Advances in Convex Optimization: Interior-point Methods, Cone Programming, and Applications

Stephen Boyd
Electrical Engineering Department
Stanford University

(joint work with Lieven Vandenberghe, UCLA)

CDC 02 Las Vegas 12/11/02
Easy and Hard Problems
Least squares (LS)

minimize $\|Ax - b\|_2^2$

$A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ are parameters; $x \in \mathbb{R}^n$ is variable

• have complete theory (existence & uniqueness, sensitivity analysis . . .)
• several algorithms compute (global) solution reliably
• can solve dense problems with $n = 1000$ vbles, $m = 10000$ terms
• by exploiting structure (e.g., sparsity) can solve far larger problems

. . . LS is a (widely used) technology
Linear program (LP)

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i, \quad i = 1, \ldots, m
\end{align*}
\]

\(c, a_i \in \mathbb{R}^n\) are parameters; \(x \in \mathbb{R}^n\) is variable

- have nearly complete theory
  (existence & uniqueness, sensitivity analysis . . .)
- several algorithms compute (global) solution reliably
- can solve dense problems with \(n = 1000\) vbles, \(m = 10000\) constraints
- by exploiting structure (e.g., sparsity) can solve far larger problems

. . . LP is a (widely used) technology
Quadratic program (QP)

\[
\text{minimize} \quad \|Fx - g\|_2^2 \\
\text{subject to} \quad a_i^T x \leq b_i, \quad i = 1, \ldots, m
\]

• a combination of LS & LP
• same story . . . QP is a technology
• solution methods reliable enough to be **embedded in real-time control applications** with little or no human oversight
• basis of **model predictive control**
The bad news

- LS, LP, and QP are exceptions

- most optimization problems, even some very simple looking ones, are intractable
Polynomial minimization

minimize $p(x)$

$p$ is polynomial of degree $d$; $x \in \mathbb{R}^n$ is variable

- except for special cases (e.g., $d = 2$) this is a very difficult problem
- even sparse problems with size $n = 20$, $d = 10$ are essentially intractable
- all algorithms known to solve this problem require effort exponential in $n$
What makes a problem easy or hard?

classical view:

- **linear** is easy
- **nonlinear** is hard(er)
What makes a problem easy or hard?

emerging (and correct) view:

... the great watershed in optimization isn’t between linearity and nonlinearity, but convexity and nonconvexity.

— R. Rockafellar, SIAM Review 1993
Convex optimization

minimize \( f_0(x) \)
subject to \( f_1(x) \leq 0, \ldots, f_m(x) \leq 0 \)

\( x \in \mathbb{R}^n \) is optimization variable; \( f_i : \mathbb{R}^n \to \mathbb{R} \) are convex:

\[
f_i(\lambda x + (1 - \lambda)y) \leq \lambda f_i(x) + (1 - \lambda)f_i(y)
\]

for all \( x, y, 0 \leq \lambda \leq 1 \)

- includes LS, LP, QP, and many others
- like LS, LP, and QP, convex problems are fundamentally tractable
Example: Robust LP

minimize \( c^T x \)
subject to \( \text{Prob}(a_i^T x \leq b_i) \geq \eta, \quad i = 1, \ldots, m \)

coefficient vectors \( a_i \) IID, \( \mathcal{N}(\bar{a}_i, \Sigma_i) \); \( \eta \) is required reliability

- for fixed \( x \), \( a_i^T x \) is \( \mathcal{N}(\bar{a}_i^T x, x^T \Sigma_i x) \)
- so for \( \eta = 50\% \), robust LP reduces to LP

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i, \quad i = 1, \ldots, m
\end{align*}
\]

and so is easily solved

- what about other values of \( \eta \), e.g., \( \eta = 10\%? \ \eta = 90\%? \)
Hint

\[ \{ x \mid \text{Prob}(a_i^T x \leq b_i) \geq \eta, \ i = 1, \ldots, m \} \]

\( \eta = 10\% \)
\( \eta = 50\% \)
\( \eta = 90\% \)
That’s right

robust LP with reliability $\eta = 90\%$ is convex, and \textbf{very easily solved}

robust LP with reliability $\eta = 10\%$ is not convex, and \textbf{extremely difficult}

moral: \textbf{very difficult} and \textbf{very easy} problems can look \textbf{quite similar}
(to the untrained eye)
Convex Analysis and Optimization
Convex analysis & optimization

nice properties of convex optimization problems known since 1960s

- local solutions are global
- duality theory, optimality conditions
- simple solution methods like alternating projections

convex analysis well developed by 1970s *Rockafellar*

- separating & supporting hyperplanes
- subgradient calculus
What’s new (since 1990 or so)

- primal-dual interior-point (IP) methods
  extremely efficient, handle nonlinear large scale problems,
  polynomial-time complexity results, software implementations

- new standard problem classes
  generalizations of LP, with theory, algorithms, software

- extension to generalized inequalities
  semidefinite, cone programming

... convex optimization is becoming a technology
Applications and uses

• lots of applications
  \textit{control, combinatorial optimization, signal processing, circuit design, communications, \ldots}

• robust optimization
  \textit{robust versions of LP, LS, other problems}

• relaxations and randomization
  \textit{provide bounds, heuristics for solving hard problems}
Recent history

- 1984–97: interior-point methods for LP
  - 1984: Karmarkar’s interior-point LP method
  - theory Ye, Renegar, Kojima, Todd, Monteiro, Roos, . . .
  - practice Wright, Mehrotra, Vanderbei, Shanno, Lustig, . . .
- 1988: Nesterov & Nemirovsky’s self-concordance analysis
- 1989–: LMIs and semidefinite programming in control
- 1990–: semidefinite programming in combinatorial optimization
  Alizadeh, Goemans, Williamson, Lovasz & Schrijver, Parrilo, . . .
- 1994: interior-point methods for nonlinear convex problems
  Nesterov & Nemirovsky, Overton, Todd, Ye, Sturm, . . .
- 1997–: robust optimization Ben Tal, Nemirovsky, El Ghaoui, . . .
New Standard Convex Problem Classes
Some new standard convex problem classes

- second-order cone program (SOCP)
- geometric program (GP) (and entropy problems)
- semidefinite program (SDP)

for these new problem classes we have

- complete duality theory, similar to LP
- good algorithms, and robust, reliable software
- wide variety of new applications
Second-order cone program

second-order cone program (SOCP) has form

\[
\begin{align*}
\text{minimize} & \quad c_0^T x \\
\text{subject to} & \quad \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m
\end{align*}
\]

with variable \( x \in \mathbb{R}^n \)

- includes LP and QP as special cases
- nondifferentiable when \( A_i x + b_i = 0 \)
- new IP methods can solve (almost) as fast as LPs
Example: robust linear program

minimize \[ c^T x \]
subject to \[ \text{Prob}(a_i^T x \leq b_i) \geq \eta, \quad i = 1, \ldots, m \]

where \( a_i \sim \mathcal{N}(\bar{a}_i, \Sigma_i) \)

equivalent to

minimize \[ c^T x \]
subject to \[ \bar{a}_i^T x + \Phi^{-1}(\eta)\|\Sigma_i^{1/2} x\|_2 \leq 1, \quad i = 1, \ldots, m \]

where \( \Phi \) is (unit) normal CDF

robust LP is an SOCP for \( \eta \geq 0.5 \) (\( \Phi(\eta) \geq 0 \))
**Geometric program (GP)**

**log-sum-exp** function:

\[
\text{lse}(x) = \log (e^{x_1} + \cdots + e^{x_n})
\]

\[
\ldots \text{a smooth } \textbf{convex} \text{ approximation of the } \text{max} \text{ function}
\]

**geometric program:**

\[
\begin{align*}
\text{minimize} & \quad \text{lse}(A_0x + b_0) \\
\text{subject to} & \quad \text{lse}(A_i x + b_i) \leq 0, \quad i = 1, \ldots, m
\end{align*}
\]

\[A_i \in \mathbb{R}^{m_i \times n}, \ b_i \in \mathbb{R}^{m_i}; \text{ variable } x \in \mathbb{R}^n\]
Entropy problems

unnormalized negative entropy is convex function

\[- \text{entr}(x) = \sum_{i=1}^{n} x_i \log(x_i / 1^T x)\]

defined for $x_i \geq 0$, $1^T x > 0$

entropy problem:

minimize $- \text{entr}(A_0 x + b_0)$
subject to $- \text{entr}(A_i x + b_i) \leq 0$, $i = 1, \ldots, m$

$A_i \in \mathbb{R}^{m_i \times n}$, $b_i \in \mathbb{R}^{m_i}$
Solving GPs (and entropy problems)

- GP and entropy problems are **duals** (if we solve one, we solve the other)
- new IP methods can solve large scale GPs (and entropy problems) almost as fast as LPs
- applications in many areas:
  - information theory, statistics
  - communications, wireless power control
  - digital and analog circuit design
CMOS analog/mixed-signal circuit design via GP

given

• circuit cell: opamp, PLL, D/A, A/D, SC filter, . . .
• specs: power, area, bandwidth, nonlinearity, settling time, . . .
• IC fabrication process: TSMC 0.18μm mixed-signal, . . .

find

• electronic design: device $L$ & $W$, bias $I$ & $V$, component values, . . .
• physical design: placement, layout, routing, GDSII, . . .
The challenges

• complex, multivariable, highly nonlinear problem

• dominating issue: robustness to
  – model errors
  – parameter variation
  – unmodeled dynamics

(sound familiar?)
Two-stage op-amp

- **design variables:** device lengths & widths, component values
- **constraints/objectives:** power, area, bandwidth, gain, noise, slew rate, output swing, . . .
Op-amp design via GP

- express design problem as GP
  (using change of variables, and a few good approximations . . . )
- 10s of vbles, 100s of constraints; solution time $\ll 1$sec

**robust** version:

- take 10 (or so) different parameter values (‘PVT corners’)
- replicate all constraints for each parameter value
- get 100 vbles, 1000 constraints; solution time $\approx 2$sec
Minimum noise versus power & BW

Minimum noise in nV/Hz_{0.5}

Power in mW

- \( f_c = 30 \text{MHz} \)
- \( f_c = 60 \text{MHz} \)
- \( f_c = 90 \text{MHz} \)
Cone Programming
Cone programming

general cone program:

minimize \( c^T x \)
subject to \( Ax \preceq_K b \)

- **generalized inequality** \( Ax \preceq_K b \) means \( b - Ax \in K \), a proper convex cone

- LP, QP, SOCP, GP can be expressed as cone programs
Semidefinite program

semidefinite program (SDP):

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad x_1 A_1 + \cdots + x_n A_n \preceq B
\end{align*}
\]

$B, \, A_i$ are symmetric matrices; variable is $x \in \mathbb{R}^n$

- constraint is \textbf{linear matrix inequality} (LMI)
- inequality is matrix inequality, i.e., $K$ is positive semidefinite cone
- SDP is special case of cone program
Early SDP applications

(around 1990 on)

- control (*many*)
- combinatorial optimization & graph theory (*many*)
More recent SDP applications

- structural optimization: *Ben-Tal, Nemirovsky, Kocvara, Bendsoe, ...*
- signal processing: *Vandenberghe, Stoica, Lorenz, Davidson, Shaked, Nguyen, Luo, Sturm, Balakrishnan, Saadat, Fu, de Souza, ...*
- circuit design: *El Gamal, Vandenberghe, Boyd, Yun, ...*
- algebraic geometry: *Parrilo, Sturmfels, Lasserre, de Klerk, Pressman, Pasechnik, ...*
- communications and information theory: *Rasmussen, Rains, Abdi, Moulines, ...*
- quantum computing: *Kitaev, Waltrous, Doherty, Parrilo, Spedalieri, Rains, ...*
- finance: *Iyengar, Goldfarb, ...*
Convex optimization heirarchy

convex problems
cone problems
SDP
SOCP
QP
LP
LS

more general
more specific
Relaxations & Randomization
Relaxations & randomization

convex optimization is increasingly used

• to find good bounds for hard (i.e., nonconvex) problems, via relaxation

• as a heuristic for finding good suboptimal points, often via randomization
Example: Boolean least-squares

Boolean least-squares problem:

\[
\begin{align*}
\text{minimize} & \quad \|Ax - b\|^2 \\
\text{subject to} & \quad x_i^2 = 1, \quad i = 1, \ldots, n
\end{align*}
\]

• basic problem in digital communications
• could check all \(2^n\) possible values of \(x\) . . .
• an NP-hard problem, and very hard in practice
• many heuristics for approximate solution
Boolean least-squares as matrix problem

\[ \|Ax - b\|^2 = x^T A^T Ax - 2b^T Ax + b^T b \]
\[ = \text{Tr} A^T AX - 2b^T A^T x + b^T b \]

where \( X = xx^T \)

hence can express BLS as

\[ \text{minimize} \quad \text{Tr} A^T AX - 2b^T A^T x + b^T b \]
\[ \text{subject to} \quad X_{ii} = 1, \quad X \succeq xx^T, \quad \text{rank}(X) = 1 \]

\[ \ldots \text{still a very hard problem} \]
SDP relaxation for BLS

ignore rank one constraint, and use

\[ X \succeq xx^T \iff \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0 \]

to obtain SDP relaxation (with variables \( X, x \))

\[
\begin{align*}
\text{minimize} & \quad \text{Tr} A^T A X - 2b^T A^T x + b^T b \\
\text{subject to} & \quad X_{ii} = 1, \quad \begin{bmatrix} X & x \\ x^T & 1 \end{bmatrix} \succeq 0
\end{align*}
\]

- optimal value of SDP gives lower bound for BLS
- if optimal matrix is rank one, we’re done
Interpretation via randomization

- can think of variables $X$, $x$ in SDP relaxation as defining a normal distribution $z \sim \mathcal{N}(x, X - xx^T)$, with $\mathbf{E} z_i^2 = 1$
- SDP objective is $\mathbf{E} \|Az - b\|^2$

suggests randomized method for BLS:

- find $X^*$, $x^*$, optimal for SDP relaxation
- generate $z$ from $\mathcal{N}(x^*, X^* - x^*x^T)$
- take $x = \text{sgn}(z)$ as approximate solution of BLS (can repeat many times and take best one)
Example

- (randomly chosen) parameters $A \in \mathbb{R}^{150 \times 100}$, $b \in \mathbb{R}^{150}$
- $x \in \mathbb{R}^{100}$, so feasible set has $2^{100} \approx 10^{30}$ points

**LS approximate solution:** minimize $\|Ax - b\|$ s.t. $\|x\|^2 = n$, then round yields objective 8.7% over SDP relaxation bound

**randomized method:** (using SDP optimal distribution)
- best of 20 samples: 3.1% over SDP bound
- best of 1000 samples: 2.6% over SDP bound
\[ \|Ax - b\| / (\text{SDP bound}) \]
Interior-Point Methods
Interior-point methods

- handle linear and **nonlinear** convex problems *Nesterov & Nemirovsky*
- based on Newton’s method applied to ‘barrier’ functions that trap $x$ in **interior** of feasible region (hence the name IP)
- worst-case complexity theory: \# Newton steps $\sim \sqrt{\text{problem size}}$
- in practice: \# Newton steps between 10 & 50 (!) — over wide range of problem dimensions, type, and data
- 1000 variables, 10000 constraints feasible on PC; far larger if structure is exploited
- readily available (commercial and noncommercial) packages
Typical convergence of IP method

LP, GP, SOCP, SDP with 100 variables
Typical effort versus problem dimensions

- LPs with $n$ variables, $2n$ constraints
- 100 instances for each of 20 problem sizes
- avg & std dev shown
Computational effort per Newton step

- Newton step effort dominated by solving linear equations to find primal-dual search direction
- Equations inherit structure from underlying problem
- Equations same as for least-squares problem of similar size and structure

Conclusion:

we can solve a convex problem with about the same effort as solving 30 least-squares problems
Problem structure

common types of structure:

- sparsity
- state structure
- Toeplitz, circulant, Hankel; displacement rank
- Kronecker, Lyapunov structure
- symmetry
Exploiting sparsity

- well developed, since late 1970s

- direct (sparse factorizations) and iterative methods (CG, LSQR)

- standard in general purpose LP, QP, GP, SOCP implementations

- can solve problems with $10^5$, $10^6$ variables, constraints (depending on sparsity pattern)
Exploiting structure in SDPs

in **combinatorial optimization**, major effort to exploit structure

- structure is mostly (extreme) sparsity
- IP methods and others (bundle methods) used
- problems with $10000 \times 10000$ LMI s, 10000 variables can be solved

*Ye, Wolkowicz, Burer, Monteiro* . . .
Exploiting structure in SDPs

in control

- structure includes sparsity, Kronecker/Lyapunov
- substantial improvements in order, for particular problem classes

Balakrishnan & Vandenberghe, Hansson, Megretski, Parrilo, Rotea, Smith, Vandenberghe & Boyd, Van Dooren, . . .

. . . but no general solution yet
Conclusions
Conclusions

convex optimization

- theory fairly mature; practice has advanced tremendously last decade
- qualitatively different from general nonlinear programming
- becoming a **technology** like LS, LP (esp., new problem classes), reliable enough for embedded applications
- cost only 30× more than least-squares, but far more expressive
- lots of applications still to be discovered
Some references

- Semidefinite Programming, *SIAM Review 1996*
- Applications of Second-order Cone Programming, *LAA 1999*
- Linear Matrix Inequalities in System and Control Theory, *SIAM 1994*
- Lectures on Modern Convex Optimization, *SIAM 2001, Ben Tal & Nemirovsky*
Shameless promotion

Convex Optimization, Boyd & Vandenberghe

- to be published 2003

- good draft available at Stanford EE364 (UCLA EE236B) class web site as course reader