

# Cours Apprentissage - ENS Math/Info

## Optimisation Convexe

Francis Bach

24 octobre 2014

Ce cours s'appuie sur le livre "Convex Optimization" de Stephen Boyd et Lieven Vandenberghe (disponible gratuitement : <http://www.stanford.edu/~boyd/cvxbook/>) et les livres "Non-linear programming" de Dimitri Bertsekas (Athena Scientific), et "Introductory lectures on convex optimization : A basic course" de Yurii Nesterov (Kluwer Academic Publishers).

### 1 Boîte à outil générique

- Utile pour la programmation linéaire, i.e.,  $\min_{Ax=b, x \geq 0} c^\top x = \max_{A^\top y \leq c} b^\top y$
- Optimisation à moyenne échelle (pas plus de dizaines de milliers de contraintes et/ou variables)
- Voir <http://cvxr.com/cvx/>

### 2 Méthode de l'ellipsoïde

- Minimisation d'une fonction convexe dérivable sur  $\mathbb{R}^n$  (extensions possibles aux cas non-différentiables et contraints).
- Complexité théorique polynômiale (résultat très général), peu utilisé en pratique.
- **Algorithme** :
  - Initialisation : ellipsoïde  $\{(x - x_0)^\top P_0^{-1}(x - x_0) \leq 1\}$  qui contient  $x_*$  (avec typiquement  $P_0 = R^2 I$ )
  - Pour  $t \geq 0$ , construire l'ellipsoïde de volume minimum qui contient  $\{(x - x_t)^\top P_t^{-1}(x - x_t) \leq 1\}$  et le demi-plan  $\{f'(x_t)^\top (x - x_t) \leq 0\}$ .
  - Ceci correspond à  $x_{t+1} = x_t - \frac{1}{n+1} P_t g_t$  et  $P_{t+1} = \frac{n^2}{n^2-1} (P_t - \frac{2}{n+1} P_t g_t g_t^\top P_t)$  avec  $g_t = \frac{1}{\sqrt{f'(x_t)^\top P_t f'(x_t)}} f'(x_t)$ .  
Voir <http://www-math.mit.edu/~goemans/18433S09/ellipsoid.pdf> et <http://sma.epfl.ch/~eisenbra/OptInFinance/Slides/ellipsoid.pdf>
- **Propriété** : le volume de  $P_{t+1}$  est plus petit que  $e^{-1/2n}$  fois le volume de  $P_t$ .
- **Conséquence** : si  $f$  est  $B$ -Lipschitz, alors la précision  $\varepsilon$  est atteinte après au plus  $2n^2 \log(RG/\varepsilon)$ .  
Voir <http://www.stanford.edu/class/ee392o/elp.pdf>.

### 3 Descente de gradient

- Minimisation d'une fonction convexe dérivable sur  $\mathbb{R}^n$
- **Algorithme** :
  - Initialisation :  $x_0 \in \mathbb{R}^n$ ,
  - Pour  $t \geq 0$ ,  $x_{t+1} = x_t - \gamma_t f'(x_t)$ .
- **Valeurs de pas  $\gamma_t$**  :
  - Pas constant :  $\gamma_t = 1/L$  où  $L$  est une borne supérieure uniforme de la plus grande valeur propre de la Hessienne de  $f$
  - "Line search" : optimisation totale ou partielle par rapport à  $\gamma$ . Voir par exemple <http://www-personal.umich.edu/~mepelman/teaching/IOE511/Handouts/511notes07-5.pdf> pour la règle dite d'Armijo.
- **Analyse théorique**
  - Hypothèses pour résultat théorique simple :  $f$  deux fois dérivable *convexe*,  $LI \succcurlyeq f''(x) \succcurlyeq \mu I$
  - $x_{t+1} - x_* = x_t - x_* - \gamma_t f''(y_t)(x_t - x_*)$  for some  $y_t \in [x_t, x_*]$
  - Pour  $\gamma_t = 1/L$ , on peut montrer que  $\|x_{t+1} - x_*\|^2 \leq (1 - \frac{\mu}{L})\|x_t - x_*\|^2 \leq (1 - \frac{\mu}{L})^{2(t+1)}\|x_0 - x_*\|^2$
  - La convergence est alors dite linéaire
- Cadre convexe : convergence vers un minimum global
- Cadre non convexe : convergence vers un point stationnaire
- Critère d'arrêt : gradient de norme inférieure à  $\varepsilon$

### 4 Méthode de Newton

- Minimisation d'une fonction convexe deux fois dérivable sur  $\mathbb{R}^n$
- **Principe** : minimiser l'approximation quadratique autour de  $x_t$
- **Algorithme** :
  - Initialisation :  $x_0 \in \mathbb{R}^n$ ,
  - Pour  $t \geq 0$ ,  $x_{t+1} = x_t - f''(x_t)^{-1} f'(x_t)$ .
- **Convergence quadratique** (cas convexe) :  $\|x_{t+1} - x_*\|^2 \leq C(\|x_t - x_*\|^2)^2$