Université Joseph Fourier – Grenoble 1

Cours de Mathématiques Appliquées – MAP35G

Année 2015-2016

Travaux Pratiques n° 2

Analyse

I Introduction à l'optimisation pour l'apprentissage

Dans le précédent TP, nous avons observé deux phénomènes gènants liés à la taille des données traitées. D'abord, quand les données sont de taille importante, les algorithmes mis en œuvre pour trouver une solution exacte du problème sont très lents à cause notamment de l'inversion de matrices de grande taille. Aussi, lorsque l'on utilise des modèles trop adaptés au modèle (ex. régression de 100 points par un polynôme de degré 200), le résultat obtenu n'est pas réaliste malgré le nombre de points de données.

Dans ce TP, nous allons considérer une formulation plus réaliste des problèmes d'apprentissage sous la forme d'un problème d'optimisation. Nous étudierons ensuite l'algorithme du gradient, fondamental en apprentissage.

Formellement, nous considérons une variable $y = (y_1, ..., y_M) \in \mathbb{R}^M$ de M mesures liées à une seconde variable $X = (x_1; ...; x_M) \in \mathcal{M}_{M\times d}$ (chacune des M lignes est un vecteur de $\mathbb{R}^d$), on dit que l'on cherche à *expliquer* y par régression par rapport à X.

Comme nous disposons du résultat y des mesures, nous nous plaçons dans le cadre de l'*apprentissage supervisé*. Les vecteurs $x_1, ..., x_M$ sont les *données* (ou variables explicatives), composées de d *features*. Le vecteur y représente les *observations* (ou variables expliquées), si les composantes sont réelles on parle de *régression*, si elles appartiennent à un ensemble fini $\{0, 1, ..., C\}$ on parle de *classification*.

Un objectif classique en apprentissage est de trouver un *prédicateur linéaire* $\alpha \in \mathbb{R}^d$ qui explique le lien entre les données $x_1, ..., x_M$ et les observations $y_1, ..., y_M$; c'est-à-dire tel que le prduit scalaire $\langle \alpha; x_i \rangle$ est proche de y_i pour tout $i = 1, ..., M$. Les *dimensions* importantes du problème sont donc M et d.

Afin de calculer ce prédicateur α, il est d'usage de formuler le problème d'optimisation suivant [1] :

$$\min_{\alpha\in\mathbb{R}^d} \sum_{i=1}^{M} \phi\left(y_i; \langle \alpha; x_i \rangle\right) + \lambda r(\alpha) \tag{1}$$

où $\phi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ est une fonction d'*attache aux données*, convexe, minimale quand $y_i \approx \langle \alpha; x_i \rangle$; λ est un paramètre réel et $r : \mathbb{R}^d \to \mathbb{R}$ est une fonction de régularisation dont nous verrons l'utilité plus tard.

Les fonctions les plus utilisées pour ϕ et r sont :

$\phi(a; b)$		$r(\alpha)$	
$1/2(a-b)^2$	régression quadratique	$\|\alpha\|_1$	régularisation ℓ_1
$\log(1 + \mathrm{e}^{-ab})$	régression logistique	$\|\alpha\|_2^2$	régularisation ℓ_2

1. Cette formulation est largement étudiée dans les domaines de l'optimisation et de l'apprentissage et est toujours au centre de nombreuses recherches actuelles.

1) *Formuler le problème* $(\mathcal{P})$ *considéré dans le TP précédent sous la forme du Problème* (1). *À quoi correspond la dimension* d *? Est-on dans un problème de régression ou de classification ?*

II Algorithme du gradient

Afin de résoudre le Problème (1), il est très courant d'utiliser un algorithme du gradient. L'algorithme du gradient trouve un minimiseur d'une fonction f :

— de classe $\mathcal{C}^1$

— convexe

— de gradient L-Lipschitzien $(\forall x, y, \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|)$

en itérant

$$\alpha^{k+1} = \alpha^k - \gamma \nabla f(\alpha^k) \tag{2}$$

où γ est appelé *pas* du gradient. L'algorithme du gradient converge pour tout pas $\gamma < 2/L$ et $1/L$ est communément utilisé. Si le pas est trop grand, l'algorithme ne converge pas, s'il est trop petit, il met beaucoup de temps à converger ; sa valeur est donc centrale en optimisation.

2) *Montrer que la fonction minimisée dans le TP précédent est de classe* $\mathcal{C}^1$ *et convexe.*

3) *Retrouver l'algorithme de la question 13) du TP précédent.*
Indice : à partir d'une norme $\|x\|$ *sur un vecteur* $x \in \mathbb{R}^n$*, on définit la norme induite* $\|A\|$ *d'une matrice* $A \in \mathcal{M}_{m\times n}$ *comme* $\|A\| = \sup\{\|Ax\|/\|x\| : x \in \mathbb{R}^{n*}\}$

4) *Télécharger le jeu de données* `data_1` *sur* `www.iutzeler.org/docs_teach/MAP35G/data_1.m`*. Écrire une fonction qui normalise les données* $x_1, ..., x_N$ *de* `data_1` *en les ramenant entre* 0 *et* 1 *par transformation affine. Comparer les constantes de Lipschitz des fonctions associées au jeu de données avec et sans normalisation pour la régression polynomiale de degré* 9*. Conclure.*

5) *Implémenter un algorithme du gradient pour la régression polynomiale de degré* 9 *sur ce jeu de données avec et sans normalisation. Observer les régressions obtenues après* 1000 *itérations dans les deux cas. Conclure.*

III Intérêt de la régularisation

6) *Examiner la solution obtenue à la question 5) entre* -0.5 *et* 1.5*. Cette régression vous parait-elle réaliste en pratique ?*

La situation observée est caractéristique d'un *sur-apprentissage* : le modèle généré comprend trop d'information par rapport aux données observées et ne permet pas de généraliser un comportement sous-jacent. Afin d'éviter ce phénomène, il est commun d'ajouter un terme de régularisation dans le problème d'optimisation résolu, typiquement sous la forme d'une norme du descripteur (c'est la fonction r dans la partie I). Cette fonction ajoutée au problème

va modifier la solution en pénalisant les grandes valeurs du descripteur.

a) Régularisation ℓ_2 – *Ridge regression*

Nous allons d'abord considérer le problème de régression polynomiale précédent et ajouter une régulation ℓ_2, c'est-à-dire que nous cherchons la solution du problème suivant :

$$\min_{\alpha \in \mathbb{R}^d} \frac{1}{2}\|X\alpha - y\|_2^2 + \lambda \|\alpha\|_2^2 \tag{3}$$

où λ est un paramètre positif.

7) *Calculer l'algorithme du gradient avec pas $1/L$ pour résoudre le Problème* (3) .

8) *Représenter les solutions obtenues pour une régression polynomiale de degré 9 sur le jeu de données* `data_1` *avec normalisation après* 1000 *itérations pour $\lambda = 0$, $\lambda = 0.001$,$\lambda = 0.028$, et $\lambda = 0.1$. Conclure.*

b) Régularisation ℓ_1 – *Lasso*

Cette fois-ci nous considérons le problème de régression polynomial précédent et ajoutons une régulation ℓ_1, c'est-à-dire que nous cherchons la solution du problème suivant :

$$\min_{\alpha \in \mathbb{R}^d} \frac{1}{2}\|X\alpha - y\|_2^2 + \lambda \|\alpha\|_1 \tag{4}$$

où λ est un paramètre positif.

9) *La fonction à minimiser est-elle toujours de classe $\mathcal{C}^1$? Convexe ?*
Peut-on calculer le gradient de la fonction minimisée ? Sa constante de Lipschitz ?

IV Algorithmes avancés d'optimisation

Nous allons désormais examiner quelques méthodes très utilisées en optimisation afin d'accélérer la vitesse de convergence des algorithmes d'optimisation ou de traiter des cas non-différentiables.

a) Algorithme de Newton

En 1669, Isaac Newton a publié dans *De analysi per aequationes numero terminorum infinitas* la méthode suivante pour trouver un zéro [2] d'une fonction de classe $\mathcal{C}^1$ f de $\mathbb{R} \to \mathbb{R}$. Soit x^k, le point courant ; le développement de Taylor à l'ordre 1 de f en x^k est $f(x) \approx f(x^k) + f'(x^k)(x - x^k)$. Si x est un zéro de f, alors $0 \approx f(x^k) + f'(x^k)(x - x^k)$, donc une nouvelle estimation d'un zéro est obtenue par l'itération

$$x^{k+1} = x^k - \frac{1}{f'(x^k)} f(x^k).$$

2. c'est-à-dire un point x tel que $f(x) = 0$.

10) *Soit f de classe $\mathcal{C}^2$ de $\mathbb{R}^n \to \mathbb{R}$. Justifier brièvement que l'algorithme*

$$x^{k+1} = x^k - [H(f)(x^k)]^{-1}\nabla f(x^k)$$

où $H(f)(x)$ est la Hessienne de f en x, estime un minimum de f. Quelles conditions doit vérifier f ?

11) *Calculer l'algorithme de Newton pour le problème de la Section III-a. Comparer les vitesses des algorithmes du gradient et de Newton en traçant $f(x^k)$ en fonction du nombre d'itérations k (Observer les résultats pour 10 itérations). Conclure.*

b) Gradient proximal

Lorsque les fonctions traitées ne sont pas de classe $\mathcal{C}^1$, il est habituel d'utiliser des gradients *implicites*, encore appelés *opérateurs proximaux*. Ce type d'opérations a été popularisé dans les années 1950/1960, notamment par les analystes français dont Jean-Jacques Moreau (*Fonctions convexes duales et points proximaux dans un espace hilbertien*, 1962). Ces opérations sont toujours le sujet de nombreuses recherches actuelles notamment pour des problèmes d'apprentissage.

Soit f une fonction convexe. L'opérateur proximal de f en x s'écrit

$$\text{prox}_f(x) = \arg\min_u \{f(u) + \frac{1}{2}\|x-u\|_2^2\}.$$

Heureusement, cette opération se simplifie pour des fonctions bien choisies comme les projections ou ... la norme ℓ_1 :

$$\left[\text{prox}_{\lambda\|\cdot\|_1}(x)\right]_i = \left[\arg\min_u \{\lambda\|u\|_1 + \frac{1}{2}\|x-u\|_2^2\}\right]_i = \begin{cases} x_i - \lambda & \text{si } x_i > \lambda \\ 0 & \text{si } |x_i| \leq \lambda \\ x_i + \lambda & \text{si } x_i < -\lambda \end{cases},$$

cette opération s'appelle *seuillage doux* (Soft-thresholding).

Finalement, si l'on cherche à minimiser $f + g$ où f est convexe de classe $\mathcal{C}^1$, de gradient L-Lipschitzien et g est convexe ; on utilise généralement l'algorithme du *gradient proximal* :

$$x^{k+1} = \text{prox}_{g/L}(x^k - \frac{1}{L}\nabla f(x^k)).$$

12) *Calculer l'algorithme du gradient proximal pour le problème de la Section III-b.*

13) *Appliquer l'algorithme du gradient proximal pour le problème de la Section III-b avec $\lambda = 0.01$, $\lambda = 1$, $\lambda = 10$, et $\lambda = 100$. Qu'observe-t-on sur la solution trouvée ? Représenter les polynômes obtenus et conclure.*