Mathematical optimization techniques and their applications

in the analysis of biological systems

Pablo A. Parrilo

www.mit.edu/~parrilo

Laboratory for Information and Decision Systems
Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Massachusetts Institute of Technology

Outline

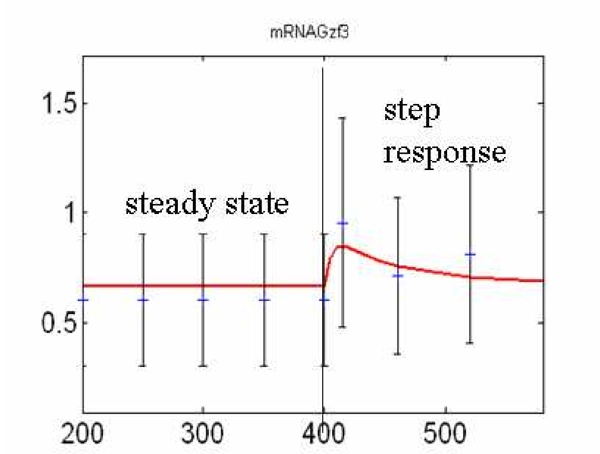
- Two motivating examples:
  - Steady state data validation in mass action kinetics
  - Graphical model selection with hidden variables
- Mathematical optimization
- Taxonomy, features, techniques
- Key desirable aspects: *scalability* and *robustness*
- Convexity
  - Advantages, modeling power
  - Semidefinite programming
- Validation / selection problems: our approach
- Perspectives, limitations, and challenges

Data consistency

Elementary reaction models for gene expression in yeast

$$
\begin{aligned}
\frac{d[TF]}{dt} &= -K_{A,1} \cdot [TF] \cdot [GENE] + K_{D,1} \cdot [TF \bullet GENE] \\
\frac{d[GENE]}{dt} &= -K_{A,1} \cdot [TF] \cdot [GENE] + K_{D,1} \cdot [TF \bullet GENE] \\
\frac{d[TF \bullet GENE]}{dt} &= K_{A,1} \cdot [TF] \cdot [GENE] - K_{D,1} \cdot [TF \bullet GENE] - \\
& -K_{A,2} \cdot [TF \bullet GENE] \cdot [RNAPol] + K_{D,2} \cdot [TF \bullet GENE \bullet RNAPol] + \\
& +K_{TC} \cdot [TF \bullet GENE \bullet RNAPol]
\end{aligned}
$$

- Nonlinear dynamics
- Microarray data of wildtype and mutants
- Steady state + dynamic measurements
- Extract as much information as possible



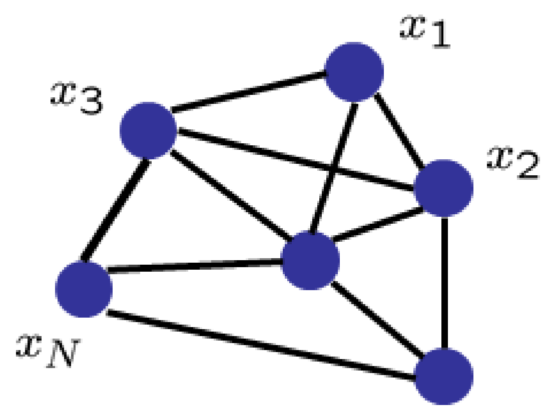
What parameter/rate values are consistent with measurements?

Joint work with L. Küpfer and U. Sauer (ETH Zürich)

Massachusetts Institute of Technology

Graphical model identification

- *Graphical models* are statistical models where conditional independence is modelled by a graph
- "Marriage" of probability and graph theory.
- Very interesting theory and applications.



If edge (i, j) does not exist, then the random variables X_i and X_j are independent, conditioned on all other variables.

Given samples of a collection of random variables, what is the "best" graph that describes the system?

What if only we only have access to a *subset* of the relevant variables?

Joint work with V. Chandrasekaran and A. Willsky (MIT)

Motivation

Different problems, that share common properties.

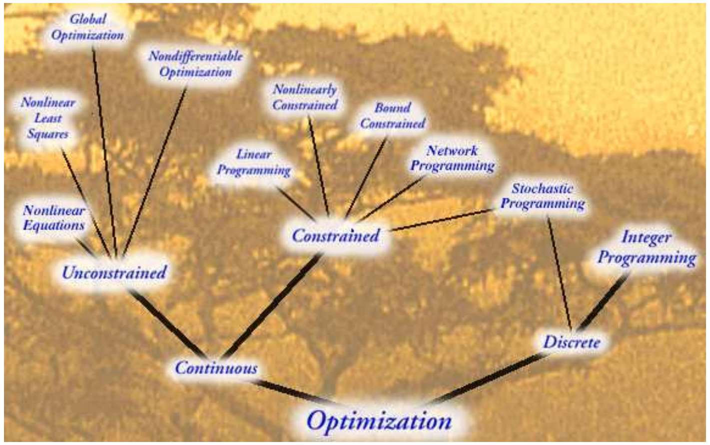
- Can be expressed/approximated in crisp mathematical terms
- Continuous variables, possibly also discrete
- May include dynamical aspects
- Provably difficult (NP-complete, or worse)

The techniques of *mathematical optimization* can be fruitfully applied to many of these problems.

(other names: mathematical programming, optimization, etc.)

Optimization is ubiquitous

- Optimization is essential across many engineering applications (e.g., signal processing, control, routing, VLSI, machine learning, mechanical design, revenue management, etc)
- Often, defines what an "acceptable solution" is
- Enables whole industries (e.g., airlines)



Demand for increasingly sophisticated mathematical programming methods:

- From 1950s on: linear programming, nonlinear, global, convex, quadratic, semidefinite, hyerbolic, etc.
- Combinatorial, network, packing/covering, integer, submodular, etc.

Goal is to develop mathematical infrastructure and associated computational methods for engineering and scientific applications.

Mathematical Optimization

General formulation, that models questions in terms of

- *Decision variables*: values under our control, "knobs" that we can change
- *Constraints*: restrict the values the decision variables can take
- Possibly, *objective function(s)*, whose value we try to optimize

"Typical" form:

$$\text{minimize}\, f_0(x_1, \ldots, x_n) \qquad \text{s.t.} \qquad f_i(x_1, \ldots, x_n) \leq 0 \quad i = 1, \ldots, m$$

The nature of variables x_i and constraints f_i can be quite diverse:

- Variables may represent individual values, or sets, or functions, ...
- Constraints may be deterministic, stochastic, logical, ...

Linear programming (LP)

A well-known and important case, where both the objective and constraints are *affine* functions

$$\text{minimize } c^T x \qquad \text{s.t.} \qquad Ax \leq b$$

- Well-developed, mature theory
- Algebraic and geometry aspects are well understood
- Many applications (e.g., flux balance analysis)
- Efficiently solvable, both combinatorial (simplex) and continuous (interior-point) algorithms

Many other kinds...

Many classes of optimization techniques have been developed, and are of interest in this context:

- Linear network optimization: specialization of LP to problems with graph structure (shortest path, transportation, transshipment, etc).
- Integer programming: decision variables are allowed only integer, or 0-1 values.
- Nonlinear programming: objective and constraints are nonlinear (usually differentiable) functions.
- Combinatorial optimization: decision variables have nice combinatorial structures (e.g., trees, permutations, matchings)

Not a strict separation, often merge...

Methodological issues and caveats

Optimization formulations are great (whenever they work), but their successful application is not always straightforward.

- An optimization approach is only as good as its formulation. *Even if mathematically correct*, could be nearly useless.
- Often many equivalent formulations exist, that differ substantially in efficiency and solvability. (Moderate) expertise is needed to choose appropriate descriptions.
- Optimization methods only know what they are told. All relevant features of the desired solution must be part of the model, either implicitly or explicitly (e.g., optimal control, learning)

Let's see this a bit more detail...

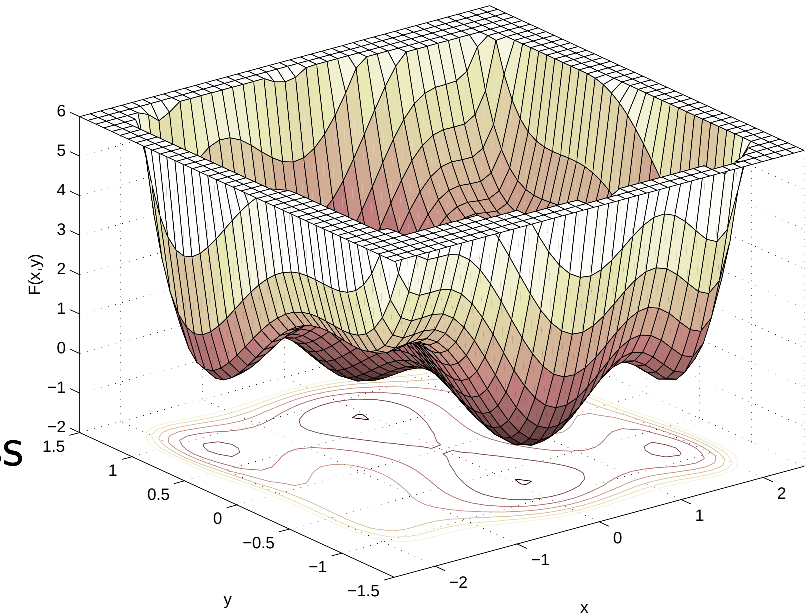
Example: Global optimization

Consider unconstrained optimization $\min_{x_1,\ldots,x_n} F(x_1,\ldots,x_n)$.

Typically, extremely difficult, many local minima.

Many questions can be posed in these terms (e.g., protein folding).

- Very flexible formulation (everything is an optimization problem!)
- But, hard to do anything substantial with it
- Complexity-theoretic obstacles (P vs. NP)
- Heuristics can work well, difficult to assess performance



Just because we can write an optimization problem, does not mean we can solve it...

Sensitivity and Robustness

Every model we write is only a coarse description of reality.

Conclusions about the model may or may not correspond to actual behavior.

Classically, this is not made too explicit.

- Validity of models often informal, implicit.
- These considerations sometimes incorporated through *sensitivity analysis*

Good step, but often incomplete (parametric assumptions, only small perturbations, etc.).

Sensitivity and Robustness (II)

More recently, better techniques to explicit account of difference between "real world" and "model" (robust control, robust optimization). Ex:

- Constraints and/or objective known only approximately
- "implemented" solution different from computed one

Different ways of assessing uncertainty (e.g., deterministic, stochastic, parametric, nonparametric, etc.)

In optimization: stochastic programming, robust optimization (e.g., Ben-Tal/El Ghaoui/Nemirosvski, etc.

Convex optimization

Particularly nice class of optimization problems: *Convex Optimization*

- Objective function is *convex*
- Feasible set is convex.

Many advantages:

- Modelling flexibility
- Tractability and scalability
- Sensitivity analysis relatively simple
- Can naturally incorporate robustness considerations

Key mathematical reason: duality theory.

Duality

Consequence of the *dual* nature of convex sets: points and hyperplanes.

In the case of LP:

$$\min c^T x \qquad \text{s.t.} \quad \begin{cases} Ax &= b \\ x &\geq 0 \end{cases} \tag{P}$$

Every LP problem has a corresponding *dual* problem, which in this case is:

$$\max b^T y \qquad \text{s.t.} \quad c - A^T y \geq 0. \tag{D}$$

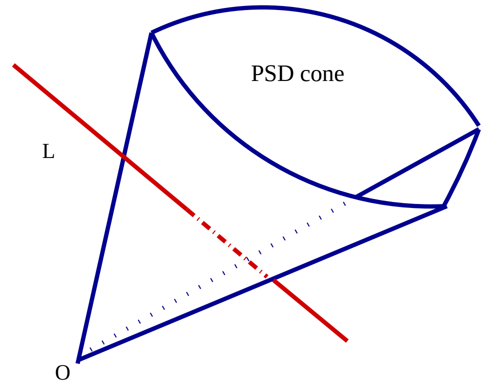
Weak duality: For any feasible solutions x, y of (P) and (D):

$$c^T x - b^T y = x^T c - (Ax)^T y = x^T (c - A^T y) \geq 0.$$

In other words, from any feasible dual solution we can obtain a lower bound on the primal.

Conversely, primal feasible solutions give upper bounds on the dual.

Semidefinite programming



A broad generalization of LP to symmetric matrices

$$\min \operatorname{Tr} CX \qquad \text{s.t.} \quad X \in \mathcal{L} \cap \mathcal{S}_+^n$$

- The intersection of an affine subspace $\mathcal{L}$ and the cone of positive semidefinite matrices.
- *Lots* of applications. A true "revolution" in computational methods for engineering applications
- Originated in control theory (Boyd *et al.*, etc) and combinatorial optimization (e.g., Lovász). Nowadays, applied everywhere.
- Convex finite dimensional optimization. Nice duality theory.
- Essentially, solvable in polynomial time (interior point, etc.)

Massachusetts Institute of Technology

Semidefinite programming (II)

An SDP problem in standard primal form is written as:

$$\begin{array}{lrcll} \text{minimize} & C \bullet X & & & \\ \text{subject to} & A_i \bullet X & = & b_i, & i = 1, \ldots, m \\ & X & \succeq & 0, & \end{array} \qquad (1)$$

where C, A_i are $n \times n$ symmetric matrices, and $X \bullet Y := \mathrm{Tr}(XY)$.

- Set of feasible solutions is always convex.
- Weak duality always holds:

$$C \bullet X - b^T \mathbf{y} = (C - \sum_{i=1}^{m} A_i y_i) \bullet X \geq 0.$$

Algorithms to solve SDP (interior-point) are generalizations of those for LP.
Other methods also exist, more suitable for truly large-scale problems.

What about nonconvexity?

What happens when problems/formulations are not convex?

Several alternatives:

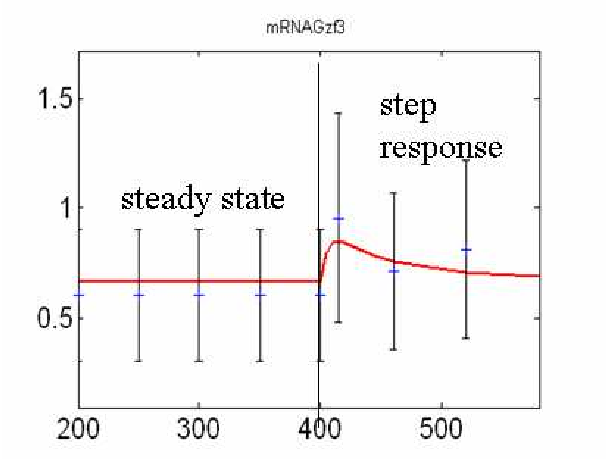
- Suitable local algorithms exist, but hard to assess global performance properties. Heuristics also used, sometimes (often?) adequate.
- Convexify! Replace/relax with an associated convex problem, sometimes with suboptimality/performance guarantees. Very powerful, particularly if reformulated problem can be solved efficiently.
- Combination of techniques, e.g., convex relaxations + branch/bound. Often, yields best available methods (e.g., integer programming).

Data consistency

Elementary reaction models for gene expression in yeast

$$
\begin{aligned}
\frac{d[TF]}{dt} &= -K_{A,1} \cdot [TF] \cdot [GENE] + K_{D,1} \cdot [TF \bullet GENE] \\
\frac{d[GENE]}{dt} &= -K_{A,1} \cdot [TF] \cdot [GENE] + K_{D,1} \cdot [TF \bullet GENE] \\
\frac{d[TF \bullet GENE]}{dt} &= K_{A,1} \cdot [TF] \cdot [GENE] - K_{D,1} \cdot [TF \bullet GENE] - \\
&\quad -K_{A,2} \cdot [TF \bullet GENE] \cdot [RNAPol] + K_{D,2} \cdot [TF \bullet GENE \bullet RNAPol] + \\
&\quad +K_{TC} \cdot [TF \bullet GENE \bullet RNAPol]
\end{aligned}
$$

- Nonlinear dynamics
- Microarray data of wildtype and mutants
- Steady state + dynamic measurements
- Extract as much information as possible



What parameter/rate values are consistent with measurements?

Joint work with L. Küpfer and U. Sauer (ETH Zürich)

Steady-state data consistency

(Küpfer-Sauer-P., BMC Bioinformatics 12:8, 2007)

Want to determine parameter/rate values that are inconsistent with data.

- Equilibrium conditions give a system of quadratic equations
- Steady-state measurements give linear constraints

After homogenization, can rewrite as

$$x^T Q_i x = 0, \qquad Ax \geq 0.$$

Sufficient condition for infeasibility (weak duality):

$$\sum_i \lambda_i Q_i + A^T N A \preceq 0, \qquad N \geq 0.$$

Can find suitable λ_i, N by solving a semidefinite program!

From singletons to sets

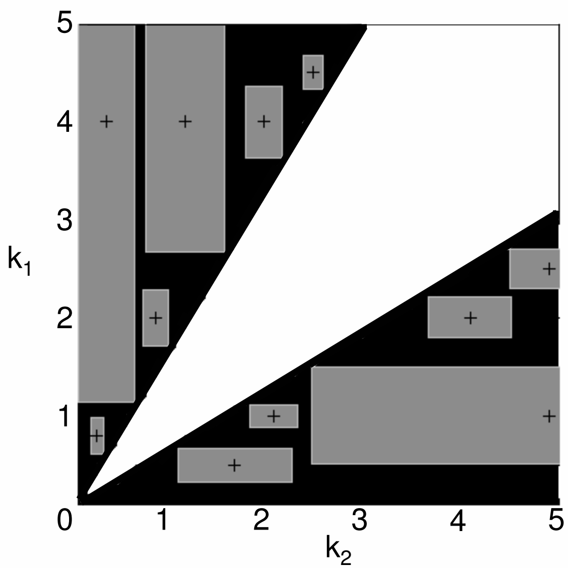
- What if we want to invalidate whole *sets* of parameters at a time?
- Can adapt the techniques, to provide *boxes* of invalid parameters.

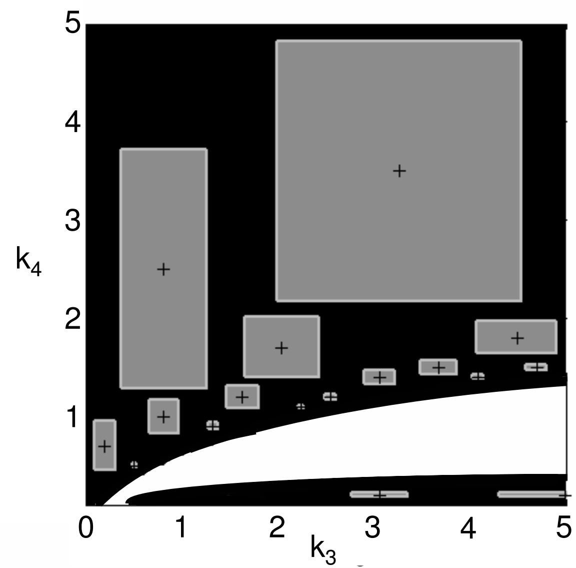
Key idea: assume we want to show that

$$A_0 + \sum_{i=1}^{n} \delta_i A_i \succeq 0, \qquad \text{for all } |\delta_i| \leq 1.$$

A sufficient condition: there exist matrices W_i, such that

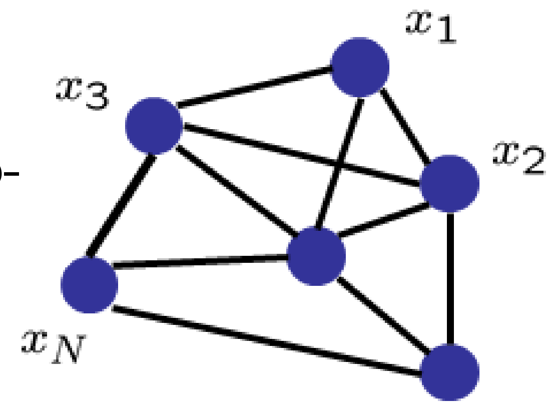
$$A_0 = \sum_{i=1}^{n} W_i, \qquad \begin{cases} W_i + A_i \succeq 0 \\ W_i - A_i \succeq 0 \end{cases}, \quad i = 1, \ldots, n.$$





Graphical model identification

Recall notion of *conditional* independence: random variables X and Y are independent *conditioned* on Z if $P(X|Z)P(Y|Z) = P(X,Y|Z)$.



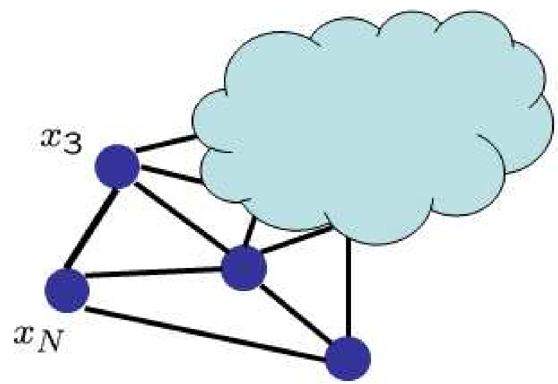
A Gaussian model whose conditional independence properties are given by a graph.

The covariance Σ^{-1} is sparse.

No longer true if the model has *hidden variables*.

But, it is the sum of a sparse and a low-rank matrix:

$$\hat{K}_o = \Sigma_o^{-1} = K_o - K_{o,h} K_h^{-1} K_{h,o}$$



Massachusetts Institute of Technology

Optimization formulation

Determine the structure of a statistical model with hidden variables.

Simultaneously find the graph (sparse) and the hidden variables (low-rank).

Consider the sample covariance

$$\Sigma_O^n = \frac{1}{n}\sum_{i=1}^{n} X_O^i (X_O^i)^T$$

Proposal: Optimize regularized log-likelihood (convex)

$$\begin{aligned}(\hat{S}_n, \hat{L}_n) \quad := \quad & \arg\min_{S,L} \mathsf{Tr}(S-L)\Sigma_O^n - \log\det(S-L) + \lambda_n[\gamma||S||_1 + \mathsf{Tr}L] \\ & \text{s.t. } S - L \succ 0, \quad L \succeq 0.\end{aligned}$$

Under suitable identifiability conditions, and parameters, the estimate given by this convex program yields the correct sign and support for S_n, and the correct rank for L_n. Can obtain explicit sample complexity rates.

Massachusetts Institute of Technology

Summary

- Optimization techniques have many applications in bioengineering
- Requires careful thought at modeling time, no “black box” (yet!?)
- Methods have enabled many new applications
- Mathematical structure must be exploited for reliability and efficiency