Master MIMSE - Année 2

Optimisation Quadratique

Optimisation quadratique avec contraintes

Méthodes numériques

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Algorithmes “extérieurs”

- On veut optimiser une fonction sous des contraintes.
- Si à l’optimum aucune contrainte n’est active, la solution aurait pu être trouvée au moyen d’une méthode d’optimisation sans contrainte.
- Sinon, à l’optimum une ou plusieurs contraintes sont actives.
- PL : identifier les contraintes actives mène à l’optimum.
- en général “dans un coin” ou facette optimale.
- PNL : optimum dans un coin, sur toute une “facette” ou au milieu d’une facette.
- Algos “extérieurs” qui “se baladent” le long de la frontière du domaine,
- qui “jonglent” avec l’ensemble actif (des contraintes actives) $J(x) = \{j | g_j(x) = 0\}$.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Gradient projeté

- Cette méthode est une méthode qui combine deux idées :
  - une direction de descente (plus forte pente)
  - une direction tangentielle (rester à la frontière).
- Soit un point $\boldsymbol{x}$ et $\boldsymbol{\nabla g_j}$ les gradients des fonctions des contraintes *actives* en $\boldsymbol{x}$.
- On cherche une direction de descente orthogonale aux $\boldsymbol{\nabla g_j}$,
- Projection de $\boldsymbol{-\nabla f}$ sur l'orthogonal de $\boldsymbol{S = Vect\{\nabla g_j\}}$.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Formule de projection

- En un point $\boldsymbol{x}$ sur la frontière,
- une direction $\boldsymbol{s}$ dans $\boldsymbol{S}$ est de la forme

$$\boldsymbol{s} = \sum_{\boldsymbol{j} \in \boldsymbol{J}(\boldsymbol{x})} \boldsymbol{\mu}_j \boldsymbol{\nabla g}_j(\boldsymbol{x}) = \boldsymbol{N\mu},$$

avec

$$\boldsymbol{N} = \left(\gamma_1 \middle| \cdots \middle| \gamma_l \right)$$

où les

$$\gamma_j = \frac{\boldsymbol{\nabla g}_j(\boldsymbol{x})}{||\boldsymbol{\nabla g}_j(\boldsymbol{x})||}$$

.

- Soit $\boldsymbol{g} = \boldsymbol{\nabla f}(\boldsymbol{x})$
- $\boldsymbol{g} = \boldsymbol{r} + \boldsymbol{s}$ avec $\boldsymbol{s} \in \boldsymbol{S}$ et $\boldsymbol{r} \in \boldsymbol{O}(\boldsymbol{S})$.
- On utilisera $-\boldsymbol{r}$ comme direction.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Projection 2

- On a $\boldsymbol{g = r + N\mu}$
- donc $\boldsymbol{N^T g = N^T r + N^T N\mu}$
- On suppose que les γ sont linéairement indépendants
- (sinon on aurait sélectionné une base de $\boldsymbol{S}$)
- alors $\boldsymbol{N^T N}$ est inversible !
- donc $\boldsymbol{\mu = (N^T N)^{-1} N^T g}$
- d'où

$$\boldsymbol{r = (I - N(N^T N)^{-1} N^T)g.}$$

- $\boldsymbol{r}$ est tangente (dans $\boldsymbol{O(S)}$)
- direction de descente.
- Si $\boldsymbol{r = 0}$?
- Point de KKT

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Algorithme

1. (Initialisation) $\boldsymbol{X_1}$ un point faisable,
2. (Itération k) Calculer $\boldsymbol{g = \nabla f(X_k)}$
3. Calculer $\boldsymbol{d_k}$ la projection sur $\boldsymbol{O(S_k)}$ (si pas de contraintes actives, on garde le gradient)
4. Si $\boldsymbol{d_k = 0}$ une relation de KKT est vérifiée
   (a) Si les $\boldsymbol{\mu_i}$ satisfont KKT : fini,
   (b) Sinon prendre un $\boldsymbol{\mu_i}$ de mauvais signe, enlever $\boldsymbol{i}$ de $\boldsymbol{J(x)}$ et réitérer.
5. Maximiser le long de la direction, tout en restant faisable,
6. $\boldsymbol{X_{k+1}}$
7. Réitérer
8. Si on sort du domaine faisable, calculer une direction de retour
   (a) $\boldsymbol{w}$ vecteur du viol des contraintes,
   (b) direction de retour $\boldsymbol{s = N(N^T N)^{-1} w}$

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Remarques

- Cette méthode marche très bien pour des contraintes linéaires.
- Peut résoudre un PL de façon analogue au simplexe.
- Mais peut aussi trouver l'optimum pour une fonction objectif non linéaire.
- Appliquée à des contraintes quadratiques, marche encore assez bien
- mais est pénalisée par les "sauts de faisabilité"
- en fait "linéarise" par morceaux la frontière
- Le calcul de $(\boldsymbol{N}^{\boldsymbol{T}}\boldsymbol{N})^{-1}$ à chaque itération, est une faiblesse de la méthode,
- en fait on ajoute ou on retranche des hyperplans de l'ensemble actif
- Il existe des relations de récurrence pour *mettre à jour* la matrice plutôt que de la recalculer.
- Autres directions à projeter que $-\boldsymbol{\nabla f}$?

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Quasi-Newton avec contraintes d'égalité

- On peut incorporer les contraintes d'égalité linéaires dans une méthode DFP
- Contraintes d'inégalité actives $\leftrightarrow$ égalités
- Contraintes d'inégalité non actives $\leftrightarrow$ ignorées
- Si on n'a qu'une contrainte d'égalité $\boldsymbol{a^T x = b}$
- En l'absence de contraintes, on a trouvé une direction $\boldsymbol{d_k = -H_k g_k}$ par DFP, DFP complémentaire ou gradients conjugués.
- La "vraie" direction de recherche devient

  $$\boldsymbol{d_{k1} = -H_{k1} g_k}$$

  avec

  $$\boldsymbol{H_{k1} = H_k - \frac{H_k a a^T H_k}{a^T H_k a}}$$

- Cette direction permet de vérifier la contrainte tout le long de la recherche linéaire :

$$\begin{aligned} \boldsymbol{a^T(X_k + \rho d_{k1})} &= \boldsymbol{a^T X_k - \rho a^T \left(H_k - \frac{H_k a a^T H_k}{a^T H_k a} \right)} \\ &= \boldsymbol{b - \rho \left(a^T H_k - \frac{a^T H_k a a^T H_k}{a^T H_k a} \right)} \\ &= \boldsymbol{b.} \end{aligned}$$

- Les directions $\boldsymbol{d_{k1}}$ ont encore la propriété d'être mutuellement conjuguées,
- donc si $\boldsymbol{f}$ est quadratique, termine en $\boldsymbol{n - 1}$ itérations.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

m contraintes d'égalité

- On range les vecteurs des contraintes d'égalité comme colonnes d'une matrice $\boldsymbol{A}$ telle que :

$$\boldsymbol{A^T x = b.}$$

- La direction de recherche est donnée par une formule analogue :

$$\boldsymbol{d_{km} = -H_{km} g_k}$$

avec

$$\boldsymbol{H_{km} = H_k - H_k A (A^T H_k A)^{-1} A^T H_k}$$

- Cette direction permet de vérifier la contrainte tout le long de la recherche linéaire :

$$\begin{aligned} \boldsymbol{A^T(X_k + \rho d_{km})} &= \boldsymbol{A^T X_k} \\ &\quad \boldsymbol{-\rho A^T (H_k - H_k A (A^T H_k A)^{-1} A^T H_k)} \\ &= \boldsymbol{b - \rho A^T H_k} \\ &\quad \boldsymbol{+\rho A^T H_k A (A^T H_k A)^{-1} A^T H_k} \\ &= \boldsymbol{b.} \end{aligned}$$

- Encore une fois, terminaison quadratique car directions conjuguées.
- On peut calculer $\boldsymbol{H_{km}}$ en calculant des $\boldsymbol{H_{k1}}$ $\boldsymbol{m}$ fois, les uns à la suite des autres (pour $\boldsymbol{a_1}$ puis $\boldsymbol{a_2}$ puis $\boldsymbol{a_3}$ etc.)
- Evite de calculer $\boldsymbol{(A^T H_k A)^{-1}}$.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Inégalités linéaires

- A $\boldsymbol{X_k}$ inégalités actives traitées comme des égalités
- inégalités non actives ignorées.
- Si $\boldsymbol{X_k + \rho^* d_k}$ viole une inégalité non active en $\boldsymbol{X_k}$ on doit calculer $\boldsymbol{X_{k+1}}$ intersection de la droite de recherche et de la contrainte.
- Il faut alors incorporer cette nouvelle contrainte active dans le calcul de $\boldsymbol{H_{k+1,m+1}}$.
- De même, on peut être amené à relâcher une contrainte pour chercher dans une direction où elle est non active ("changement de direction dans un coin") :

$$\boldsymbol{H_k = H_{k1} + \frac{a a^T}{a^T a}}$$

si on n'a qu'une égalité en tout, ou si on relâche *une* contrainte $\boldsymbol{a_r}$ parmi $\boldsymbol{q}$ formant la matrice $\boldsymbol{A}$:

$$\boldsymbol{H_{k,q-1} = H_{k,q} + \frac{P_{q-1} a_r a_r^T P_{q-1}}{a_r^T P_{q-1} a_r}}$$

avec

$$\boldsymbol{P_{q-1} = I - A_{q-1}(A_{q-1}^T A_{q-1})^{-1} A_{q-1}^T}$$

où $\boldsymbol{A_{q-1}}$ est la matrice $\boldsymbol{A}$ privée de la $\boldsymbol{r}$ème colonne $\boldsymbol{a_r}$.

- Plusieurs contraintes à enlever en même temps : on peut faire l'opération successivement pour chaque contrainte.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Revoilà Lagrange...

- Quand enlever une contrainte et laquelle ?
- A chaque point, calculer les coefficients de Lagrange

  $$\boldsymbol{g_k = A\lambda}$$

  donc

  $$\boldsymbol{\lambda = (A^T A)^{-1} A^T g_k}$$

  pour les contraintes actives.
- Si (pour un $\mathbf{min}$) des $\boldsymbol{\lambda}$ sont négatifs,
- on relâche la contrainte correspondant au $\boldsymbol{\lambda}$ le plus négatif.

- Cette méthode revient à jongler avec les contraintes actives comme le gradient projeté.
- On traite des contraintes non linéaires en les approchant localement par un hyperplan tangent et avec des directions de retour.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Algorithmes de points intérieurs

- Jusqu'ici, on se balade sur la frontière du domaine.
- Par contraste, les algorithmes de points intérieurs restent à l'intérieur du domaine des contraintes d'inégalité,
- et s'approchent de la frontière sans jamais l'atteindre (fonctions barrière)
- donc, pas de relaxation
- ou sont dissuadés de sortir trop du domaine faisable (pénalités simples)
- donc, relaxation.

- On supposera que

  $\exists \boldsymbol{x}$ tel que $\boldsymbol{g(x) < 0}$ pour toute contrainte $\boldsymbol{g(x) \leq 0}$.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Fonctions de pénalité

- Idée : trouver une fonction $\boldsymbol{P(x)}$
  telle que l'optimum de

  $$\mathbf{min}\ \boldsymbol{f(x)},\ \boldsymbol{g_k(x)} \leq \mathbf{0}$$

  soit proche de l'optimum de

  $$\mathbf{min}\ \boldsymbol{f(x)} + \boldsymbol{P(x)}.$$

- Ne pas perturber $\boldsymbol{f}$ quand on est à la frontière,
- Pénaliser fortement quand on viole une contrainte.
- Le lagrangien

  $$\boldsymbol{f(x)} + \sum \boldsymbol{\lambda_k g_k(x)}$$

  est une forme *très* particulière de pénalité.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Exemples de fonctions de pénalité usuelles

- Pour confiner une variable x dans $[-X, X]$:
  - $P(x) = k\left(\frac{x}{X}\right)^{2M}$ avec $k > 0$ et M entier non nul.
  - $P(x) = -k \ln\left(-\cos\frac{\pi x}{X}\right)$
- Pour confiner x dans $[X_1, X_2]$:
  - $P(x) = k\left(\frac{2x-(X_1+X_2)}{X_2-X_1}\right)^{2M}$ avec $k > 0$ et M entier non nul.
  - $P(x) = -k \ln\left(-\cos\frac{\pi[2x-(X_1+X_2)]}{X_2-X_1}\right)$
- Inégalités $g_i(x) \geq 0$
  - $P(x) = \sum_i k_i [g_i(x)]^2$ avec $k_i > 0$
  - $P(x) = \sum_i k_i [g_i(x)]^{2M}$ avec $k_i > 0$

  la somme étant prise sur les contraintes actives
- Egalités $g_i(x) = 0$
  - *idem* que les inégalités
  - $P(x) = \sum_i k_i |g_i(x)|$ avec $k_i > 0$

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Remarques

- Méthode simple,
- assez efficace,
- attention : on peut perturber fortement dans l'intérieur donc rester proche de la frontière
- de même perturbations parfois chaotiques "loin" à l'extérieur
- faire des petits pas ?
- Problème sans contraintes à résoudre par une technique d'optimisation sans contrainte.
- Mais fortement non linéaire, parfois même pas quadratique.
- Une contrainte est remplacée par une "crevasse"
- donc **attention** la méthode de plus forte pente risque de mal marcher.
- Efficacité dépend beaucoup des ajustements des k_i (qui peuvent varier d'une itération à l'autre).

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Ajustement des paramètres

- En posant

$$Z(x) = f(x) + \sum_i k_i g_i(x)^2$$

  (contraintes d'égalité ou inégalités actives)

- on cherchera, quand on s'approche de l'optimum, à vérifier

$$\sum_j \left(\frac{\partial f}{\partial x_j} + 2 \sum_i k_i g_i \frac{\partial g_i}{\partial x_j} \right) \frac{\partial g_r}{\partial x_j} = 0.$$

- on résout ce système pour trouver les k_i.
- On peut aussi utiliser $k'_i = \frac{|g_i|}{\varepsilon_i} k_i$ avec ε_i une certaine tolérance.
- En principe, k_i augmente à chaque itération
- $g_i \to 0$ quand $k_i \to +\infty$.
- Coûteux d'évaluer k_i ? Tâtonnements !

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Fonctions barrières

- Ajouter une pénalité qui empêche même d'atteindre la frontière.
- On part d'un point strictement faisable,
- on résout le problème perturbé sans contrainte,
- on a le nouveau point courant,
- on change les paramètres pour s'autoriser à approcher de la frontière
- on réitère.

- 2 types de fonctions couramment utilisées pour remplacer $g(x) \leq 0$:
  - Fonctions inverses : $\frac{1}{g(x)}$,
  - Fonctions logarithmiques : $\log |g(x)|$.
- A chaque itération on doit rester faisable donc si on sort du domaine, on réajuste le pas.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Méthode SUMT

- *Sequential Unconstrained Minimization Technique*, Carroll et Fiacco-McCormick.
- Pour une minimisation sous contraintes d'inégalité $g_i(x) \geq 0$.
- Algorithme :
  1. Point initial strictement faisable X_1
  2. Déterminer des paramètres w_i et r_1
  3. Itération courante :
     (a) X_{k+1} minimise

$$Z_k(x, r_k) = f(x) + r_k \sum_i \frac{w_i}{g_i(x)}.$$

     en partant de X_k
     (b) Calculer r_{k+1}
     (c) Si pas critère d'arrêt (ex. lié à la précision), reboucler.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Convergence

- La méthode converge vers un unique minimum si on a :
  - L'ensemble $\boldsymbol{R_0 = \{x : g_i(x) > 0}$ est non vide
  - $\boldsymbol{f}$ et $\boldsymbol{-g_i}$ sont $\boldsymbol{c^2}$ (OK) et convexes
  - Pour tout $\boldsymbol{K}$ fini, l'ensemble $\boldsymbol{\{x \in R : f(x) \leq K\}}$ est borné avec $\boldsymbol{R}$ fermeture de $\boldsymbol{R_0}$.
  - pour tout $\boldsymbol{r_k > 0}$, $\boldsymbol{Z_k(x, r_k)}$ est strictement convexe.
- En pratique, converge souvent même si certaines de ces conditions sont violées !

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Choix des paramètres

- Les w_i sont calculés une fois pour toutes,
- ils déterminent l'importance relative de chaque contrainte
- "facteurs d'échelle".
- Choisir w_i pour que les $w_i/g_i(x)$ soient du même ordre de grandeur.
- Les r_k déterminent de combien on peut s'approcher de la frontière
- $r_k \to 0$ permettant $g_i(x) \to +\infty$
- La valeur de r_1 est assez importante et détermine le temps de calcul de l'algorithme,
- le schéma de décroissance des r_k importe peu :
- plus les r_k convergent vite vers 0,
- moins il y a d'itérations,
- mais plus les calculs sont longs à chaque itération.
- En pratique, $r_{k+1} = r_k/10$.
- On recommande :

$$r_1 = \left(\frac{\nabla f(x_1)^T [\nabla^2 P(x_1)]^{-1} \nabla f(x_1)}{\nabla P(x_1)^T [\nabla^2 P(x_1)]^{-1} \nabla P(x_1)} \right)^{1/2}$$

- ou par tâtonnements,
- en général $0,5 \leq r_1 \leq 50$.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --

Remarques

- Méthode assez efficace,
- La précision des calculs joue un rôle vraiment important.
- Optimisation très non linéaire.
- On peut incorporer des contraintes d'égalité sous forme de pénalités simples :

$$r_k^{-1/2} \sum_i [g_i(x)]^2.$$

- Vers la fin, on voit très vite quelles sont les contraintes actives à l'optimum donc on peut arrêter les calculs et prendre une autre méthode.

DRAFT -- DRAFT -- DRAFT -- DRAFT -- DRAFT --