problème afin que celui-ci se généralise mieux à des nouveaux exemples. Le but étant d'apporter une solution au dilemme entre biais et variance.

Afin d'assurer la séléction des variables les plus pertinentes uniquement, il serait donc judicieux de rajouter un terme de coût lié au nombre de variables non nulles utilisées. le problème d'optimisation prendrait ainsi la forme suivante :

$\min_u ||Au - b||_2^2 + \lambda |u|_0$ *où* $|u|_0$ *est la norme qui représente le nombre de composantes non nulles de* u

Toutefois, ce problème est beaucoup plus difficile à résoudre que le problème des moindres carrés, étant donné qu'il correspond à un problème d'optimisation combinatoire de taille n. La norme de comptage peut cependant être approximée par la norme 1. Le problème qu'on cherchera à résoudre lors de ce tp, prendra alors la forme suivante :

$$\min_u ||Au - b||_2^2 + \lambda |u|_1 \tag{1.1}$$

La fonction objectif étant composée d'une fonction différentiable F(u) ($F : u \rightarrow \quad ||Au-b||_2^2$), ainsi que d'une fonction non différentiable $G(u)$ ($G : u \rightarrow \quad \lambda|u|_1$), le problème correspond à un problème d'optimisation non différentiable, pour lequel on utilisera l'algorithme du gradient proximal.

Afin d'appliquer l'algorithme du gradient proximal, on calcule tout d'abord l'opérateur proximal correspondant à la fonction non différentiable G, ainsi que le gradient de la fonction différentiable F :

$$\nabla F(u) = A^T.(A.u - b)$$

$$(P_{\epsilon G}(u))_i = \begin{cases} u_i - \lambda\epsilon & \text{si } u_i > \lambda\epsilon \\ 0 & \text{si } |u_i| \leq \lambda\epsilon \\ u_i + \lambda\epsilon & \text{si } u_i < -\lambda\epsilon \end{cases}$$

2 Algorithme du gradient proximal

L'algorithme du gradient proximal que nous avons implémenté suit le schéma classique suivant :

— 1 - On choisit un point initial u_0, et on définit le pas $= \frac{1}{L}$

— 2 - à l'itération k :

$$u_{k+1} = P_{\epsilon G}\left[u_k - \epsilon.\nabla F(u_k)\right] = P_{\epsilon G}\left[u_k - \epsilon.A^T.(A.u_k - b)\right]$$

— 3 - Tant que le test d'arrêt n'est pas vérifié, $k \leftarrow k+1$ puis retour à l'étape 2.

3 FISTA : Version accélérée de l'algorithme du gradient proximal

Comme on a pu le voir en cours une version accélérée de l'algorithme du gradient proximal existe, elle est donnée par l'algorithme ci-dessous :

— 1 - On choisit un point initial $u_=v_0$ dans le domaine de G, et on définit le pas $= \frac{1}{L}$

— 2 - à l'itération k :

$$u_{k+1} = P_{\epsilon G}\left[v_k - \epsilon.\nabla F(u_k)\right] = P_{\epsilon G}\left[v_k - \epsilon.A^T.(A.u_k - b)\right] \qquad v_{k+1} = u_{k+1} + \frac{k-1}{k+2}(u_{k+1} - u_k)$$

— 3 - Tant que le test d'arrêt n'est pas vérifié, $k \leftarrow k+1$ puis retour à l'étape 2.

Dans cet algorithme on voit que la varible v correspond à une extrapolation de la variable u au cours des itérations. En pratique cet algorithme converge beaucoup plus rapidement que la version précedemment décrite (voir section résultats).

4 Choix des paramètres

4.1 Choix du pas et critère de convergence de l'algorithme

On vérifie facilement que les fonctions F et G que l'on a définies dans la partie 1 vérifient les **hypothèses 3** du cours. On a $\nabla F(u) = A^T.(A.u - b)$ d'où :

$$\begin{aligned}\forall u, v ||\nabla F(u) - \nabla F(v)|| =& ||A^T.(A.(u-v))|| \\ &\leq \max_{s \in Sp(A^TA)} s||u-v||\end{aligned}$$

D'où F est convexe, différentiable, à gradient lipschitzien de paramètre $L = \max_{s \in Sp(A^TA)} s$ où $Sp(M)$ represente le spectre de la matrice M.
La fonction G est convexe et propre (car c'est une norme).
La fonction $F + G$ est fini, de plus elle est convexe et coercive donc elle atteint un minimum en un point $u \in \mathcal{U}$

Ainsi, pour un pas fixé au cours des itérations de l'algorithme ϵ qui vérifie :

$$\epsilon \leq \frac{1}{L}$$

On a d'après le **Théorême 15** du cours une preuve de convergence des algorithmes du gradient proximal et de sa version accélérée FISTA.

On a donc décidé de fixer le pas au cours de nos travaux à une valeur $\epsilon = \frac{1}{L}$

4.2 Critère d'arrêt

Pour le choix du critère d'arrêt, on s'est basé sur quatres erreurs, qu'on a implémentées et essayées : les normes 1 et 2 de la différence $u_{k+1} - u_k$, ainsi que les normes 1 et 2 de la différence $F(u_{k+1}) + G(u_{k+1}) - G(u_k) - F(u_k)$, et finalement les deux critères d'arrêt ensemble.

5 Test et résultats

5.1 Comparaison des résultats de l'algorithme du gradient proximal "classique" et de FISTA

On a vu qu'en théorie l'algorithme du gradient proximal et FISTA ont des résultats de convergence équivalent. L'algorithme FISTA qui fait à chaque étape une extrapolation de la future variable du coefficient, se veut être une accélaration de l'algorithme du gradient proximal de base.
En pratique sur **tous** les tests que l'on a pu faire, l'algorithme FISTA donne des résultats équivalents à l'algorithme du gradient proximal de base en un nombre d'itérations réduit. Un exemple de cette accélération est donné dans l'annexe. Lors des tests on peut tracer des courbes qui donnent l'évolution de la fonction objective pour un paramètre de régularisation donné afin de voir si l'algorithme FISTA est effectuement plus rapide que l'algorithme "classique". On obtient alors ce genre de courbe (pour A aléatoire de taille (400,200) , b de taille 200, en utilisant le critère d'arrêt basé sur la norme 2 de la différence $u_{k+1} - u_k$, un $\lambda = 10$ et un critère d'arrêt fixé à 10^-7.

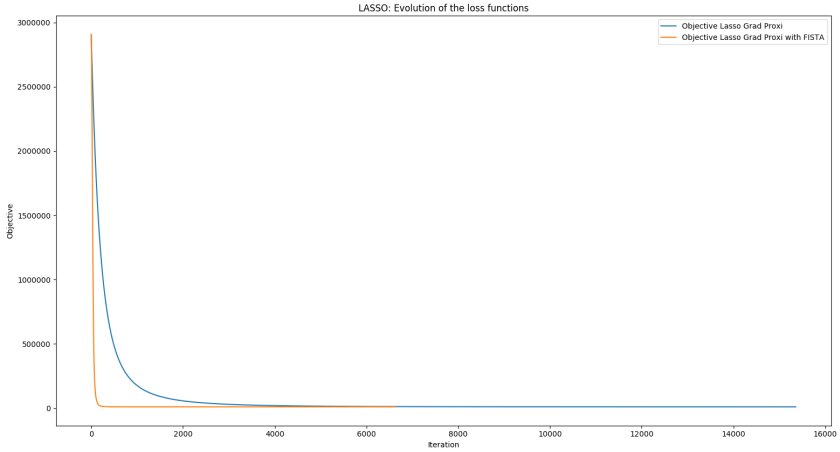

Figure 1: Evolution des fonctions objectifs en fonction des itérations

On observe que la courbe résultant de l'algorithme FISTA converge bien plus tôt que la version "classique". On voit qu'effectivement FISTA permet une accélaration significative de l'algorithme du gradient proximal tout en permettant d'obtenir un résultat équivalent. Ceci correspond bien aux attentes qu'on avait.

5.2 Evolution des résultats en fonction du choix du critère d'arrêt

Comme on l'a décrit précédemment, on laisse en option dans notre algorithme la possibilité de choisir entre deux fonctions objectifs. Afin de bien voir l'effet qu'a le critère d'arrêt, on dresse un tableau qui montre les résultats optimaux ainsi que le nombre nécéssaire d'itérations avant convergence. On a choisit de fixer une matrice A de taille 1000×500, et $\lambda = 1000$, afin de bien mesurer les différences entre critères d'arrêt :

Critère d'arrêt	Valeur du critère	Fonction Obj. Optimale	Nombres d'itérations
$\|u_{k+1} - u_K\|_2$	$1e^{-6}$	220573	41695
$\|u_{k+1} - u_K\|_1$	$1e^{-6}$	220565	78652
$\|u_{k+1} - u_K\|_2$	$1e^{-5}$	221177	17001
$\|u_{k+1} - u_K\|_1$	$1e^{-5}$	220567	48691
$\|(F+G)(u_{k+1}) - (F+G)(u_K)\|_1$	0.1	221063	18014

Comme le montre le tableau ci-dessus, la valeur optimale ne change pas avec le choix d'un différent critère d'arrêt. Choisir la norme 1, la norme 2 ou l'évolution de la fonction objectif, conduit plus ou moins aux mêmes résultats. Par contre, ce qui change, c'est le temps nécessaire pour effectuer les calculs. En effet, choisir l'évolution de la fonction objectif comme critère de convergence demande beaucoup plus de calculs et d'itérations (ce qui croît avec la taille de la matrice), que le critère de convergence correspondant aux composantes optimales u.
On rappelle toutefois, que la valeur du critère d'arrêt (10^{-6}, 10^{-4}...) dépends principalement du problème et des matrices choisies.

5.3 Evolution des résultats en fonction de l'évolution du poids de la régularisation λ

5.3.1 Tableau de résultats pour des paramètres aléatoires fixés

Afin d'analyser l'impact du paramètre λ sur le problème de minimisation que l'on cherche à résoudre, on tire aléatoirement des matrices A et b de taille et de valeur aléatoire et on regarde l'évolution de la norme 1 du paramètre u en fonction de l'évolution du paramètre λ. Des résultats bien détaillés pour la courbe en bleu sont présentés dans l'annexe.

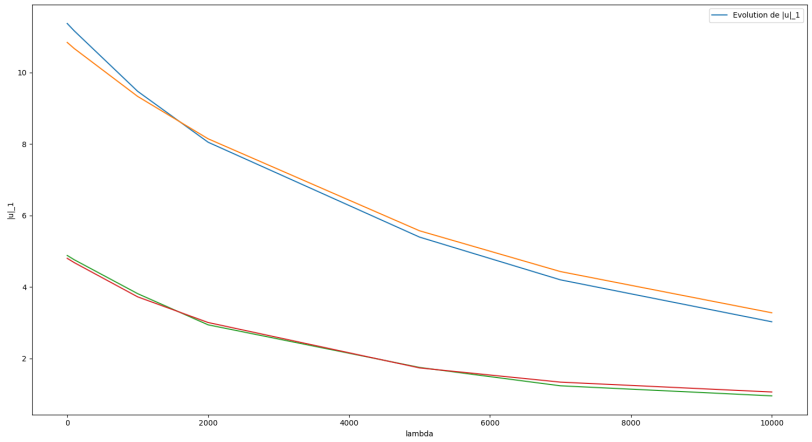

Figure 2: Evolution de la norme 1 du paramètre u

Ces courbes nous permettent de voir que comme on s'y attendait : plus on augmente le paramètre λ plus le terme de régularisation $|u|_1$ prend de l'importance. Néanmoins ce qui nous intéresse réellement c'est de voir si ce terme de régularisation correspond à une bonne approximation de la norme qui vaut la somme des termes non nuls de u. C'est à dire que u devient bien u vecteur creux lorsque λ augmente.

Comme indiqué plus haut, le critère de régularisation λ permet de séléctionner les variables les plus pertinantes, et éviter de distribuer l'erreur sur toutes les composantes. Afin de mesurer cet effet, on dresse

l'histogramme des valeurs optimales de u, en fonction du choix du terme de régularisation :

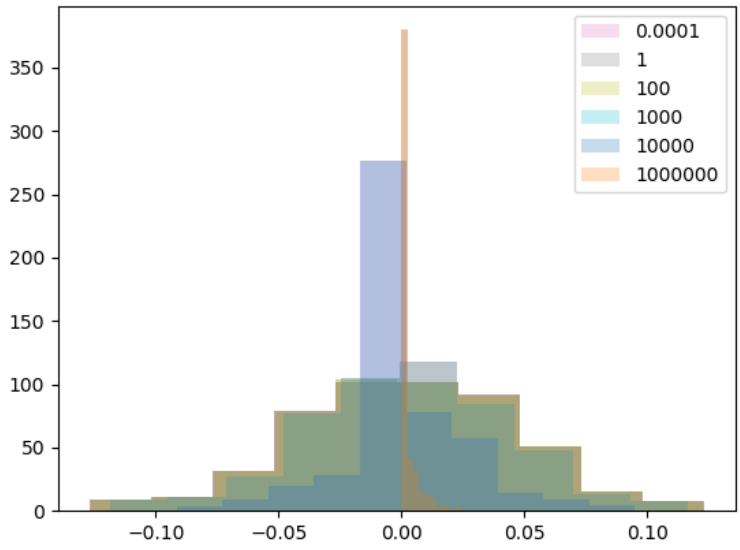

Figure 3: Histogramme des valeurs optimales en fonction de λ en utilisant Grad Proximal

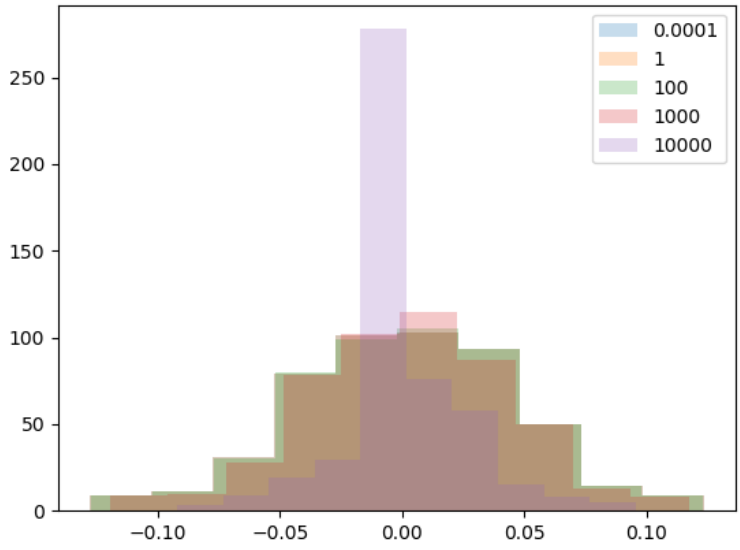

Figure 4: Histogramme des valeurs optimales en fonction de λ en utilisant FISTA

Comme on le remarque sur les deux figures du dessus, et comme on l'a montré théoriquement, le nombre de composantes optimales nulles croît en fonction de λ. En effet, pour les deux algorithmes, les composantes

optimales deviennent plus creuses pour une régularisation importante.

5.4 Sur l'importance de la régularisation

Enfin, on peut voir l'importance de la régularisation en créant un problème artificiel. Pour se faire, nous avons crée une fonction qui génère une matrice A d'une taille donnée et génère un vecteur u^* creux optimal. Etant donné ces deux entités, on crée le vecteur b en faisant :

$$b = A.u^* + \epsilon \qquad \epsilon \rightsquigarrow \mathcal{N}(0, \sigma)$$

On peut donc avec notre algorithme voir en fonction de λ l'écart du u trouvé au $u*$ creux optimal. Afin d'avoir un exemple illustratif, on prend une matrice A de taille 400,200 aléatoire avec des valeurs dans [0,100], un vecteur u^* de taille 200 avec seulement 20 coefficients non nuls, et on prend $\sigma = 40$. On fixe le critère d'arrêt à 10^{-4}. Dans le tableau ci-dessous l'écart à l'optimal "Ecart opt" représente la norme 2 entre la valeur de u trouvé par l'algorithme et u^*.

Lambda	Ecart opt FISTA
0.00	0.323471
0.01	0.323470
0.05	0.323465
0.10	0.323459
0.50	0.323412
1.00	0.323353
5.00	0.322878
10.00	0.322273
100.00	0.313224
1000.00	0.246619
2000.00	0.208394
5000.00	0.157864
7000.00	0.150123
10000.00	0.149147

Comme on peut le voir à travers ce tableau, l'écart à l'optimal décroit avec λ ce qui correspond bien au fait que λ contrôle un terme de régularisation qui sera très utile lorsque u^* est très creux. En effet, puisque b est bruité lorsque λ est petit alors notre algorithme aura tendance à surapprendre et à apprendre sur le bruit. Utiliser ce terme de régularisation a donc tout son intérêt lorsque le prétraitement des données est léger et plusieurs caractéristiques de notre problème s'avèrent être inutiles.

6 Conclusion

Ce premier TP du cours "Optimisation non différentiable et méthodes proximales" nous a permis de mettre en pratique la théorie vue en cours par l'implémentation simple de l'algorithme du LASSO par gradient proximal. On a également pu implémenter la version accélérée FISTA. Tout au long de ce TP, nous avons cherché à comprendre et analyser les différents choix qui peuvent être faits. On a vu que l'algorithme FISTA permet de largement accélérer l'algorithme du gradient proximal que l'on avait initialement. On a vu également que le choix du critère d'arrêt ainsi que du paramètre de régularisation était important et pouvait changer les résultats du problème. Enfin on a terminé par montrer que le terme de régularisation permettait effectivement d'obtenir des coefficients de sortie creux et que cela était très adapté aux problème de la vie réelle.

7 Annexe

On a tiré aléatoirement une matrice A et b dans le problème décrit en 4. On fixe la taille de la matrice A à (400,200) et la taille du vecteur b à 200. Les valeurs de A et b sont compris de manière uniforme entre 0 et 100. Pour ce problème donné, les deux tableaux ci-dessous donnent les valeurs de différentes données finales importantes obtenues.

Notations :

(i) λ : correspond à la valeur du paramètre de régularisation dans 4

(ii) Le préfixe A correspond aux valeurs obtenues en utilisant l'algorithme du gradient proximal dans sa version "classique" et le préfixe B correspond aux valeurs obtenues en utilisant l'algorithme FISTA qui est un algorithme d'accélération de l'algorithme de base.

(iii) Nb it : correspond au nombre d'itérations

(iv) Valeur norm 2 : correspond à la valeur du terme $||Au - b||_2^2$ dans la fonction objectif

(v) Valeur norm 1 : correspond à la valeur du terme $|u|_1$ dans la fonction objectif

(vi) Obj : correspond à la valeur de la fonction objectif finale

(vii) Crit Conv : correspond à la valeur du critère d'arrêt de l'algorithme soit $u^{k+1} - u^k$

λ	Nb it A	Nb it B	Valeur norm 2 A	Valeur norm 2 B	Valeur norm 1 A	Valeur norm 1 B
0.00	38952.0	22499.0	84838.156053	84838.150001	11.369925	11.373083
0.01	38952.0	22499.0	84838.156085	84838.150002	11.369904	11.373062
0.05	38951.0	22499.0	84838.156216	84838.150005	11.369817	11.372975
0.10	38951.0	22499.0	84838.156381	84838.150014	11.369710	11.372868
0.50	38946.0	22499.0	84838.157910	84838.150279	11.368847	11.372005
1.00	38940.0	22499.0	84838.160307	84838.151096	11.367769	11.370927
5.00	38892.0	22497.0	84838.198884	84838.177046	11.359143	11.362300
10.00	38829.0	22494.0	84838.295633	84838.258012	11.348360	11.351517
100.00	38136.0	21176.0	84848.683608	84848.390197	11.162163	11.165038
1000.00	32318.0	17613.0	85759.662244	85757.202837	9.470883	9.473337
10000.00	11613.0	7624.0	113733.979567	113731.744381	3.023406	3.023630

λ	Obj A	Obj B	Crit Conv A	Crit Conv B
0.00	84838.156053	84838.150001	9.998782e-08	9.966955e-08
0.01	84838.269784	84838.263732	9.998588e-08	9.966483e-08
0.05	84838.724706	84838.718654	9.999470e-08	9.964587e-08
0.10	84839.293352	84839.287300	9.998500e-08	9.962206e-08
0.50	84843.842334	84843.836282	9.999009e-08	9.943328e-08
1.00	84849.528075	84849.522023	9.999192e-08	9.919525e-08
5.00	84894.994597	84894.988546	9.998977e-08	9.905085e-08
10.00	84951.779230	84951.773181	9.998841e-08	9.896309e-08
100.00	85964.899890	85964.893976	9.998432e-08	9.959501e-08
1000.00	95230.545314	95230.539855	9.998529e-08	9.973472e-08
10000.00	143968.042962	143968.041104	9.998443e-08	9.980062e-08