I.6. LMI SOLVERS

Didier HENRION
henrion@laas.fr

Belgian Graduate School on
Systems, Control, Optimization and Networks

Leuven - April and May 2010

History

Convex programming

- logarithmic barrier function (Frisch 1955)
- method of centers (Huard 1967)

Interior-point (IP) methods

- ellipsoid algorithm (Khachiyan 1979)

polynomial bound on worst-case iteration count

- IP methods for LP (Karmarkar 1984)

improved complexity bound and efficiency - now about 50% of commercial LP solvers

- self-concordant barrier functions (Nesterov, Nemirovski 1988) - IP methods for general convex programs, in particular SDP and LMI

Logarithmic barrier function

For the optimization problem

$$\begin{array}{ll} \text{min} & g_0(x) \\ \text{s.t.} & g_i(x) \geq 0 \end{array}$$

where the $g_i(x)$ are twice continuously differentiable convex functions, we define the logarithmic barrier function

$$g(x) = -\sum_i \log g_i(x) = \log \prod_i g_i(x)^{-1}$$

which is convex in the interior $g_i(x) > 0$ of the feasible set

Unconstrained optimization

Then we solve the unconstrained convex problem

$$\min g_0(x) + \mu g(x)$$

where $\mu > 0$ and the term $\mu g(x)$ acts as
a repellent of the boundary

The minimum is attained in the interior = interior-point method

Descent methods

To solve an unconstrained optimization problem $\min f(x)$
we produce a minimizing sequence $x_{k+1} = x_k + t_k \Delta x_k$ where
$\Delta x_k \in \mathbb{R}^n$ is the step or search direction and
$t_k \geq 0$ is the step size or step length

A descent method consists in finding a sequence $\{x_k\}$ such that
$f(x^\star) \leq \cdots f(x_{k+1}) < f(x_k)$ where $x^\star$ is the optimum

General descent method

0. given starting point x
1. determine descent direction Δx
2. line search: choose step size $t > 0$
3. update: $x = x + t\Delta x$
4. go to step 1 until a stopping criterion is satisfied

Newton's method

A particular choice of search direction is the Newton step

$$\Delta x = -\nabla^2 f(x)^{-1} \nabla f(x)$$

where $\nabla f(x)$ is the gradient and $\nabla^2 f(x)$ is the Hessian

This step $y = \Delta x$ minimizes the 2nd order Taylor approximation

$$\widehat{f}(x+y) = f(x) + \nabla f(x)^T y + y^T \nabla^2 f(x) y/2$$

and it is the steepest descent direction for the quadratic norm defined by the Hessian

Quadratic convergence near the optimum

Self-concordance

Shortcomings of Newton's method:
- number of required Newton steps

hardly estimated in practice
- analysis depends on used coordinate system

Theory of self-concordant functions:
- number of Newton steps easily estimated
- affine-invariant property

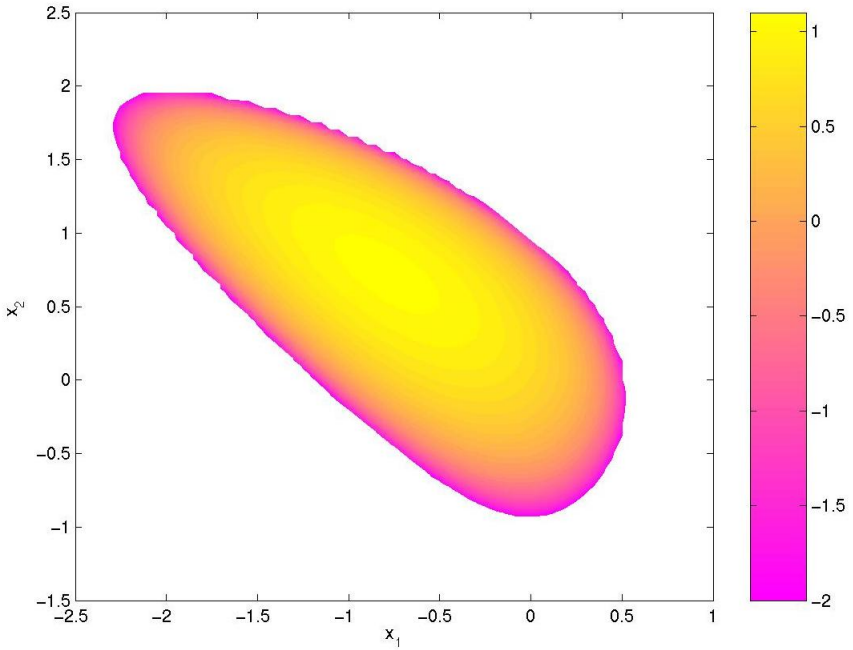
Smooth convex functions with 2nd derivatives Lipschitz continuous with respect to the metric induced by the Hessian: $|f'''(x)| \leq 2f''(x)^{3/2}$, include many logarithmic barrier functions

For LP, QP or SDP 1st and 2nd derivatives of standard self-concordant barriers can be found easily in closed form

Barrier function for an LMI

Given an LMI constraint $F(x) \succeq 0$ we define its logarithmic barrier function

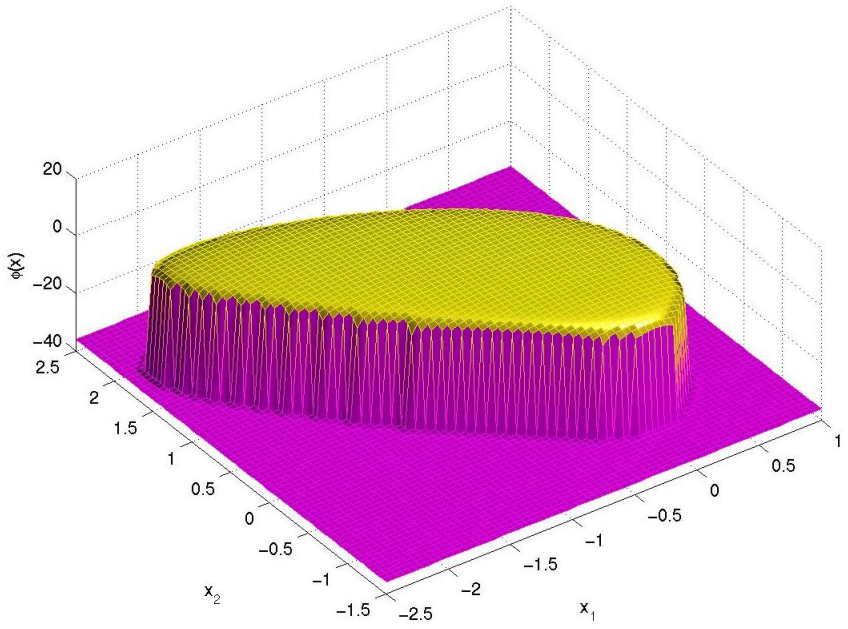
$$f(x) = \log\det F(x)^{-1}$$



This function is analytic, convex and self-concordant on $\{x : F(x) \succ 0\}$, with minimum called the analytic center of the LMI (depends on $F(x)$)

Gradient and Hessian for an LMI

Concave function $-f(x)$ is flat in the interior of the feasible set and sharply decreases toward the boundary



Closed-form expressions for gradient

$$\begin{aligned}(\nabla f(x))_i &= -\text{trace}\, F(x)^{-1}F_i \\ &= -\text{trace}\, F(x)^{-1/2}F_iF(x)^{-1/2}\end{aligned}$$

and Hessian

$$\begin{aligned}(\nabla^2 f(x))_{ij} &= \text{trace}\, F(x)^{-1}F_iF(x)^{-1}F_j \\ &= \text{trace}\left(F(x)^{-1/2}F_iF(x)^{-1/2}\right)\times \\ &\quad \left(F(x)^{-1/2}F_jF(x)^{-1/2}\right)\end{aligned}$$

IP methods for SDP

Primal / dual SDP

$$\begin{array}{ll} \min_X & \text{trace}\, CX \\ \text{s.t.} & \text{trace}\, A_i X = b_i \\ & X \succeq 0 \end{array}$$

$$\begin{array}{ll} \max_y & b^T y \\ \text{s.t.} & Z = C - \sum_i y_i A_i \\ & Z \succeq 0 \end{array}$$

Primal methods

$$\begin{array}{ll} \min_X & \text{trace}\, CX - \mu \log\det X \\ \text{s.t.} & \text{trace}\, A_i X = b_i \end{array}$$

where parameter μ is sequentially decreased to zero and iterates X_k are always primal feasible

Dual methods

$$\begin{array}{ll} \max_{y,Z} & b^T y + \mu \log\det Z \\ \text{s.t.} & Z = C - \sum_i y_i A_i \end{array}$$

where parameter μ is sequentially decreased to zero and iterates y_k, Z_k are always dual feasible

$X_k \succeq 0$ or $Z_k \succeq 0$ ensured via Newton process:

- large decreases of μ require damped Newton steps
- small updates allow full (deep) Newton steps

Primal-dual IP methods for SDP

Primal-dual methods

$$\begin{array}{ll} \min_{x,y,Z} & \text{trace}\, XZ - \mu \log\det XZ \\ \text{s.t.} & \text{trace}\, A_i X = b_i \\ & Z = C - \sum_i y_i A_i \end{array}$$

Minimizers satisfy KKT optimality conditions

$$\begin{array}{rcl} \text{trace}\, A_i X & = & b_i \\ \sum_i y_i A_i + Z & = & C \\ XZ & = & \mu I \\ X, Z & \succeq & 0 \end{array}$$

Duality gap $\text{trace}\, CX - b^T y = \text{trace}\, XZ \succeq 0$ minimized along the central path of solutions $X(\mu)$, $y(\mu)$, $Z(\mu)$, a smooth curve parametrized by scalar μ

For this reason, logarithmic barrier methods are also called path-following methods

Newton step for primal-dual methods

For primal-dual IP methods, primal and dual directions ΔX, Δy and ΔZ satisfy non-linear KKT optimality conditions

$$\begin{aligned} \text{trace}\, A_i \Delta X &= 0 \\ \textstyle\sum_i \Delta y_i A_i + \Delta Z &= 0 \\ (X + \Delta X)(Z + \Delta Z) &= \mu I \end{aligned}$$

Key point is in linearizing and symmetrizing the latter equation

Long list of primal-dual search directions, the most known of which is Nesterov-Todd's

Dynamic updates of μ result in predictor-corrector methods

Initialization

Newton's method needs an initial feasible point

Algorithms that do not require a feasible point are called infeasible-start methods

An elegant approach to bypass the issue of finding a feasible point is to embed the SDP problem in a larger problem which is its own dual (self-dual embedding) and for which a trivial feasible starting point is known

An SDP embedding that proves useful and efficient is the homogeneous embedding (de Klerk/Roos/Terlaky and Luo/Sturm/Zhang)

Drawback: iterates are primal and dual feasible only when converging

Newton step for LMI

For the SDP in LMI form

$$\begin{array}{ll} \text{min} & c^T x \\ \text{s.t.} & F(x) = F_0 + \sum_i x_i F_i \succeq 0 \end{array}$$

the centering problem is

$$\text{min}\, c^T x - \mu \log\det F(x)$$

and at each iteration Newton step Δx satisfies the LSE

$$H \Delta x = -g$$

where gradient g and Hessian H are given by

$$\begin{aligned} H_{ij} &= \text{trace}\, F(x)^{-1} F_i F(x)^{-1} F_i \\ g_i &= c_i/\mu + \text{trace}\, F(x)^{-1} F_i \end{aligned}$$

LSE typically solved via Cholesky or QR

Complexity

For the n-by-n LMI $F(x) \succeq 0$ with m variables, the flop count of IP methods for SDP is as follows:

For each iteration:
(a) $\mathcal{O}(n^2m)$ to form $F(x)$
(b) $\mathcal{O}(n^3m)$ to form $F(x)^{-1}F_iF(x)^{-1}F_j$ (m prods)
(c) $\mathcal{O}(n^2m^2)$ to form $F(x)^{-1}F_i$ (m^2 prods)
(d) $\mathcal{O}(m^3)$ to solve Newton LSE with Cholesky

Dominating terms are (b) and (c) so the complexity for solving one Newton step is $\mathcal{O}(n^3m + n^2m^2)$ but structure can be exploited in these steps !

Number of iterations with Newton's method: $\mathcal{O}(\sqrt{n}\log\varepsilon^{-1})$ where ε is the desired accuracy

In general, it is assumed that $m = \mathcal{O}(n^2)$ otherwise redundant constraints can be removed, so the global worst-case complexity for a dense LMI is

$$\mathcal{O}(n^{6.5}\log\varepsilon^{-1})$$

Much less in practice !

Primal-dual methods

In contrast with purely primal or dual methods,
in primal-dual methods..

- At each step both primal and dual variables
are updated simultaneously
- Search direction obtained from Newton's method applied to
modified KKT equations
- Work when problem is not strictly feasible

Generally for LP, QP or SDP primal-dual
methods outperform barrier methods

IP methods in general

General characteristics of IP methods:

- Efficiency: about 5 to 50 iterations, almost independent of input data (problem), each iteration is a least-squares problem (well established linear algebra)
- Theory: worst-case analysis of IP methods yields polynomial computational time
- Structure: tailored SDP solvers can exploit problem structure

Penalty/augmented Lagrangian methods

Use similar ideas, but cannot be considered as an interior-point method

When applied to LMI problem

$$\min c^T x \text{ s.t. } F(x) = F_0 + \sum_i x_i F_i \succeq 0$$

- penalty method - some eigenvalues of $F(x)$ can be negative
- barrier method - no eigenvalue of $F(x)$ can be negative

Augmented Lagrangian $L(x, Z, p) = c^T x + \text{trace } Z\Phi(x, p)$ with dual variable Z and suitable penalty function, for example $\Phi(x, p) = p^2(F(x) + pI)^{-1} - pI$ with penalty parameter p

Penalty/augmented Lagrangian methods (2)

General algorithm

1. find x_{k+1} such that $\|\nabla_x L(x, Z_k, p_k)\| \leq \epsilon_k$
2. update dual variables: $Z_{k+1} = f(x_{k+1}, Z_k)$
3. update penalty parameter: $p_{k+1} < p_k$
4. go to step 1 until a stopping criterion is satisfied

Can be considered as a primal-dual method, but dual variables are obtained in closed-form at step 2

Complexity roughly same as IP methods, but can be improved to $O(m^2K^2)$ where K is the number of non-zero terms

SDP solvers

Self-dual embedding:
- SeDuMi (Sturm, Terlaky)
- SDPT3 (Toh, Tütüncü, Todd)

Primal-dual path-following predictor-corrector:
- CSDP (Borchers)
- SDPA (Kojima et al)

Dual-scaling path-following:
- DSDP (Benson, Ye, Zhang)

exploits structure for combinatorics

Projective method (project iterate on Dikin ellipsoid within SDP cone)
- LMILAB (Gahinet, Nemirovskii)

exploits linear algebra of control LMI problems

Penalty and augmented Lagrangian:
- PENSDP (Kočvara, Stingl)

Specialized SDP solvers

Parallel implementations:

- CSDP (Borchers)
- SDPA (Kojima et al)

KYP LMIs using SDP duality:

- KYPD (Hansson et al)

Verified SDP using interval arithmetic:

- VSDP (Jansson)

Nonlinear nonconvex SDP solvers

Low-rank LMIs solved as nonconvex QPs:

- SDPLR (Burer, Monteiro)

Low-rank LMIs solved by alternating projections:

- LMIRANK (Orsi et al)

BMIs via penalty and augmented Lagrangian:

- PENBMI (Kočvara, Stingl)

Matlab dependence

The following solvers are Matlab-dependent:

- SeDuMi (Sturm, Terlaky)
- SDPT3 (Toh, Tütüncü, Todd)
- LMILAB (Gahinet, Nemirovskii)
- VSDP (Jansson)
- LMIRANK (Orsi et al)

Most of the other solvers are available under Matlab, but not Matlab-dependent

Matrices as variables

Generally, in control problems we do not encounter the LMI in canonical or semidefinite form but rather with matrix variables

Lyapunov's inequality

$$A^T P + PA \prec 0 \quad P = P^T \succ 0$$

can be written in canonical form

$$F(x) = F_0 + \sum_{i=1}^{m} F_i x_i \succ 0$$

with the notations

$$F_0 = 0 \quad F_i = -A^T B_i - B_i A$$

where B_i, $i = 1, \ldots, n(n+1)/2$ are matrix bases for symmetric matrices of size n

Most software packages for solving LMIs however work with canonical or semidefinite forms, so that a (sometimes time-consuming) pre-processing step is required

Interfaces

Matlab Robust Control Toolbox
- LMILAB (Gahinet, Nemirovski)

originally developed for INRIA's Scilab

Matlab LMI interfaces to SDP solvers
- LMITOOL (Nikoukah, Delebecque, El Ghaoui)
- SeDuMi Interface (Peaucelle)

Matlab convex optimization and modeling
- YALMIP (Löfberg)
- cvx (Grant, Boyd)

Matlab moments and polynomial SOS
- Gloptipoly (Lasserre et al)
- SOSTOOLS (Parrilo et al)
- SparsePOP (Kojima et al)

Python interface
- cvxopt (Vandenberghe et al)

Scilab and NSP
- LMITOOL (Nikoukah, Delebecque, El Ghaoui)
- GloptiPoly (Lasserre et al)
- YALMIP (Löfberg)