Robust Optimization and Applications

Laurent El Ghaoui
elghaoui@eecs.berkeley.edu

IMA Tutorial, March 11, 2003
Thanks

- **The organizers:** Mike Todd, Scot Adams (IMA)

- **Contributors:** Gert Lanckriet, Arnab Nilim, Chiru Bhattacharyya

- **Colleagues:** Peter Bartlett, Giuseppe Calafiore, Nello Cristianini, Michael Jordan, Arkadi Nemirovski
Optimization models

• Classical optimization framework:

\[
\min_x f_0(x) : f_i(x) \leq 0, \quad i = 1, \ldots, m
\]

• functions \( f_i, i = 0, \ldots, m \) are exactly known

• once \( x^* \) is computed it is implemented exactly
Pitfalls

- In practice, exact values of input data seldom known

- In addition, optimal variables may be implemented with errors

- Case study by Ben Tal & Nemirovski (2000)

*Small errors in data values can make optimal solutions highly infeasible*
Robust Optimization Paradigm

- Assume uncertainty model for data is known (e.g. bounds)

- Define robustly feasible solutions as ones that are guaranteed to remain feasible for all admissible data values

- Optimize objective over set of robustly feasible solutions (or an inner approximation to it)
Approximating a robust solution

- Robust counterpart may be much harder than original problem

- Hence we may need to approximate
Agenda

• Part I: Theory
  – Robust conic programming
  – Relaxations and restrictions
  – Extensions and variations

• Part II: Contextual applications
  – Aircraft path planning
  – Statistical classification
LP as a conic problem

- LP: \( \min c^T x : Ax = b, \ x \in \mathbb{R}^n_+ \)

- \( \mathbb{R}^n_+ \) admits "self-concordant" barrier
  \[
  \phi(x) = - \sum_i \log x_i
  \]

- Hence, LP solvable via interior-point methods for SC problems

which cones admit self-concordant barriers?
Second-order cone programming

- LP: \( \min c^T x : Ax = b, \ x \in \mathcal{K}_1 \times \ldots \times \mathcal{K}_m \)

- Each \( \mathcal{K}_i \) is an "ice-cream cone", of the form \( \{(z,t) : t \geq \|z\|_2\} \)

- Includes LP, convex quadratic programs

- SC barrier for ice-cream cone:
  \[
  \phi(z,t) = -\log(t - \frac{z^T z}{t}) \quad (t > 0)
  \]

- Cost is moderately higher than LP

\[ \mathcal{K} = \{(z,t) : t \geq \|z\|_2\} \]
Semidefinite programming

- LP: \( \min c^T x \ : \ Ax = b, \ x \in S^n_+ \)

- \( S^n_+ \) is the set of \( n \times n \), symmetric, positive semidefinite matrices

- Includes LP, SOCP

- SC barrier for SDP
  \[ \phi(x) = -\log \det x \ (x \succ 0) \]

- Cost is substantially higher than LP

\[ \mathcal{K} = \{ x = x^T \ : \ x \succeq 0 \} \]
Dual form of conic program

• Primal conic problem:

\[ \min c^T x \quad : \quad Ax = b, \quad x \in \mathcal{K} \]

\( \mathcal{K} \) is product of cones (positive orthant, ice-cream, positive semidefinite)

• Dual conic problem:

\[ \max b^T y \quad : \quad c - A^T y \in K \]

• constraint is called linear matrix inequality (LMI)
Robust conic programming

• Conic program in dual form:

$$\max b^T y : c - A^T y \in K$$

• Assume $A \in \mathcal{U}$, otherwise unknown

• Robust conic program:

$$\max b^T y : \forall A \in \mathcal{U}, \ c - A^T y \in K$$

• Still convex, but maybe much harder than original conic program
Polytopic uncertainty

- Robust conic program:

\[
\max b^T y : \forall A \in \mathcal{U}, \ c - A^T y \in K
\]

- If \( \mathcal{U} \) is a polytope described by its vertices:

\[
\mathcal{U} = \text{Conv}\{A_1, \ldots, A_L\},
\]

then robust conic program is an ordinary conic program:

\[
\max b^T y : c - A_i^T y \in K, \ i = 1, \ldots, L
\]

- **Drawback:** in practice, vertices may not be known or too many to handle
Robust LP

For LP:

- For ellipsoidal uncertainty on the coefficient vectors, robust LP is an SOCP
- For uncertainty described by LMI, robust LP is an SDP
Robust LP with ellipsoidal uncertainty

- Consider LP

\[
\min_x c^T x \ : \ a_i^T x \leq b_i, \ i = 1, \ldots, m
\]

- Ellipsoidal uncertainty:

\[a_i \in \mathcal{E}_i = \{\hat{a}_i + P_i^{1/2} u : \|u\|_2 \leq 1\}\]

- Robust counterpart:

\[
\min_x c^T x : \ a_i^T x + \|P_i^{1/2} x\|_2 \leq b_i, \ i = 1, \ldots, m
\]
Robust LP as SOCP

- Robust LP with ellipsoidal uncertainty is an SOCP
- Hence, little overhead added but lots of robustness
- Regularization terms $\|P_i^{1/2}x\|_2$ directly related to size & shape of uncertainty
Example: robust portfolio design

- Assume returns \( r_i \) are exactly known, allocate portfolio weights \( w_i \) so as to maximize total return:

\[
\max_{w} r^T w : w^T 1 = 1, \ w \geq 0
\]

A highly non-robust proposition!

- Now assume returns are only known within an ellipsoid

\[
\mathcal{E} = \{ r = \hat{r} + \kappa \Sigma^{1/2} u : \|u\|_2 \leq 1 \}
\]

(e.g. \( \Sigma \) is the empirical covariance matrix)

- robust counterpart:

\[
\max_{w} \min_{r \in \mathcal{E}} r^T w : w^T 1 = 1, \ w \geq 0
\]
Solution of robust portfolio problem

- Robust portfolio problem is an SOCP
  \[
  \max_w \hat{\mathbf{r}}^T\mathbf{w} - \kappa \|\Sigma^{1/2}\mathbf{w}\|_2 : \mathbf{w}^T\mathbf{1} = 1, \quad \mathbf{w} \geq 0
  \]

- can be interpreted as a risk vs. variance tradeoff
  (\(\|\Sigma^{1/2}\mathbf{w}\|_2\) is standard deviation of total return)

- Regularization term makes solution unique, continuous
Robust SOCP

- Robust SOCP has the form
  \[ \min e^T x \quad : \quad \forall u \in \mathcal{U}, \; \| A_i(u)x + b_i(u) \|_2 \leq c_i(u)^T x + d_i(u) \]
  may be hard in general

- Case of ellipsoidal uncertainty: if
  - \( \mathcal{U} \) is an ellipsoid
  - data \((A_i, b_i, c_i, d_i)\) are affine in \( u \)
  - uncertainties affecting \((A_i, b_i)\) are independent of those affecting \((c_i, d_i)\)
  then robust SOCP can be solved as an SDP

- see Ben Tal, Nemirovski, El Ghaoui (2000)
Example: robust least-squares

- Robust least-squares problem:

\[
\min_{x} \max_{u \in \mathcal{U}} \|A(u)x - b(u)\|_2
\]

- If \( \mathcal{U} \) is an ellipsoid, and \( A(u), b(u) \) are affine in \( u \), above is an SDP

- Related case:

\[
\min_{x} \max_{\|U\| \leq \rho} \| (A + U)x - b \|_2
\]

can be solved exactly as an SOCP

\[
\min_{x} \| Ax - b \|_2 + \rho \| x \|_2
\]

(compare with classical regularization techniques)
Robust SDP

- Robust SDP has the form

$$\min c^T x : \forall u \in \mathcal{U}, \ F_0(u) + x_1 F_1(u) + \ldots + x_m F_m(u) \succeq 0$$

may be hard in general

- Case of ellipsoidal uncertainty: if
  
  - $\mathcal{U}$ is an ellipsoid
  
  - $F_i(u)$ are affine in $u$

  then robust SDP can be approximated by an SDP

- solution to approximation problem is guaranteed to be robust
Example: robust control

- uncertain dynamical system:
  \[ \dot{x} = A(t)x, \quad A(t) \in \mathcal{A} \]

- quadratic stability:
  \[ \exists \ P \succ 0, \ \forall \ A \in \mathcal{A}, \quad A^T P + PA \prec 0 \]

- when \( \mathcal{A} = \{A + BUC \ : \ ||U|| \leq 1\} \), above equivalent to SDP condition
  \[ P \succ 0, \quad \begin{bmatrix} A^T P + PA + C^T C & PB \\ B^T P & -I \end{bmatrix} \prec 0 \]

the latter is equivalent to ”Riccati inequality”

\[ A^T P + PA + C^T C + PBB^T P \prec 0 \]
Analysis of robust conic problems

- computational complexity
- relaxations
- restrictions
- quality estimates
- sampling
Relaxations

- Robust SDP:
  \[
  \min c^T x : \forall u \in \mathcal{U}, \ F_0(u) + x_1 F_1(u) + \ldots + x_m F_m(u) \succeq 0
  \]
  assume $F_i$'s affine in $u$, and $\mathcal{U}$ is an ellipsoid

- analysis problem: given symmetric $F_j$'s, consider condition
  \[
  F_0 \succeq \sum_j u_j F_j \ \forall u, \ \|u\|_2 \leq 1
  \]
  a sufficient condition is
  \[
  \exists S > 0, \ 2F_0 \succeq S + \sum_j F_j S^{-1} F_j
  \]

- Above can be expressed as LMI in $F_j$'s and $S$
Quality estimates

- Consider robust LMI condition

\[ \forall u, \|u\|_2 \leq \rho, \quad F(x, u) \geq 0 \]

where \( F(x, u) \) is affine in \( x, u \)

- can be approximated by SDP, by an LMI \( \mathcal{L}(\rho) \)

- assume \( \mathcal{L}(1) \) is feasible, find a value of \( \rho \) such that \( \mathcal{L}(\rho) \) is not

- \( \rho \) is a measure of the quality of the approximation
Quality estimates: some results

• For robust LMI with affine dependence (on $x, u$), can obtain quality estimate that depends only on input data

• For example, with ellipsoidal uncertainty, we have

$$\rho^* \leq \min \left( \sqrt{n}, \sqrt{l} \right),$$

where $n$ is size of $x$ and $l$ of $u$

• for improved results, see Nemirovski, 2001, 2002
restriction

- Consider robust SDP

\[ \min c^T x : \forall u \in \mathcal{U}, F(x,u) \geq 0 \]

- can restrict it by SDP

\[ \min c^T x : F(x,u_j) \geq 0, \quad j = 1, \ldots, N \]

where \( u_j \)'s sample \( \mathcal{U} \)

- above does not produce guaranteed robustness

- how much to sample?
Sampling

• Consider robust convex problem

\[ \min c^T x : \forall u \in \mathcal{U}, f_i(x, u) \leq 0, \quad i = 1, \ldots, m \]

where \( f_i(x, u) \) is convex in \( x \in \mathbb{R}^n \), and \( \mathcal{U} \) is compact

• Sampling approach: replace \( \mathcal{U} \) by a randomly chosen finite subset of \( \mathcal{U} \), and solve corresponding (ordinary) convex problem

• if the number of samples satisfies

\[ N \geq \frac{n}{\epsilon \beta} - 1, \]

then with probability \( 1 - \beta \), the probability of violation of the constraints is less than \( \epsilon \)

• see Calafiore, Campi (2002)
Variations on Robust Conic Programming

- robust discrete optimization

- robustness wrt probability distributions ("chance-constrained programming")

- adjustable variables and feedback control
A Boolean problem

quadratic boolean problem:

$$\max x^T W x : x_i^2 = 1, \ldots, n$$

- arises in e.g. graph theory
- NP-hard
Max-quad as a robust LP

assume WLOG $W \succ 0$

- max-quad can be written as robust LP

$$\min t : \forall x \in \mathcal{B}, x^T W x \leq t$$

where $\mathcal{B}$ is unit hypersquare

- constraint expresses that $\mathcal{B} \subset \mathcal{W}(t)$, where

$$\mathcal{W}(t) = \{x : x^T W x \leq t\}$$
Rank relaxation

- quadratic boolean problem:

  $$\max x^T W x : x_i^2 = 1, \ i = 1, \ldots, n$$

- objective is **linear** in $X = xx^T$

  $$\max \text{Tr} WX : X \succeq 0, \ X_{ii} = 1, \ \text{Rank} \ X = 1$$

- SDP relaxation: drop rank constraint

  $$\max \text{Tr} WX : X \succeq 0, \ X_{ii} = 1$$
Boolean optimization: geometric approach

\[ \mathcal{W}(t) = \{ x : x^T W x \leq t \} \]

\[ \mathcal{Y}(t) = \{ x : x^T Y x \leq t \} \]

Condition \( \mathcal{Y}(t) \subset \mathcal{W}(t) \) holds iff \( Y \succeq W \)

If \( Y \) is diagonal, boolean problem is easy:

\[
\max_{x_i^2 = 1} x^T Y x = \text{Trace } Y
\]

Best upper bound is an SDP

\[
\min \text{ Trace } Y : Y \succeq W, \ Y \text{ diagonal}
\]
SDP for boolean / nonconvex optimization

• geometric and algebraic approaches are dual (see later), *yield the same upper bound*

• SDP provides upper bound
  *may recover primal variable by sampling*

• approach extends to many problems
  *eg, problems with (nonconvex) quadratic constraints & objective*

• in some cases, quality of relaxation is provably good
Robust boolean optimization

- quadratic boolean problem:
  \[
  \max x^T W x : \quad x_i^2 = 1, \quad i = 1, \ldots, n
  \]

- assume data matrix \( W \in \mathcal{U} \), otherwise unknown

- robust counterpart:
  \[
  \max \min_{W \in \mathcal{U}} x^T W x : \quad x_i^2 = 1, \quad i = 1, \ldots, n
  \]
SDP relaxation of robust problem

- SDP relaxation of robust quadratic problem

$$\max \min_{W \in \mathcal{U}} \text{Tr} W X : X \succeq 0, \ X_{ii} = 1$$

- by duality, express above as SDP:

$$\min_{Y,W} \text{Tr} Y : Y \succeq W, \ Y \text{ diagonal}$$
Chance-constrained programming

- Assume $a, b$ obey to some given distribution $\pi$:

$$\text{Prob}\{ a^T x \leq b \} \geq \epsilon$$

is convex in $x$ if $\pi$ has log-concave density and $\epsilon > 1/2$

- example: if $a \sim \mathcal{N}(\hat{a}, \Gamma)$ then above reads

$$\hat{a}^T x + \kappa \| \Gamma^{1/2} x \|_2 \leq b$$

where $\kappa$ depends on $\epsilon$

- if distribution is unknown, except by its first two moments, similar formula holds (with different $\kappa$)

- above can be related to ellipsoidal uncertainties
Problems with adjustable parameters

- robust counterpart with adjustable parameters

\[ \max b^T y_1 : \forall u \in \mathcal{U}, \exists y_2(u) : c - A_1^T y_1 - A_2^T y_2(u) \in \mathcal{K} \]

- interpretation: decide on \( y_1 \) now, adjust \( y_2 \) as info on \( u \) arises, later

- example: robust stability of \( \dot{x} = A(u(t))x, u(t) \in \mathcal{U} \)

\[ \forall u \in \mathcal{U}, \exists P(u) > 0, A(u)^T P(u) + P(u) A(u) < 0 \]
Adjustable parameters: some results

\[
\max b^T y_1 : \forall u \in \mathcal{U}, \exists y_2(u) : c - A_1(u)^T y_1 - A_2^T y_2(u) \in \mathcal{K}
\]

- If we assume
  - \(A_1(u), y_2(u)\) affine (cf linear feedback)
  - matrix \(A_2\) independent of \(u\)

then we can reduce problem to and ordinary robust conic program

- see Nemirovski, 2002
Link with feedback control

- linear controlled dynamic system
  \[ \dot{s} = As(t) + Bx(t), \quad t \geq 0 \]
- State feedback \( x(t) = Ks(t) \), where \( K \) is constant
- Stability of \( \dot{s} = (A + BK)s \):
  \[ \exists P > 0, (A + BK)^T P + P(A + BK) < 0 \]
- above is LMI in \( S = P^{-1}, X = KS \):
  \[ S > 0, \quad AS + BX + (AS + BX)^T < 0 \]
- state \( s \) can be interpreted as ”uncertainty” and control \( x \) as an adjustable variable
- linearity of \( x \) as a function of \( s \) is crucial
Challenges

- approximation techniques
- data-dependent quality estimates
- data-dependent sampling complexity estimates
- interplay with probabilistic descriptions
- estimation of uncertainty bounds
Set estimation

- LP decision-making model: \( \min c^T x : Ax \leq b \)
- classical approach to use LP:
  - find an estimate of data \((\hat{A}, \hat{b}, \hat{c})\)
  - solve resulting LP
- does not account for uncertainty!
- in robust optimization, we must use an estimate of the set of confidence for the data
Part I: summary

- robust optimization is a developing set of techniques to handle deterministic uncertainty in optimization problems

- many links with stochastic programming

- in some cases, much more efficient
Part II: Contextual Applications

- robust path planning of aircraft through convective weather

- classification problems in statistics
Robust path planning

- Markov decision process model
- Robust dynamic programming
- Application to aircraft routing through convective weather

Co-author: Arnab Nilim
Uncertainty in Markov Decision Process

- consider finite-state, finite-action Markov decision problems

- in practice, estimation errors corrupt input data ("transition matrices")

- optimal paths may be highly sensitive to estimation errors
Agenda

- develop robust dynamic programming
- obtain practical solutions for problems with moderate size
- describe applications in path planning of aircraft traffic
- outline research challenges for large-scale problems
Markov decision problem

Bellman recursion for the "value function" $V$:

$$V_t(i) = \max_{a \in A} r_t(i, a) + \sum_j P_a(i, j)V_{t+1}(j)$$

where

- $r_t(i, a)$ is cost of taking action $a$ in state $i$

- $P_a(i, j)$ is transition probability from state $j$ to $i$

in practice, transition probabilities are estimated
Previous Work

- Many authors recognized the need of addressing robustness wrt errors in transition matrices (White & Eldeib, Givan et al., Barnell et al.)

- most models assume componentwise uncertainty

- these models lead to very conservative policies

- an accurate yet computationally tractable model is crucial
Robust dynamic programming

Robust dynamic programming recursion:

$$V_t(i) = \min_{P_a \in \mathcal{P}_a} \max_{a \in \mathcal{A}} r_t(i, a) + \sum_j P_a(i, j)V_{t+1}(j)$$

$$= \max_{a \in \mathcal{A}} \min_{P_a \in \mathcal{P}_a} r_t(i, a) + \sum_j P_a(i, j)V_{t+1}(j)$$

- solves a game in which transition matrices are opponents
- provides exact lower bound on value function
Inner problem

- Each step of robust Bellman recursion requires to solve a problem of the form
  \[ \min \mathcal{L}(P) : P \in \mathcal{U}, \]
  where \( \mathcal{L} \) is linear, and \( \mathcal{U} \) is the uncertainty set on transition matrix \( P \)

- Problem is convex whenever \( \mathcal{U} \) is

- Computational complexity depends on \( \mathcal{U} \)
Worst-case performance of a policy

- what is the value of a given policy?

- worst-case dynamic programming recursion: given $a \in \mathcal{A}$, compute
  \[ V_t(i) = \min_{P_a \in \mathcal{P}_a} r_t(i, a) + \sum_j P_a(i, j)V_{t+1}(j) \]

- again, need to solve a (convex) optimization problem at each step
Describing uncertainty

if $\mathcal{P}_a$’s are ellipoids, or boxes, inner problem is simple
drawbacks of these models:

• statistically inaccurate

• too conservative

on the other hand, solving a general convex problem might be too expensive!

we need to be very careful in choosing the uncertainty model
Joint estimation and optimization

Set of confidence should

- be statistically accurate
- lead to a tractable inner problem

likelihood-based statistical descriptions of uncertainty are often a good answer

in statistics, these regions are approximated by "more understandable" sets (ellipsoids, intervals)

with optimization in mind, we need not approximate these regions
Estimating a transition matrix

- Log-likelihood function:

\[ L(P) = \sum_{i,j} N_{ij} \log P_{ij} \]

where \( N \) is the matrix of empirical frequencies

- \textit{a priori} constraints:

\[ P \in \mathcal{C} := \{ P : P1 = 1, P \geq 0 \} \]

- maximum-likelihood estimate:

\[ P^* = \arg \max_{P \in \mathcal{C}} L(P) \]

with \( \mathcal{C} \) given above, \( P^* = N \)
Likelihood regions

likelihood region:

\[ \mathcal{P}(\beta) := \{ P : P \in \mathcal{C}, \ L(P) \geq \beta \} \]

where

- \( \mathcal{C} \) describes \textit{a priori} constraints:

\[ \mathcal{C} := \{ P : P \mathbf{1} = \mathbf{1}, \ P \geq 0 \} \]

- \( \beta \) is a \textit{scalar} measure of uncertainty
likelihood regions

- Robust dynamic programming recursion:

\[ V_t(i) = \max_{a \in \mathcal{A}} \min_{P_a \in \mathcal{P}_a(\beta)} r_t(i, a) + \sum_j P_a(i, j)V_{t+1}(j) \]

where \( \mathcal{P}_a(\beta) \) is a (convex) likelihood region

- inner problem is: given \( V_{t+1}(j) \), \( a \), compute

\[ \min_{P_a \in \mathcal{P}_a(\beta)} \sum_j P_a(i, j)V_{t+1}(j) \]
Reduction to a 1-D problem

• log-likelihood function is a barrier for feasible set $C$

• hence problem reduces to one with two scalar constraints:

$$\min_{P} \sum_{j} P(i, j)V_{t+1}(j) : \sum N_{ij} \log P_{ij} \geq \beta, \ P1 = 1$$

• using duality, can reduce this to a convex problem in one variable

• can solve it in $O(1)$ time using (say) bisection
Complexity results

complexity of classical Bellman recursion is linear in $nmT$, where

- $n$ is number of states
- $m$ is number of actions
- $T$ is time horizon

complexity of robust Bellman recursion is the same
Application to aircraft routing

motivation: convective weather accounts for 70% of delays in air traffic

- use a Markov chain model for the weather conditions (presence & strength of storms)
- airspace is a grid with known fixed locations for potential storms

*dynamic programming solution is attractive for moderate number of storms & aircraft*
Markov chain model for the storms

- state 0: no storm
- state 1: storm

Probability transition matrix:

\[
P = \begin{bmatrix}
p & 1 - p \\
1 - q & q
\end{bmatrix},
\]

where

- \( p \): probability of storm not present, given it's not present at current stage
- \( q \): same for state "1"
information update and recourse

- storm probabilities are updated every 15 minutes
- aircraft learn the exact storm situation when they get close to it
- decisions should incorporate these information updates
Dynamic programming model

- seek to minimize one objective (e.g., expected delay)
- optimal paths are computed recursively via dynamic programming
- avoid to search a large part of the state-space, due to an effective pruning procedure
Nominal algorithm
Sample path planning
## Improvements over obvious strategies

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Conservative Strategy (avoid storm)</th>
<th>Over-optimistic Strategy (ignore storm and apply recourse at the last moment, if needed)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Scenario 1</strong></td>
<td>66.42%</td>
<td>42.76%</td>
</tr>
<tr>
<td><strong>Scenario 2</strong></td>
<td>54.78%</td>
<td>49.81%</td>
</tr>
</tbody>
</table>
Robustness

- dynamic programming assumes perfect knowledge of transition probabilities
- resulting "optimal" strategies may be extremely sensitive to estimation errors
- we have used the robust Bellman recursion with likelihood regions of uncertainty
- computational cost is roughly the same
Optimality vs. uncertainty level

- For small uncertainty levels, worst-case cost of nominal strategies are much worse than that of robust strategies.
- Discrepancy grows with uncertainty level.
- At some point, nominal strategy is even worse than conservative one.

Figure 1: Optimal value vs. uncertainty level (negative lower bound on the log-likelihood function), for both the classical Bellman recursion and its robust counterpart.
Errors in uncertainty level

- a rough guess of the uncertainty level is enough to beat the nominal strategy

- with robust strategies, errors in uncertainty level incur only a minor delay

Figure 1: Optimal value vs. uncertainty level (negative lower bound on the log-likelihood function), for the classical Bellman recursion and its robust counterpart (with exact and inexact predictions of the uncertainty level $U_L$).
Extensions

- multiple storms
  - complexity grows exponentially with \# storms
  - however, on a given path only a few storms are relevant

- uncertainty about storm location, intensity, and size
  (use Markov chain model with more than two states)

- multiple aircraft (including collision avoidance)
  - complexity is exponential in \# aircraft
  - address this using priority constraints
Summary of results

- classical DP suffers from “curse of dimensionality” but is widely used
- it suffers also from curse of sensitivity
- robust version based on likelihood enables
  - accurate description of uncertainty
  - control of sensitivity via robustness
- robustness requires very moderate increase in computational cost
Some references


- A. Nilim, L. El Ghaoui, M. Hansen and V. Duong. ”Trajectory-based Air Traffic Management (TB-ATM) under weather uncertainty”, 4th USA/EUROPE ATM R&D Seminar, Santa Fe, New Mexico, December 2001
Robust Classification

- basics of classification
- robust classification

Co-authors: Gert Lanckriet, Chiru Bhattacharyya, Mike Jordan, Nello Cristianini, Peter Bartlett
Linear Classification

two-class, linear separation problem

many classifiers are possible!
What is a classifier?

- a linear classifier corresponds to a hyperplane in the feature space

- classification rule: for an observed response vector $x$, classify according to

$$\text{sign}(w^T x + b),$$

where $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ are the parameters of the hyperplane

$$w^T x + b < 0$$

$$w^T x + b > 0$$
Classification constraints

- correct classification:
  \[ y_i(w^T x_i + b) \geq 1, \quad i = 1, \ldots, N \]
  where \( y_i = \pm 1 \) are the labels

- allowing misclassification:
  \[ y_i(w^T x_i + b) \geq 1 - z_i, \quad i = 1, \ldots, N \]
  with \( z \geq 0 \)
robust classification: support vector machine

each point can move around within a Euclidean ball

increase ball radius until separation becomes impossible

classifier maximizes robustness wrt spherical data noise
box uncertainty model

each point can move around within a box

increase box size until separation becomes impossible

selects a sparse classifier with maximal robustness
formulations

• support vector machine:

\[
\min \|w\|_2 + Cz^T1 : y_i(w^Tx_i + b) \geq 1 - z_i, \ z_i \geq 0, \ i = 1, \ldots, N
\]

• linear programming support vector machine:

\[
\min \|w\|_1 + Cz^T1 : y_i(w^Tx_i + b) \geq 1 - z_i, \ z_i \geq 0, \ i = 1, \ldots, N
\]

In the above, $C$ is a measure of trade-off between accuracy and robustness
extensions

- use ellipsoidal uncertainty models for each data points, e.g.

\[ x_i \in \mathcal{E}_i = \left\{ \hat{x}_i + P^{1/2}u : \|u\|_2 \leq 1 \right\} \]

leads to modified support vector machine:

\[ \min \|P^{1/2}w\|_2 + Cz^T1 : y_i(w^T\hat{x}_i + b) \geq 1 - z_i, \ z_i \geq 0, \ i = 1, \ldots, N \]

- similarly, can use different box models

in practice, we often have information (e.g. intervals of confidence) that can help in building models
minimax probability machine

goal:

- make assumptions about the data generating process
- do not assume Gaussian distributions
- use second-moment analysis of the two classes
Problem statement

let \( \hat{x}_\pm, \Gamma_\pm \) be the mean and covariance matrix of class \( y = \pm 1 \)

**MPM:** maximize \( \epsilon \) such that there exists \((a, b)\) such that

\[
\inf_{x \sim (\hat{x}_+, \Gamma_+)} \text{Prob}\{a^T x \leq b\} \geq 1 - \epsilon \\
\inf_{x \sim (\hat{x}_-, \Gamma_-)} \text{Prob}\{a^T x \geq b\} \geq 1 - \epsilon
\]
SOCP formulation

two-sided, multivariable Chebychev inequality:

\[
\inf_{x \sim (\hat{x}, \Gamma)} \text{Prob}\{a^T x \leq b\} = \frac{(b - a^T \hat{x})_+^2}{(b - a^T \hat{x})_+^2 + a^T \Gamma a}
\]

MPM problem leads to SOCP

\[
\min_a \|\Gamma_+^{1/2} a\|_2 + \|\Gamma_-^{1/2} a\|_2 : a^T (\hat{x}_+ - \hat{x}_-) = 1
\]

complexity is the same as standard SVMs
Dual problem

- express problem as unconstrained min-max problem:

\[
\min_a \max_{\|u\|_2 \leq 1} u^T \Gamma^1/2 a - u^T \Gamma^1/2 a + \lambda(1 - a^T (\hat{x}_+ - \hat{x}_-))
\]

- exchange min and max, and set \( \kappa := 1/\lambda \):

\[
\min_{\kappa, u \pm} \kappa : x_+ + \Gamma^1/2 u_+ = x_- + \Gamma^1/2 u_- , \quad \|u\|_2 \leq \kappa
\]
Geometric interpretation

dual problem amounts to find smallest $\kappa$ for which $\mathcal{E}_+(\kappa) \cap \mathcal{E}_-(\kappa) = \emptyset$, where

$$\mathcal{E}_\pm := \left\{ x_+ + \Gamma_+^{1/2}u : \|u\|_2 \leq \kappa \right\}$$
Robust classification: summary of results

lots of interplay between statistics and convex optimization

- typical example is support vector machines
- SVMs and variants can be interpreted as robust optimization models
- parallel leads to useful variations
Wrap-up