Optimization Toolbox - Capitolul II

*Acest **curs** prezinta **Optimization Toolbox - Capitolul II**.*
In acest PDF poti vizualiza cuprinsul si bibliografia (daca sunt disponibile) si aproximativ doua pagini din documentul original.
*Arhiva completa de pe site contine 20 fisiere, intr-un numar total de **22 pagini**.*
Fisierele documentului original au urmatoarele extensii: doc, m.

Extras

Cele mai mici patrate neliniare cu întreg. Modelul Jacobian de raritate.

Modelele de dimensiuni mari în lsqlin, lsqcurvefît sl fsolve pot fi folosite în problemele de mici dimensiuni pâna la probleme de dimensiuni medii fara a calcula Iacobinul în fun sau formarea modelului Jacobian de raritate. (Acest exemplu se apica de asemenea în care se folosesc frnincon sau fminunc fara a calcula Hessianul sau furnizarea modelului Hessian de raritate). Cât de mica este dimensiunea mica pâna la cea medie? Nu este disponibil nici un raspuns absolut, asa cum aceasta depinde de cantitatea virtuala a memoriei în configuratia sistemului calculator. Sa presupunem ca problema noastra are m ecuatii si n necunoscute. Daca comanda J=sparse(ones(m, n)) cauzeaza o eroare de depasire a memoriei calculatorului tau atunci aceasta problema este cu siguranta prea mare (depasita). Daca nu rezulta eroare, problema poate fi în continuare prea mare, dar putem sa aflam rulând si sa vedem daca MATLAB este capabil sa ruleze în cadrul cantitatii de memorie virtuala disponibila pe sistemul tau.

Sa presupunem ca avem o problema cu 10 ecuatii si doua necunoscute, ca de exemplu gaseste valoarea x care minimizeaza :

începând de la punctul x=[0.3, 0.4].

Pentru ca lsqnonlin presupune ca suma patratelor nu este formata explicit în functia folosita, functia introdusa lsqnonlin ar trebui în schimb sa calculeze vectorul functiei evaluate

, pentru k=l pâna la 10 (adica F ar trebui sa aiba k componente)

Pasul 1: Deschide un M-fisier myfun.m care calculeaza valorile functiei obiectiv

function f=myfun(x)

k=1:10;

f=2+2*k-exp(k*x(1))-exp(k*x(2));

Pasul2: Apeleaza programul celor mai mici patrate neliniare

x0=[0.3, 0.4]

[x, resnorm]=lsqnonlin('myfun',x0)

Întrucât Jacobianul nu este calculat în myfun.m (si parametrul Jacobian în options este "off" prin eroare) si nici un model de raritate Jacobian nu este format folosind parametrul JacobPattern în options, lsqnonlin va apela metoda de dimensiuni mari) eroarea pentru ldqnonlin, cu JacobPattern introdus în jstr=sparse(ones(10, 2)). Când programul de diferenta finita este apelat pentru prima data va detecta ca jstr este de fapt o matrice densa, adica nici un beneficiar al vitezei nu este dedus din magazinarea lui ca matrice rara, si ca de acum înainte va folosi jstr=ones(10, 2) (o matrice întreaga) pentru calculele optimizarii.

Dupa 24 de evaluari ale functiei acest exemplu da solutia:

x=

0.2578 0.2578

resnorm

resnorm=

124.3622

Majoritatea sistemelor calculator vor fi capabile sa lucreze cu probleme întregi mult mai mari, pâna la ecuatii si variabile de ordinul 100 .

Daca informatia exista, atunci se poate profita de existenta unei structuri de raritate în Jacobian (sau Hessian), metodele de mari dimensiuni vor fi întotdeauna rulate mai repede daca aceasta informatie este oferita.

Minimizarea neliniara cu gradient si Hessian cunoscut

Acest exemplu implica o rezolvare a unei probleme de minimizare neliniara cu o matrice H(x) Hessian tridiagonala, prima data calculata în mod explicit si apoi prin formarea structurii de raritate Hessian pentru problemele de diferenta finita. Problema este de a gasi x pentru a minimiza:

(1-7)

unde n=1000

Pasul1: Deschide un M-fisier brownfgh.m care calculeaza functia obiectiv, gradientul functiei obiectiv si matricea tridiagonala rara Hessian.

Acest fisier este mai degraba lung si nu este inclus aici. Poti vedea codul cu comanda

. . . .
. . . .
. . . .

Imagini din documentul complet: