# Semidefinite Optimization Using MOSEK

CO@work, October 8th, 2015

Joachim Dahl

www.mosek.com

Section 1

# Conic optimization

- Consider a standard linear problem (LP)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

with variables $x \in \mathbb{R}^n$ and data $(A, b, c) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$.

- The simplest class of interesting problems we can easily solve.
- An extremely mature technology.
- Used extensively in all areas of industry.
- To what extent can we generalize this model?

- Consider a standard linear problem (LP)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

with variables $x \in \mathbb{R}^n$ and data $(A, b, c) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$.

- The simplest class of interesting problems we can easily solve.
- An extremely mature technology.
- Used extensively in all areas of industry.
- To what extent can we generalize this model?

- Consider a standard linear problem (LP)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

  with variables $x \in \mathbb{R}^n$ and data $(A, b, c) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$.
- The simplest class of interesting problems we can easily solve.
- An extremely mature technology.
- Used extensively in all areas of industry.
- To what extent can we generalize this model?

# Conic optimization (for symmetric cones)

## The natural extension of linear optimization

- Linear optimization over symmetric cones remains tractable.
- Same efficient interior-point methods as for LPs.
- Much more flexible than one might think!

Three symmetric cones:

- Linear cone $x \in \mathbb{R}^n_+$.
- Quadratic cones:

$$\mathcal{Q}^n := \{x \in \mathbb{R}^n \mid x_1 \geq \sqrt{x_2^2 + \cdots + x_n^2}\},$$

$$\mathcal{Q}_r^n := \{x \in \mathbb{R}^n \mid 2x_1x_2 \geq x_3^2 + \cdots + x_n^2,\ (x_1, x_2) \geq 0\}.$$

- Semidefinite cone:

$$\mathcal{S}^n_+ := \{X \in \mathbb{R}^{n \times n} \mid X = X^T,\ z^T X z \geq 0, \forall z \in \mathbb{R}^n\}.$$

# Conic optimization (for symmetric cones)

## The natural extension of linear optimization

- Linear optimization over symmetric cones remains tractable.
- Same efficient interior-point methods as for LPs.
- Much more flexible than one might think!

Three symmetric cones:

- Linear cone $x \in \mathbb{R}^n_+$.
- Quadratic cones:

$$\mathcal{Q}^n := \{x \in \mathbb{R}^n \mid x_1 \geq \sqrt{x_2^2 + \cdots + x_n^2}\},$$
$$\mathcal{Q}_r^n := \{x \in \mathbb{R}^n \mid 2x_1x_2 \geq x_3^2 + \cdots + x_n^2,\ (x_1, x_2) \geq 0\}.$$

- Semidefinite cone:

$$\mathcal{S}_+^n := \{X \in \mathbb{R}^{n\times n} \mid X = X^T,\ z^T X z \geq 0, \forall z \in \mathbb{R}^n\}.$$

# Linear cone problems optimization

## Canonic form

We can write a linear cone problem as

$$
\begin{aligned}
\text{minimize} \quad & \langle c^l, x^l \rangle + \sum_{j=1}^{n_q} \langle c_j^q, x_j^q \rangle + \sum_{j=1}^{n_s} \langle C_j^s, X_j^s \rangle \\
\text{subject to} \quad & \langle a_i^l, x^l \rangle + \sum_{j=1}^{n_q} \langle a_{i,j}^q, x_j^q \rangle + \sum_{j=1}^{n_s} \langle A_{i,j}^s, X_j^s \rangle = b_i, \; i = 1, \ldots, m \\
& x^l \in \mathbb{R}_+^{n_l}, \; x_j^q \in \mathcal{Q}^{q_j}, \; X_j^s \in \mathcal{S}_+^{s_j}
\end{aligned}
$$

where

- $c^l, a_i^l, c_j^q, a_{i,j}^q$ are vectors,
- $C_j^s, A_{i,j}^s$ are symmetric matrices with inner-product

$$
\langle V, W \rangle := \mathrm{tr}(V^T W) = \sum_{ij} V_{ij} W_{ij} = \mathbf{vec}(V)^T \mathbf{vec}(W),
$$

If we define

$$A^T = \begin{pmatrix} a_1^l & \dots & a_m^l \\ a_{1,1}^q & \dots & a_{1,m}^q \\ \vdots & & \vdots \\ a_{1,n_q}^q & \dots & a_{m,n_q}^q \\ \mathbf{vec}(A_{1,1}^s) & \dots & \mathbf{vec}(A_{m,1}^s) \\ \vdots & & \vdots \\ \mathbf{vec}(A_{1,n_s}^s) & \dots & \mathbf{vec}(A_{m,n_s}^s) \end{pmatrix}, c = \begin{pmatrix} c^l \\ c_1^q \\ \vdots \\ c_{n_q}^q \\ \mathbf{vec}(C_1^s) \\ \vdots \\ \mathbf{vec}(C_{n_s}^s) \end{pmatrix}, x = \begin{pmatrix} x^l \\ x_1^q \\ \vdots \\ x_{n_q}^q \\ \mathbf{vec}(X_1^s) \\ \vdots \\ \mathbf{vec}(X_{n_s}^s) \end{pmatrix}$$

we get a lighter notation (SeDuMi format)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array}$$

where $\mathcal{K} = \mathbb{R}^{n_l} \times \mathcal{Q}^{q_1} \times \cdots \times \mathcal{Q}^{q_{n_q}} \times \mathcal{S}_+^{s_1} \times \cdots \times \mathcal{S}_+^{s_{n_s}}$.

# Duality in conic problems

## Almost as simple as for linear optimization

As for LPs, we have primal and dual problems:

$$
\begin{array}{ll}
\text{minimize} & c^T x \\
\text{subject to} & Ax = b \\
& x \in \mathcal{K}
\end{array}
\qquad\qquad
\begin{array}{ll}
\text{maximize} & b^T y \\
\text{subject to} & c - A^T y = s \\
& s \in \mathcal{K}.
\end{array}
$$

- Weak duality.

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

- Strong duality. If a strictly feasible point exists then

$$c^T x = b^T y.$$

- For linear problems, we only need feasibility for strong duality.

# Duality in conic problems

## Almost as simple as for linear optimization

As for LPs, we have primal and dual problems:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array} \qquad\qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & c - A^T y = s \\ & s \in \mathcal{K}. \end{array}$$

- Weak duality.

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

- Strong duality. If a strictly feasible point exists then

$$c^T x = b^T y.$$

- For linear problems, we only need feasibility for strong duality.

# Duality in conic problems

## Almost as simple as for linear optimization

As for LPs, we have primal and dual problems:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array} \qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & c - A^T y = s \\ & s \in \mathcal{K}. \end{array}$$

- Weak duality.

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

- Strong duality. If a strictly feasible point exists then

$$c^T x = b^T y.$$

- For linear problems, we only need feasibility for strong duality.

# Duality in conic problems

## Almost as simple as for linear optimization

As for LPs, we have primal and dual problems:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array} \qquad\qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & c - A^T y = s \\ & s \in \mathcal{K}. \end{array}$$

- Weak duality.

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

- Strong duality. If a strictly feasible point exists then

$$c^T x = b^T y.$$

- For linear problems, we only need feasibility for strong duality.

# Duality in conic problems

## An example with nonzero duality gap

- Consider the problem

$$\begin{array}{ll} \text{minimize} & x_1 \\ \text{subject to} & \begin{bmatrix} 0 & x_1 & 0 \\ x_1 & x_2 & 0 \\ 0 & 0 & 1+x_1 \end{bmatrix} \in \mathcal{S}^3_+, \end{array}$$

  with feasible set $\{x_1 = 0, x_2 \geq 0\}$ and optimal value $p^\star = 0$.

- The dual can be expressed as

$$\begin{array}{ll} \text{maximize} & -z_2 \\ \text{subject to} & \begin{bmatrix} z_1 & (1-z_2)/2 & 0 \\ (1-z_2)/2 & 0 & 0 \\ 0 & 0 & z_2 \end{bmatrix} \in \mathcal{S}^3_+, \end{array}$$

  with feasible set $\{z_1 \geq 0, z_2 = 1\}$ and optimal value $d^\star = -1$.

- Both problems are feasible, but not strictly feasible.

# Duality in conic problems

## An example with nonzero duality gap

- Consider the problem

$$\begin{array}{ll} \text{minimize} & x_1 \\ \text{subject to} & \begin{bmatrix} 0 & x_1 & 0 \\ x_1 & x_2 & 0 \\ 0 & 0 & 1+x_1 \end{bmatrix} \in \mathcal{S}_+^3, \end{array}$$

  with feasible set $\{x_1 = 0, x_2 \geq 0\}$ and optimal value $p^\star = 0$.

- The dual can be expressed as

$$\begin{array}{ll} \text{maximize} & -z_2 \\ \text{subject to} & \begin{bmatrix} z_1 & (1-z_2)/2 & 0 \\ (1-z_2)/2 & 0 & 0 \\ 0 & 0 & z_2 \end{bmatrix} \in \mathcal{S}_+^3, \end{array}$$

  with feasible set $\{z_1 \geq 0, z_2 = 1\}$ and optimal value $d^\star = -1$.

- Both problems are feasible, but not strictly feasible.

# Duality in conic problems

## An example with nonzero duality gap

- Consider the problem

$$\begin{array}{ll} \text{minimize} & x_1 \\ \text{subject to} & \begin{bmatrix} 0 & x_1 & 0 \\ x_1 & x_2 & 0 \\ 0 & 0 & 1+x_1 \end{bmatrix} \in \mathcal{S}^3_+, \end{array}$$

  with feasible set $\{x_1 = 0, x_2 \geq 0\}$ and optimal value $p^\star = 0$.
- The dual can be expressed as

$$\begin{array}{ll} \text{maximize} & -z_2 \\ \text{subject to} & \begin{bmatrix} z_1 & (1-z_2)/2 & 0 \\ (1-z_2)/2 & 0 & 0 \\ 0 & 0 & z_2 \end{bmatrix} \in \mathcal{S}^3_+, \end{array}$$

  with feasible set $\{z_1 \geq 0, z_2 = 1\}$ and optimal value $d^\star = -1$.
- Both problems are feasible, but not strictly feasible.

## Farka's lemma

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array} \qquad\qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & c - A^T y = s \\ & s \in \mathcal{K}. \end{array}$$

### Farka's lemma

Given $A$ and $b$, exactly one of the two statements are true:

1. There exists an $x \in \mathcal{K}$ such that $Ax = b$.
2. There exists a $y$ such that $-A^T y \in \mathcal{K}$ and $b^T y > 0$.

- The certificate $y$ is an unbounded direction for the dual.
- If 2) then for $t$ large enough we have

$$c - tA^T y \in \mathcal{K}.$$

while $tb^T y \to \infty$ as $t \to \infty$.

# Infeasibility in conic problems

## Farka's lemma

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array} \qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & c - A^T y = s \\ & s \in \mathcal{K}. \end{array}$$

### Farka's lemma

Given $A$ and $b$, exactly one of the two statements are true:

1. There exists an $x \in \mathcal{K}$ such that $Ax = b$.
2. There exists a $y$ such that $-A^T y \in \mathcal{K}$ and $b^T y > 0$.

- The certificate $y$ is an unbounded direction for the dual.
- If 2) then for $t$ large enough we have

$$c - tA^T y \in \mathcal{K}.$$

while $tb^T y \to \infty$ as $t \to \infty$.

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \end{array} \qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & c - A^T y = s \\ & s \in \mathcal{K}. \end{array}$$

### Farka's lemma (dual variant)

Given $A$ and $b$, exactly one of the two statements are true:

1. There exists a $y$ such that $c - A^T y \in \mathcal{K}$.
2. There exists an $x \in \mathcal{K}$ such that $Ax = 0$ and $c^T x < 0$.

Either the problem is dual feasible, or $x$ is an unbounded direction for the primal problem.

# Infeasibility in conic problems

## An example with primal and dual infeasibility

Consider

$$\begin{array}{ll} \text{minimize} & -x_1 - x_2 \\ \text{subject to} & x_1 = -1 \\ & x_1,\, x_2 \geq 0 \end{array}$$

with a dual problem

$$\begin{array}{ll} \text{maximize} & -y \\ \text{subject to} & \left[\begin{array}{c} 1 \\ 0 \end{array}\right] y \leq \left[\begin{array}{c} -1 \\ -1 \end{array}\right]. \end{array}$$

- Both problems are trivially infeasible.
- $y = -1$ is a certificate of primal infeasibility.
- $x = (0, 1)$ is a certificate of dual infeasiblity.

# The homogeneous model for conic problems

The homogenous model beautifully encapsulates all cases:

$$\begin{bmatrix} s \\ 0 \\ \kappa \end{bmatrix} + \begin{bmatrix} 0 & A^T & -c \\ A & 0 & -b \\ c^T & -b^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \tau \end{bmatrix} = 0, \quad x, s \in \mathcal{K},\ \tau, \kappa \geq 0.$$

- If $\tau > 0$, $\kappa = 0$ then $\frac{1}{\tau}(x, y, s)$ is optimal,

$$Ax = b\tau, \quad c\tau - A^T y = s, \quad c^T x - b^T y = x^T s = 0.$$

- If $\tau = 0$, $\kappa > 0$ then the problem is infeasible,

$$Ax = 0, \quad -A^T y = s, \quad c^T x - b^T y < 0.$$

- If $\tau = 0$, $\kappa = 0$ then the problem is ill-posed.

# The homogeneous model for conic problems

The homogenous model beautifully encapsulates all cases:

$$\begin{bmatrix} s \\ 0 \\ \kappa \end{bmatrix} + \begin{bmatrix} 0 & A^T & -c \\ A & 0 & -b \\ c^T & -b^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \tau \end{bmatrix} = 0, \quad x, s \in \mathcal{K}, \; \tau, \kappa \geq 0.$$

- If $\tau > 0$, $\kappa = 0$ then $\frac{1}{\tau}(x, y, s)$ is optimal,

$$Ax = b\tau, \quad c\tau - A^T y = s, \quad c^T x - b^T y = x^T s = 0.$$

- If $\tau = 0$, $\kappa > 0$ then the problem is infeasible,

$$Ax = 0, \quad -A^T y = s, \quad c^T x - b^T y < 0.$$

- If $\tau = 0$, $\kappa = 0$ then the problem is ill-posed.

# The homogeneous model for conic problems

The homogenous model beautifully encapsulates all cases:

$$\begin{bmatrix} s \\ 0 \\ \kappa \end{bmatrix} + \begin{bmatrix} 0 & A^T & -c \\ A & 0 & -b \\ c^T & -b^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \tau \end{bmatrix} = 0, \quad x, s \in \mathcal{K},\ \tau, \kappa \geq 0.$$

- If $\tau > 0$, $\kappa = 0$ then $\dfrac{1}{\tau}(x, y, s)$ is optimal,

$$Ax = b\tau, \quad c\tau - A^T y = s, \quad c^T x - b^T y = x^T s = 0.$$

- If $\tau = 0$, $\kappa > 0$ then the problem is infeasible,

$$Ax = 0, \quad -A^T y = s, \quad c^T x - b^T y < 0.$$

- If $\tau = 0$, $\kappa = 0$ then the problem is ill-posed.

# The homogeneous model for conic problems

The homogenous model beautifully encapsulates all cases:

$$\begin{bmatrix} s \\ 0 \\ \kappa \end{bmatrix} + \begin{bmatrix} 0 & A^T & -c \\ A & 0 & -b \\ c^T & -b^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \tau \end{bmatrix} = 0, \quad x, s \in \mathcal{K}, \; \tau, \kappa \geq 0.$$

- If $\tau > 0$, $\kappa = 0$ then $\frac{1}{\tau}(x, y, s)$ is optimal,

$$Ax = b\tau, \quad c\tau - A^T y = s, \quad c^T x - b^T y = x^T s = 0.$$

- If $\tau = 0$, $\kappa > 0$ then the problem is infeasible,

$$Ax = 0, \quad -A^T y = s, \quad c^T x - b^T y < 0.$$

- If $\tau = 0$, $\kappa = 0$ then the problem is ill-posed.

Section 2

# Conic quadratic modeling

$$\mathcal{Q}^n := \{x \in \mathbb{R}^n \mid x_1 \geq \sqrt{x_2^2 + \cdots + x_n^2}\},$$
$$\mathcal{Q}_r^n := \{x \in \mathbb{R}^n \mid 2x_1 x_2 \geq x_3^2 + \cdots + x_n^2,\ (x_1, x_2) \geq 0\}.$$

- Absolute values:
$$|x| \leq t \quad \Longleftrightarrow \quad (t, x) \in \mathcal{Q}^2.$$

- Euclidean norms $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$:
$$\|x\|_2 \leq t \quad \Longleftrightarrow \quad (t, x) \in \mathcal{Q}^{n+1}.$$

- Squared Euclidean norms:
$$\|x\|_2^2 \leq t \quad \Longleftrightarrow \quad (1/2, t, x) \in \mathcal{Q}_r^{n+2}.$$

$$\mathcal{Q}^n := \{x \in \mathbb{R}^n \mid x_1 \geq \sqrt{x_2^2 + \cdots + x_n^2}\},$$
$$\mathcal{Q}_r^n := \{x \in \mathbb{R}^n \mid 2x_1x_2 \geq x_3^2 + \cdots + x_n^2,\ (x_1, x_2) \geq 0\}.$$

- Absolute values:
$$|x| \leq t \quad \iff \quad (t, x) \in \mathcal{Q}^2.$$

- Euclidean norms $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$:
$$\|x\|_2 \leq t \quad \iff \quad (t, x) \in \mathcal{Q}^{n+1}.$$

- Squared Euclidean norms:
$$\|x\|_2^2 \leq t \quad \iff \quad (1/2, t, x) \in \mathcal{Q}_r^{n+2}.$$

$$\mathcal{Q}^n := \{x \in \mathbb{R}^n \mid x_1 \geq \sqrt{x_2^2 + \cdots + x_n^2}\},$$
$$\mathcal{Q}_r^n := \{x \in \mathbb{R}^n \mid 2x_1x_2 \geq x_3^2 + \cdots + x_n^2,\ (x_1, x_2) \geq 0\}.$$

- Absolute values:

$$|x| \leq t \quad \Longleftrightarrow \quad (t, x) \in \mathcal{Q}^2.$$

- Euclidean norms $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$:

$$\|x\|_2 \leq t \quad \Longleftrightarrow \quad (t, x) \in \mathcal{Q}^{n+1}.$$

- Squared Euclidean norms:

$$\|x\|_2^2 \leq t \quad \Longleftrightarrow \quad (1/2, t, x) \in \mathcal{Q}_r^{n+2}.$$

We call $\mathcal{Q}_r$ the *rotated quadratic cone*. Let

$$T_n := \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & I_{n-2} \end{bmatrix}.$$

Then it is easy to verify that

$$x \in \mathcal{Q}^n \quad \Longleftrightarrow \quad (T_n x) \in \mathcal{Q}_r^n.$$

- $T_n$ corresponds to an orthogonal transformation.
- Many sets are naturally characterized using rotated cones.
- Computational advantages by explicitly handling rotated cones.

We call $\mathcal{Q}_r$ the *rotated quadratic cone*. Let

$$T_n := \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & I_{n-2} \end{bmatrix}.$$

Then it is easy to verify that

$$x \in \mathcal{Q}^n \iff (T_n x) \in \mathcal{Q}_r^n.$$

- $T_n$ corresponds to an orthogonal transformation.
- Many sets are naturally characterized using rotated cones.
- Computational advantages by explicitly handling rotated cones.

We call $\mathcal{Q}_r$ the *rotated quadratic cone*. Let

$$T_n := \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & I_{n-2} \end{bmatrix}.$$

Then it is easy to verify that

$$x \in \mathcal{Q}^n \quad \Longleftrightarrow \quad (T_n x) \in \mathcal{Q}_r^n.$$

- $T_n$ corresponds to an orthogonal transformation.
- Many sets are naturally characterized using rotated cones.
- Computational advantages by explicitly handling rotated cones.

We call $\mathcal{Q}_r$ the *rotated quadratic cone*. Let

$$T_n := \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & I_{n-2} \end{bmatrix}.$$

Then it is easy to verify that

$$x \in \mathcal{Q}^n \quad \iff \quad (T_n x) \in \mathcal{Q}_r^n.$$

- $T_n$ corresponds to an orthogonal transformation.
- Many sets are naturally characterized using rotated cones.
- Computational advantages by explicitly handling rotated cones.

# Other simple quadratic represetable sets

- Hyperbolic sets $\left\{(t, x) \mid t \geq \frac{1}{x},\, x > 0\right\}$:

$$(x, t, \sqrt{2}) \in \mathcal{Q}_r^3 \quad \Leftrightarrow \quad 2xt \geq 2,\, x, t \geq 0 \quad \Leftrightarrow \quad t \geq \frac{1}{x},\, x > 0.$$

- Square roots $\{(t, x) \mid t \leq \sqrt{x},\, x \geq 0\}$:

$$(\frac{1}{2}, x, t) \in \mathcal{Q}_r^3 \quad \Leftrightarrow \quad x \geq t^2,\, x \geq 0 \quad \Leftrightarrow \quad \sqrt{x} \geq t,\, x \geq 0.$$

- Simple rational $\left\{(t, x) \mid t \geq \frac{1}{x^2},\, x > 0\right\}$:

$$\begin{aligned} & (t, \frac{1}{2}, s), (x, s, \sqrt{2}) \in \mathcal{Q}_r^3 \\ \Leftrightarrow \quad & t \geq s^2,\, 2xs \geq 2,\, s, t, x \geq 0 \\ \Leftrightarrow \quad & t \geq \frac{1}{x^2},\, x > 0. \end{aligned}$$

# Other simple quadratic represetable sets

- Hyperbolic sets $\left\{(t,x) \mid t \geq \frac{1}{x},\, x > 0\right\}$:

$$(x, t, \sqrt{2}) \in \mathcal{Q}_r^3 \quad \Leftrightarrow \quad 2xt \geq 2,\ x, t \geq 0 \quad \Leftrightarrow \quad t \geq \frac{1}{x},\ x > 0.$$

- Square roots $\{(t,x) \mid t \leq \sqrt{x},\, x \geq 0\}$:

$$(\frac{1}{2}, x, t) \in \mathcal{Q}_r^3 \quad \Leftrightarrow \quad x \geq t^2,\ x \geq 0 \quad \Leftrightarrow \quad \sqrt{x} \geq t,\ x \geq 0.$$

- Simple rational $\left\{(t,x) \mid t \geq \frac{1}{x^2},\, x > 0\right\}$:

$$\begin{aligned} & (t, \frac{1}{2}, s), (x, s, \sqrt{2}) \in \mathcal{Q}_r^3 \\ \Leftrightarrow \quad & t \geq s^2,\ 2xs \geq 2,\ s, t, x \geq 0 \\ \Leftrightarrow \quad & t \geq \frac{1}{x^2},\ x > 0. \end{aligned}$$

# Other simple quadratic represetable sets

- Hyperbolic sets $\left\{(t,x) \mid t \geq \frac{1}{x},\ x > 0\right\}$:

$$(x, t, \sqrt{2}) \in \mathcal{Q}_r^3 \quad \Leftrightarrow \quad 2xt \geq 2,\ x, t \geq 0 \quad \Leftrightarrow \quad t \geq \frac{1}{x},\ x > 0.$$

- Square roots $\left\{(t,x) \mid t \leq \sqrt{x},\ x \geq 0\right\}$:

$$(\frac{1}{2}, x, t) \in \mathcal{Q}_r^3 \quad \Leftrightarrow \quad x \geq t^2,\ x \geq 0 \quad \Leftrightarrow \quad \sqrt{x} \geq t,\ x \geq 0.$$

- Simple rational $\left\{(t,x) \mid t \geq \frac{1}{x^2},\ x > 0\right\}$:

$$\begin{aligned} & (t, \frac{1}{2}, s), (x, s, \sqrt{2}) \in \mathcal{Q}_r^3 \\ \Leftrightarrow \quad & t \geq s^2,\ 2xs \geq 2,\ s, t, x \geq 0 \\ \Leftrightarrow \quad & t \geq \frac{1}{x^2},\ x > 0. \end{aligned}$$

We can characterize a convex quadratic inequality

$$\frac{1}{2}x^T Q x - c^T x - r \leq 0$$

equivalently as

$$\frac{1}{2}x^T F^T F x \leq c^T x + r,$$

where $Q = F^T F \succeq 0$, $F \in \mathbb{R}^{k \times n}$, resulting in

$$(1, c^T x + r, Fx) \in \mathcal{Q}_r^{2+k}.$$

So we can write convex QPs/QCQPs as quadratic cone problems.

A second-order cone is occasionally specified as

$$\|Ax + b\|_2 \leq c^T x + d$$

where $A \in \mathbb{R}^{m \times n}$ and $c \in \mathbb{R}^n$. This is quivalent to

$$(Ax + b, c^T x + d) \in \mathcal{Q}^{m+1},$$

corresponding to our dual form (conic membership of an affine expression).

# Ellipsoidal sets

Ellipsoid centered at $r$:

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid y = P(x - r),\ \|y\|_2 \leq 1\}.$$

Suppose $P \succ 0$. We then have an alternatively characterization

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid x = P^{-1}y + r,\ \|y\|_2 \leq 1\}.$$

Useful for robustifying linear models, e.g.,

$$\begin{array}{ll} \text{minimize} & \sup_{c \in \mathcal{E}} c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array} \qquad \begin{array}{ll} \text{minimize} & r^T x + t \\ \text{subject to} & Ax = b \\ & (t, P^{-1}x) \in \mathcal{Q}^{n+1} \\ & x \geq 0 \end{array}$$

**Exercise:** Derive the robust quadratic cone problem.

# Ellipsoidal sets

Ellipsoid centered at $r$:

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid y = P(x - r), \ \|y\|_2 \leq 1\}.$$

Suppose $P \succ 0$. We then have an alternatively characterization

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid x = P^{-1}y + r, \ \|y\|_2 \leq 1\}.$$

Useful for robustifying linear models, e.g.,

$$\begin{array}{ll} \text{minimize} & \sup_{c \in \mathcal{E}} c^T x \\ \text{subject to} & Ax = b \\ & x \geq 0 \end{array} \qquad \begin{array}{ll} \text{minimize} & r^T x + t \\ \text{subject to} & Ax = b \\ & (t, P^{-1}x) \in \mathcal{Q}^{n+1} \\ & x \geq 0 \end{array}$$

**Exercise:** Derive the robust quadratic cone problem.

# Markowitz portfolio optimization

## Minimum-risk formulation

The *minimum-risk* problem is a standard quadratic problem

$$\begin{array}{ll} \text{minimize} & x^T \Sigma x \\ \text{subject to} & e^T x = 1 \\ & \mu^T x \geq \rho \\ & x \geq 0. \end{array}$$

where

- $x$ is the allocation of $n$ assets,
- $\mu$ is the vector of average return for the assets,
- $\Sigma \succeq 0$ is the covariance of the assets,
- we have a budget constraint $e^T x = 1$,
- we require a minimum return of investment of $\rho$.

Often we also have semi-continuous *threshold constraints*,

$$x_i \in 0 \cup [l_i, u_i]$$

which gives a much harder MI-QP

$$\begin{array}{ll} \text{minimize} & x^T \Sigma x \\ \text{subject to} & e^T x = 1 \\ & \mu^T x \geq \rho \\ & l_i y_i \leq x_i \leq u_i y_i, \ i = 1, \dots, n \\ & y \in \{0, 1\}^n. \end{array}$$

The perspective reformulation can be written as

$$\begin{array}{ll} \text{minimize} & x^T(\Sigma - \mathbf{Diag}(d))x + d^T\phi \\ \text{subject to} & x_i^2 \leq y_i\phi_i \\ & e^T x = 1 \\ & \mu^T x \geq \rho \\ & l_i y_i \leq x_i \leq u_i y_i, \; i = 1, \dots, n \\ & \phi \geq 0 \\ & y \in \{0,1\}^n. \end{array}$$

for some $d > 0$ with $\Sigma \succeq \mathbf{Diag}(d)$.

- $\phi_i = x_i^2/y_i$ at optimality (perspective function).
- Same feasible set.
- Continuous relaxation with $0 \leq y \leq e$ is tighter.
- We discuss how to choose $d$ later (using SDO).

The perspective reformulation can be written as

$$\begin{array}{ll} \text{minimize} & x^T(\Sigma - \mathbf{Diag}(d))x + d^T\phi \\ \text{subject to} & x_i^2 \le y_i\phi_i \\ & e^T x = 1 \\ & \mu^T x \ge \rho \\ & l_i y_i \le x_i \le u_i y_i, \; i = 1, \dots, n \\ & \phi \ge 0 \\ & y \in \{0,1\}^n. \end{array}$$

for some $d > 0$ with $\Sigma \succeq \mathbf{Diag}(d)$.

- $\phi_i = x_i^2 / y_i$ at optimality (perspective function).
- Same feasible set.
- Continuous relaxation with $0 \le y \le e$ is tighter.
- We discuss how to choose $d$ later (using SDO).

# Markowitz portfolio optimization

## Perspective reformulation

The perspective reformulation can be written as

$$
\begin{array}{ll}
\text{minimize} & x^T(\Sigma - \mathbf{Diag}(d))x + d^T\phi \\
\text{subject to} & x_i^2 \leq y_i \phi_i \\
& e^T x = 1 \\
& \mu^T x \geq \rho \\
& l_i y_i \leq x_i \leq u_i y_i, \; i = 1, \ldots, n \\
& \phi \geq 0 \\
& y \in \{0, 1\}^n.
\end{array}
$$

for some $d > 0$ with $\Sigma \succeq \mathbf{Diag}(d)$.

- $\phi_i = x_i^2 / y_i$ at optimality (perspective function).
- Same feasible set.
- Continuous relaxation with $0 \leq y \leq e$ is tighter.
- We discuss how to choose $d$ later (using SDO).

The perspective reformulation can be written as

$$
\begin{array}{ll}
\text{minimize} & x^T(\Sigma - \mathbf{Diag}(d))x + d^T\phi \\
\text{subject to} & x_i^2 \leq y_i \phi_i \\
& e^T x = 1 \\
& \mu^T x \geq \rho \\
& l_i y_i \leq x_i \leq u_i y_i, \ i = 1, \ldots, n \\
& \phi \geq 0 \\
& y \in \{0, 1\}^n.
\end{array}
$$

for some $d > 0$ with $\Sigma \succeq \mathbf{Diag}(d)$.

- $\phi_i = x_i^2 / y_i$ at optimality (perspective function).
- Same feasible set.
- Continuous relaxation with $0 \leq y \leq e$ is tighter.
- We discuss how to choose $d$ later (using SDO).

The perspective reformulation can be written as

$$\begin{array}{ll}
\text{minimize} & x^T(\Sigma - \textbf{Diag}(d))x + d^T\phi \\
\text{subject to} & x_i^2 \leq y_i\phi_i \\
& e^T x = 1 \\
& \mu^T x \geq \rho \\
& l_i y_i \leq x_i \leq u_i y_i, \; i = 1, \ldots, n \\
& \phi \geq 0 \\
& y \in \{0,1\}^n.
\end{array}$$

for some $d > 0$ with $\Sigma \succeq \textbf{Diag}(d)$.

- $\phi_i = x_i^2/y_i$ at optimality (perspective function).
- Same feasible set.
- Continuous relaxation with $0 \leq y \leq e$ is tighter.
- We discuss how to choose $d$ later (using SDO).

Let $V^T V = \Sigma - \mathbf{Diag}(d)$, and observe that

$$\begin{aligned} & l_i y_i \leq x_i \leq u_i y_i \\ \Leftrightarrow \quad & (u_i y_i - x_i)(l_i y_i - x_i) \leq 0 \\ \Leftrightarrow \quad & x_i^2 - (u_i + l_i) x_i y_i + u_i l_i y_i^2 \leq 0. \end{aligned}$$

We then get a conic MIP:

$$\begin{array}{ll} \text{minimize} & \alpha + d^T \phi \\ \text{subject to} & ((1/2), \alpha, Vx) \in \mathcal{Q}_r^{n+2} \\ & ((1/2)\phi_i, y_i, x_i) \in \mathcal{Q}_r^3, \ i = 1, \ldots, n \\ & e^T x = 1 \\ & \mu^T x \geq \rho \\ & \phi_i - (u_i + l_i) x_i + u_i l_i y_i \leq 0, \ i = 1, \ldots, n \\ & y \in \{0, 1\}^n. \end{array}$$

Let $V^T V = \Sigma - \mathbf{Diag}(d)$, and observe that

$$\begin{aligned} & l_i y_i \le x_i \le u_i y_i \\ \Leftrightarrow \quad & (u_i y_i - x_i)(l_i y_i - x_i) \le 0 \\ \Leftrightarrow \quad & x_i^2 - (u_i + l_i) x_i y_i + u_i l_i y_i^2 \le 0. \end{aligned}$$

We then get a conic MIP:

$$\begin{array}{ll} \text{minimize} & \alpha + d^T \phi \\ \text{subject to} & ((1/2), \alpha, Vx) \in \mathcal{Q}_r^{n+2} \\ & ((1/2)\phi_i, y_i, x_i) \in \mathcal{Q}_r^3, \; i = 1, \ldots, n \\ & e^T x = 1 \\ & \mu^T x \ge \rho \\ & \phi_i - (u_i + l_i) x_i + u_i l_i y_i \le 0, \; i = 1, \ldots, n \\ & y \in \{0, 1\}^n. \end{array}$$

# Markowitz portfolio optimization

## A Python Fusion implementation

```
def perspective(V, d, mu, r, l, u):

  with Model('Perspective') as M:
    a   = M.variable('a',   1,       Domain.greaterThan(0.0))
    x   = M.variable('x',   len(mu), Domain.greaterThan(0.0))
    phi = M.variable('phi', len(mu), Domain.greaterThan(0.0))
    y   = M.variable('y',   len(mu), Domain.binary())

    # (0.5*phi_i, y_i, x_i) \in Qr
    M.constraint(Expr.hstack(Expr.mul(0.5, phi), y, x), Domain.inRotatedQCone())

    # sum(x) == 1
    M.constraint(Expr.sum(x), Domain.equalsTo(1.0))

    # mu'*x >= r
    M.constraint(Expr.dot(mu,x), Domain.greaterThan(r))

    # phi - (L+U)*x + L*U*y <= 0
    M.constraint(Expr.add(Expr.sub(phi, Expr.mulElm(l+u,x)), Expr.mulElm(l*u,y)), Domain.lessThan(0.0))

    # (0.5, a, V*x) \in Qr
    M.constraint(Expr.vstack(0.5, a, Expr.mul(V, x)), Domain.inRotatedQCone())

    # minimize a + d'*phi
    M.objective('obj', ObjectiveSense.Minimize, Expr.add(a, Expr.dot(d, phi)))

    M.setLogHandler(sys.stdout)
    M.solve()
    return x.level()
```

Data from Frangioni and Gentile ($n = 200$):
`http://www.di.unipi.it/optimize/Data/MV.html`

```
BRANCHES RELAXS   ACT_NDS  DEPTH    BEST_INT_OBJ         BEST_RELAX_OBJ       REL_GAP(%)  TIME
0        1        0        0        NA                   4.5946919096e+01     NA          1.7
Cut generation started.
0        2        0        0        NA                   4.5946919512e+01     NA          1.8
Cut generation terminated. Time = 0.20
0        3        1        0        NA                   4.5946919512e+01     NA          2.2
0        3        1        0        NA                   4.5946919512e+01     NA          2.5
0        3        1        0        2.4617106593e+02     4.5946919512e+01     81.34       2.7
0        3        1        0        2.4617106593e+02     4.5946919512e+01     81.34       2.9
1        4        2        0        2.4617106593e+02     4.5946919512e+01     81.34       3.0
3        6        4        1        2.4617106593e+02     4.7251665363e+01     80.81       3.1
7        10       8        2        2.4617106593e+02     4.9007463791e+01     80.09       3.2
15       18       16       3        2.4617106593e+02     5.0468332901e+01     79.50       3.3
31       34       32       4        2.4617106593e+02     5.2337435796e+01     78.74       3.5
.
.
.
452235   452313   240764   79       2.2214876543e+02     1.3985833801e+02     37.04       3589.8
452754   452832   240979   92       2.2214876543e+02     1.3985833801e+02     37.04       3593.4
453275   453353   241202   106      2.2214876543e+02     1.3989876667e+02     37.02       3596.9
```

Timeout after 1 hour.

# Markowitz portfolio optimization

## Solving the perspective reformulation

Data from Frangioni and Gentile ($n = 200$):
http://www.di.unipi.it/optimize/Data/MV.html

```
BRANCHES RELAXS   ACT_NDS  DEPTH    BEST_INT_OBJ         BEST_RELAX_OBJ       REL_GAP(%)  TIME
0        1        0        0        NA                   2.0508019535e+02     NA          2.3
Cut generation started.
0        2        0        0        NA                   2.0508019535e+02     NA          2.5
Cut generation terminated. Time = 0.36
0        3        1        0        2.2042170197e+02     2.0508019535e+02     6.96        6.6
0        3        1        0        2.2042170197e+02     2.0508019535e+02     6.96        7.1
0        3        1        0        2.2042170197e+02     2.0508019535e+02     6.96        7.3
0        3        1        0        2.2042170197e+02     2.0508019535e+02     6.96        7.6
1        4        2        0        2.2042170197e+02     2.0508019535e+02     6.96        7.8
3        6        4        1        2.2042170197e+02     2.0517416168e+02     6.92        8.0
7        10       8        2        2.2042170197e+02     2.0529861536e+02     6.86        8.2
15       18       16       3        2.2042170197e+02     2.0530565821e+02     6.86        9.0
31       34       32       4        2.2042170197e+02     2.0531091010e+02     6.86        10.5
51       54       52       5        2.2042170197e+02     2.0538010066e+02     6.82        12.5
91       94       92       6        2.2042170197e+02     2.0539391269e+02     6.82        16.1
171      174      168      9        2.2042170197e+02     2.0555122515e+02     6.75        23.4
331      334      292      14       2.2042170197e+02     2.0624274122e+02     6.43        38.1
611      613      362      21       2.2042170197e+02     2.0624274122e+02     6.43        65.5
966      968      477      30       2.2042170197e+02     2.0624274122e+02     6.43        98.7
1410     1413     591      42       2.2042170197e+02     2.0639253138e+02     6.36        136.7
1862     1861     711      54       2.0672595157e+02     2.0660993751e+02     0.06        173.7
2561     1868     14       24       2.0672595157e+02     2.0672595157e+02     0.00e+00    175.0
```

Section 3

# Semidefinite modeling

# Positive semidefinite matrices

## Some well-known facts

A symmetric matrix $X \in \mathbb{R}^{n\times n}$ is positive semidefinite iff:

1. $z^T X z \geq 0,\ \forall z \in \mathbb{R}^n$.
2. All the eigenvalues of $X$ are nonnegative.
3. $X$ has a Grammian representation, $X = V^T V$.

**Exercise:** Show 1) $\Leftrightarrow$ 2) $\Leftrightarrow$ 3) given an eigen-decomposition $X = \sum_{i=1}^{n} \lambda_i q_i q_i^T$.

Schur's lemma is another useful result. A symmetric matrix

$$A = \begin{pmatrix} B & C^T \\ C & D \end{pmatrix}$$

is positive definite iff

$$B - C^T D^{-1} C \succ 0, \quad C \succ 0, \quad D \succ 0.$$

# Positive semidefinite matrices

## Some well-known facts

A symmetric matrix $X \in \mathbb{R}^{n \times n}$ is positive semidefinite iff:

1. $z^T X z \geq 0, \; \forall z \in \mathbb{R}^n$.
2. All the eigenvalues of $X$ are nonnegative.
3. $X$ has a Grammian representation, $X = V^T V$.

**Exercise:** Show 1) $\Leftrightarrow$ 2) $\Leftrightarrow$ 3) given an eigen-decomposition $X = \sum_{i=1}^{n} \lambda_i q_i q_i^T$.

Schur's lemma is another useful result. A symmetric matrix

$$A = \begin{pmatrix} B & C^T \\ C & D \end{pmatrix}$$

is positive definite iff

$$B - C^T D^{-1} C \succ 0, \quad C \succ 0, \quad D \succ 0.$$

- The linear cone $x \geq 0$ corresponds to $x \in \mathcal{S}_+^1$.
- Two-dimensional matrices,

$$X = \begin{bmatrix} x_1 & x_3 \\ x_3 & x_2 \end{bmatrix} \in \mathcal{S}_+^2 \quad \Longleftrightarrow \quad x_1 x_2 \geq x_3^2,\ x_1, x_2 \geq 0,$$

  or $(x_1, x_2, x_3/\sqrt{2}) \in \mathcal{Q}_r^3$.
- Quadratic cones,

$$(t, x) \in \mathcal{Q}^{n+1} \quad \Longleftrightarrow \quad \begin{bmatrix} t & x^T \\ x & tI \end{bmatrix} \in \mathcal{S}_+^{n+1}.$$

- The linear cone $x \geq 0$ corresponds to $x \in \mathcal{S}_+^1$.
- Two-dimensional matrices,

$$X = \left[\begin{array}{cc} x_1 & x_3 \\ x_3 & x_2 \end{array}\right] \in \mathcal{S}_+^2 \quad \Longleftrightarrow \quad x_1 x_2 \geq x_3^2,\ x_1, x_2 \geq 0,$$

or $(x_1, x_2, x_3/\sqrt{2}) \in \mathcal{Q}_r^3$.

- Quadratic cones,

$$(t, x) \in \mathcal{Q}^{n+1} \quad \Longleftrightarrow \quad \left[\begin{array}{cc} t & x^T \\ x & tI \end{array}\right] \in \mathcal{S}_+^{n+1}.$$

- The linear cone $x \geq 0$ corresponds to $x \in \mathcal{S}_+^1$.
- Two-dimensional matrices,

$$X = \left[\begin{array}{cc} x_1 & x_3 \\ x_3 & x_2 \end{array}\right] \in \mathcal{S}_+^2 \quad \Longleftrightarrow \quad x_1 x_2 \geq x_3^2,\ x_1, x_2 \geq 0,$$

or $(x_1, x_2, x_3/\sqrt{2}) \in \mathcal{Q}_r^3$.

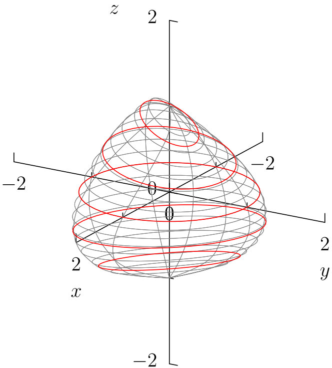
- Quadratic cones,

$$(t, x) \in \mathcal{Q}^{n+1} \quad \Longleftrightarrow \quad \left[\begin{array}{cc} t & x^T \\ x & tI \end{array}\right] \in \mathcal{S}_+^{n+1}.$$

The convex set

$$S = \left\{ (x, y, z) \in \mathbb{R}^3 \mid \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \in \mathcal{S}^3_+ \right\},$$

is called a *pillow*,




**Exercise:** Characterize the restriction $S|_{z=0}$.

## Eigenvalue optimization

### Symmetric matrices

$$F(x) = F_0 + x_1 F_1 + \cdots + x_m F_m, \quad F_i \in \mathcal{S}_m.$$

- Minimize largest eigenvalue $\lambda_1(F(x))$:

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \gamma I \succeq F(x), \end{array}$$

- Maximize smallest eigenvalue $\lambda_n(F(x))$:

$$\begin{array}{ll} \text{maximize} & \gamma \\ \text{subject to} & F(x) \succeq \gamma I, \end{array}$$

- Minimize eigenvalue spread $\lambda_1(F(x)) - \lambda_n(F(x))$:

$$\begin{array}{ll} \text{minimize} & \gamma - \lambda \\ \text{subject to} & \gamma I \succeq F(x) \succeq \lambda I, \end{array}$$

# Eigenvalue optimization

## Symmetric matrices

$$F(x) = F_0 + x_1 F_1 + \cdots + x_m F_m, \quad F_i \in \mathcal{S}_m.$$

- Minimize largest eigenvalue $\lambda_1(F(x))$:

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \gamma I \succeq F(x), \end{array}$$

- Maximize smallest eigenvalue $\lambda_n(F(x))$:

$$\begin{array}{ll} \text{maximize} & \gamma \\ \text{subject to} & F(x) \succeq \gamma I, \end{array}$$

- Minimize eigenvalue spread $\lambda_1(F(x)) - \lambda_n(F(x))$:

$$\begin{array}{ll} \text{minimize} & \gamma - \lambda \\ \text{subject to} & \gamma I \succeq F(x) \succeq \lambda I, \end{array}$$

# Eigenvalue optimization

## Symmetric matrices

$$F(x) = F_0 + x_1 F_1 + \cdots + x_m F_m, \quad F_i \in \mathcal{S}_m.$$

- Minimize largest eigenvalue $\lambda_1(F(x))$:

$$\begin{array}{ll} \text{minimize} & \gamma \\ \text{subject to} & \gamma I \succeq F(x), \end{array}$$

- Maximize smallest eigenvalue $\lambda_n(F(x))$:

$$\begin{array}{ll} \text{maximize} & \gamma \\ \text{subject to} & F(x) \succeq \gamma I, \end{array}$$

- Minimize eigenvalue spread $\lambda_1(F(x)) - \lambda_n(F(x))$:

$$\begin{array}{ll} \text{minimize} & \gamma - \lambda \\ \text{subject to} & \gamma I \succeq F(x) \succeq \lambda I, \end{array}$$

# Matrix norms
## Nonsymmetric matrices

$$F(x) = F_0 + x_1 F_1 + \cdots + x_m F_m, \quad F_i \in \mathbb{R}^{n \times p}.$$

- Frobenius norm: $\|F(x)\|_F := \sqrt{\langle F(x), F(x) \rangle}$,

$$\|F(x)\|_F \le t \quad \Leftrightarrow \quad (t, \mathbf{vec}(F(x))) \in \mathcal{Q}^{np+1},$$

- Induced $\ell_2$ norm: $\|F(x)\|_2 := \max_k \sigma_k(F(x))$,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} tI & F(x)^T \\ F(x) & tI \end{bmatrix} \succeq 0, \end{array}$$

corresponds to the largest eigenvalue for $F(x) \in \mathcal{S}_+^n$.

# Matrix norms
## Nonsymmetric matrices

$$F(x) = F_0 + x_1 F_1 + \cdots + x_m F_m, \quad F_i \in \mathbb{R}^{n \times p}.$$

- Frobenius norm: $\|F(x)\|_F := \sqrt{\langle F(x), F(x) \rangle}$,

$$\|F(x)\|_F \le t \quad \Leftrightarrow \quad (t, \mathbf{vec}(F(x))) \in \mathcal{Q}^{np+1},$$

- Induced $\ell_2$ norm: $\|F(x)\|_2 := \max_k \sigma_k(F(x))$,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{cc} tI & F(x)^T \\ F(x) & tI \end{array}\right] \succeq 0, \end{array}$$

corresponds to the largest eigenvalue for $F(x) \in \mathcal{S}_+^n$.

# Combinatorial relaxations

We consider a binary problem

$$\begin{array}{ll} \text{minimize} & x^T Q x + c^T x \\ \text{subject to} & x_i \in \{0,1\}, \quad i = 1,\dots,n. \end{array}$$

where $Q \in \mathcal{S}^n$ can be indefinite.

- Rewrite binary constraints $x_i \in \{0,1\}$:

$$x_i^2 = x_i \quad \Longleftrightarrow \quad X = xx^T, \quad \mathbf{diag}(X) = x.$$

  Still non-convex, since $\mathbf{rank}(X) = 1$.
- Semidefinite relaxation:

$$X \succeq xx^T, \quad \mathbf{diag}(X) = x.$$

# Combinatorial relaxations

We consider a binary problem

$$\begin{array}{ll} \text{minimize} & x^T Q x + c^T x \\ \text{subject to} & x_i \in \{0,1\}, \quad i = 1, \ldots, n. \end{array}$$

where $Q \in \mathcal{S}^n$ can be indefinite.

- Rewrite binary constraints $x_i \in \{0,1\}$:

  $$x_i^2 = x_i \quad \Longleftrightarrow \quad X = xx^T, \quad \mathbf{diag}(X) = x.$$

  Still non-convex, since $\mathbf{rank}(X) = 1$.

- Semidefinite relaxation:

  $$X \succeq xx^T, \quad \mathbf{diag}(X) = x.$$

# Combinatorial relaxations

We consider a binary problem

$$\begin{array}{ll} \text{minimize} & x^T Q x + c^T x \\ \text{subject to} & x_i \in \{0,1\}, \quad i = 1,\ldots,n. \end{array}$$

where $Q \in \mathcal{S}^n$ can be indefinite.

- Rewrite binary constraints $x_i \in \{0,1\}$:

$$x_i^2 = x_i \quad \Longleftrightarrow \quad X = xx^T, \quad \mathbf{diag}(X) = x.$$

  Still non-convex, since $\mathbf{rank}(X) = 1$.

- Semidefinite relaxation:

$$X \succeq xx^T, \quad \mathbf{diag}(X) = x.$$

# Combinatorial relaxations

Lifted non-convex problem:

$$
\begin{array}{ll}
\text{minimize} & \langle Q, X \rangle + c^T x \\
\text{subject to} & \mathbf{diag}(X) = x \\
& X = xx^T.
\end{array}
$$

Semidefinite relaxation:

$$
\begin{array}{ll}
\text{minimize} & \langle Q, X \rangle + c^T x \\
\text{subject to} & \mathbf{diag}(X) = x \\
& \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0.
\end{array}
$$

- Relaxation is exact if $X = xx^T$.
- Otherwise can be strengthened, e.g., by adding $X_{ij} \geq 0$.
- Typical relaxations for combinatorial optimization.

# Combinatorial relaxations

Lifted non-convex problem:

$$\begin{array}{ll} \text{minimize} & \langle Q, X\rangle + c^T x \\ \text{subject to} & \textbf{diag}(X) = x \\ & X = xx^T. \end{array}$$

Semidefinite relaxation:

$$\begin{array}{ll} \text{minimize} & \langle Q, X\rangle + c^T x \\ \text{subject to} & \textbf{diag}(X) = x \\ & \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0. \end{array}$$

- Relaxation is exact if $X = xx^T$.
- Otherwise can be strengthened, e.g., by adding $X_{ij} \geq 0$.
- Typical relaxations for combinatorial optimization.

# Combinatorial relaxations

Lifted non-convex problem:

$$\begin{array}{ll} \text{minimize} & \langle Q, X\rangle + c^T x \\ \text{subject to} & \mathbf{diag}(X) = x \\ & X = xx^T. \end{array}$$

Semidefinite relaxation:

$$\begin{array}{ll} \text{minimize} & \langle Q, X\rangle + c^T x \\ \text{subject to} & \mathbf{diag}(X) = x \\ & \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0. \end{array}$$

- Relaxation is exact if $X = xx^T$.
- Otherwise can be strengthened, e.g., by adding $X_{ij} \geq 0$.
- Typical relaxations for combinatorial optimization.

# Combinatorial relaxations

Lifted non-convex problem:

$$\begin{array}{ll} \text{minimize} & \langle Q, X \rangle + c^T x \\ \text{subject to} & \mathbf{diag}(X) = x \\ & X = xx^T. \end{array}$$

Semidefinite relaxation:

$$\begin{array}{ll} \text{minimize} & \langle Q, X \rangle + c^T x \\ \text{subject to} & \mathbf{diag}(X) = x \\ & \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0. \end{array}$$

- Relaxation is exact if $X = xx^T$.
- Otherwise can be strengthened, e.g., by adding $X_{ij} \geq 0$.
- Typical relaxations for combinatorial optimization.

# Combinatorial relaxations

Lifted non-convex problem:

$$\begin{array}{ll} \text{minimize} & \langle Q, X\rangle + c^T x \\ \text{subject to} & \mathbf{diag}(X) = x \\ & X = xx^T. \end{array}$$

Semidefinite relaxation:

$$\begin{array}{ll} \text{minimize} & \langle Q, X\rangle + c^T x \\ \text{subject to} & \mathbf{diag}(X) = x \\ & \begin{pmatrix} X & x \\ x^T & 1 \end{pmatrix} \succeq 0. \end{array}$$

- Relaxation is exact if $X = xx^T$.
- Otherwise can be strengthened, e.g., by adding $X_{ij} \geq 0$.
- Typical relaxations for combinatorial optimization.

# Relaxations for boolean optimization

Same approach used for boolean constraints $x_i \in \{-1, +1\}$.

## Lifting of boolean constraints

Rewrite boolean constraints $x_i \in \{-1, 1\}$:

$$x_i^2 = 1 \quad \Longleftrightarrow \quad X = xx^T, \quad \mathbf{diag}(X) = e.$$

## Semidefinite relaxation of boolean constraints

$$X \succeq xx^T, \quad \mathbf{diag}(X) = e.$$

# Nearest correlation matrix

Consider

$$S = \{X \in \mathcal{S}_+^n \mid X_{ii} = 1,\ i = 1, \dots, n\}.$$

For a symmetric $A \in \mathbb{R}^{n \times n}$, the *nearest correlation matrix* is

$$X^\star = \arg\min_{X \in S} \|A - X\|_F,$$

which corresponds to a mixed SOCP/SDP,

$$\begin{array}{lll} \text{minimize} & t & \\ \text{subject to} & \|\mathbf{vec}(A - X)\|_2 & \leq t \\ & \mathbf{diag}(X) & = e \\ & X \succeq 0. & \end{array}$$

MOSEK is limited by the many constraints to, say $n < 200$.

# Nearest correlation matrix

Consider

$$S = \{X \in \mathcal{S}_+^n \mid X_{ii} = 1,\ i = 1, \ldots, n\}.$$

For a symmetric $A \in \mathbb{R}^{n\times n}$, the *nearest correlation matrix* is

$$X^\star = \arg\min_{X \in S} \|A - X\|_F,$$

which corresponds to a mixed SOCP/SDP,

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|\mathbf{vec}(A - X)\|_2 \leq t \\ & \mathbf{diag}(X) = e \\ & X \succeq 0. \end{array}$$

MOSEK is limited by the many constraints to, say $n < 200$.

# The nearest correlation problem

## A Python Fusion implementation

```
def svec(e):
    N = e.getShape().dim(0)

    rows = [i for i in range(N) for j in range(i,N)]
    cols = [j for i in range(N) for j in range(i,N)]
    vals = [ 2.0**0.5  if i!=j else 1.0 for i in range(N)  for j in range(i,N)]

    return Expr.flatten(Expr.mulElm(e, Matrix.sparse(N,N,rows,cols,vals)))

def nearestcorr(A):

    n = A.shape[0]
    with Model("NearestCorrelation") as M:

      X = M.variable("X", Domain.inPSDCone(n))
      t = M.variable("t", 1, Domain.unbounded())

      # (t, svec (A-X)) \in Q
      v = svec( Expr.sub(A,X) )
      M.constraint("C1", Expr.vstack(t, v), Domain.inQCone() )

      # diag(X) = e
      M.constraint("C2",X.diag(), Domain.equalsTo(1.0))
      M.objective(ObjectiveSense.Minimize, t)

      M.setLogHandler(sys.stdout)
      M.solve()

      return X.level()
```

# Solving the nearest correlation problem

$n = 200$, $A$ is a tridiagonal matrix with all ones.

```
Optimizer  - solved problem         : the primal
Optimizer  - Constraints            : 20300
Optimizer  - Cones                  : 1
Optimizer  - Scalar variables       : 20101             conic                  : 20101
Optimizer  - Semi-definite variables: 1                 scalarized             : 20100
Factor     - setup time             : 106.40            dense det. time        : 0.00
Factor     - ML order time          : 45.87             GP order time          : 0.00
Factor     - nonzeros before factor : 2.06e+08          after factor           : 2.06e+08
Factor     - dense dim.             : 1                 flops                  : 2.79e+12
ITE PFEAS    DFEAS    GFEAS    PRSTATUS   POBJ              DOBJ              MU       TIME
0   7.1e-01  1.0e+00  1.5e+00  0.00e+00   2.000000000e+00   0.000000000e+00   1.0e+00  106.50
1   5.9e-01  8.3e-01  1.4e+00  1.01e+00   1.248225100e+01   1.082553841e+01   8.3e-01  152.32
2   1.1e-01  1.6e-01  3.9e-01  2.55e+00   1.806503701e+01   1.902214271e+01   1.6e-01  196.79
3   1.9e-02  2.7e-02  1.8e-01  1.09e+00   8.643007378e+00   8.758766674e+00   2.7e-02  239.92
4   3.5e-03  5.0e-03  7.9e-02  1.00e+00   7.505538503e+00   7.524428291e+00   5.0e-03  283.61
5   5.3e-04  7.5e-04  3.3e-02  1.01e+00   7.063986706e+00   7.066275044e+00   7.5e-04  324.20
6   6.5e-05  9.1e-05  1.2e-02  1.00e+00   6.995927741e+00   6.996169972e+00   9.1e-05  364.65
7   8.4e-06  1.2e-05  4.4e-03  1.00e+00   6.990064508e+00   6.990092742e+00   1.2e-05  405.33
8   5.8e-07  8.2e-07  1.2e-03  1.00e+00   6.989354053e+00   6.989355650e+00   8.2e-07  444.98
9   8.4e-08  1.2e-07  4.6e-04  1.00e+00   6.989301381e+00   6.989301596e+00   1.2e-07  485.93
10  1.5e-08  2.1e-08  2.0e-04  1.00e+00   6.989293061e+00   6.989293088e+00   2.1e-08  525.41
```

# Diagonal decomposition

Earlier we required a diagonal decomposition of a covariance, e.g.,

$$\begin{array}{ll} \text{maximize} & e^T d \\ \text{subject to} & Q - \mathbf{Diag}(d) \succeq 0 \\ & d \geq 0. \end{array}$$

with a dual problem

$$\begin{array}{ll} \text{minimize} & \langle Q, X \rangle \\ \text{subject to} & \mathbf{diag}(X) \geq e \\ & X \succeq 0. \end{array}$$

The dual problem is a more efficient characterization for MOSEK.

# Diagonal decomposition of covariance

## A Python Fusion implementation

```
def MaxSum(Q):

  with Model('MaxSum') as M:

      n = len(Q)
      X = M.variable('X', Domain.inPSDCone(n))

      for j in range(n):
          M.constraint(X.index(j,j), Domain.greaterThan(1.0))

      M.objective('obj', ObjectiveSense.Minimize, Expr.dot(Q, X))

      M.setLogHandler(sys.stdout)
      M.solve()

      Z = X.dual()
      return [ Q[i][i] - Z[i*(n+1)] for i in range(n) ]
```

# Markowitz portfolio optimization
## Computing a diagonal decomposition

Data from Frangioni and Gentile ($n = 200$):
`http://www.di.unipi.it/optimize/Data/MV.html`

| ITE | PFEAS | DFEAS | GFEAS | PRSTATUS | POBJ | DOBJ | MU | TIME |
|---|---|---|---|---|---|---|---|---|
| 0 | 1.1e+00 | 5.0e+02 | 7.4e+04 | 0.00e+00 | 4.747960000e+06 | 0.000000000e+00 | 1.0e+00 | 0.04 |
| 1 | 8.2e-02 | 3.6e+01 | 1.5e+03 | -9.90e-01 | 4.462882203e+06 | 1.893344218e+04 | 7.3e-02 | 0.10 |
| 2 | 1.5e-03 | 6.4e-01 | 8.5e+00 | -8.62e-01 | 1.152969863e+06 | 2.416895289e+05 | 1.3e-03 | 0.15 |
| 3 | 2.8e-04 | 1.2e-01 | 3.4e+00 | 6.17e-01 | 6.315008783e+05 | 4.235153740e+05 | 2.5e-04 | 0.20 |
| 4 | 5.3e-05 | 2.3e-02 | 1.5e+00 | 9.13e-01 | 6.043876014e+05 | 5.640115401e+05 | 4.7e-05 | 0.25 |
| 5 | 2.5e-05 | 1.1e-02 | 1.1e+00 | 9.83e-01 | 5.936737697e+05 | 5.740909250e+05 | 2.3e-05 | 0.30 |
| 6 | 1.9e-05 | 8.5e-03 | 1.0e+00 | 9.92e-01 | 5.891086789e+05 | 5.742513338e+05 | 1.7e-05 | 0.36 |
| 7 | 4.8e-06 | 2.1e-03 | 6.0e-01 | 9.94e-01 | 5.809345989e+05 | 5.772023792e+05 | 4.3e-06 | 0.41 |
| 8 | 9.3e-07 | 4.1e-04 | 2.9e-01 | 9.98e-01 | 5.787392737e+05 | 5.780155040e+05 | 8.3e-07 | 0.46 |
| 9 | 1.9e-07 | 8.3e-05 | 1.3e-01 | 1.00e+00 | 5.782823943e+05 | 5.781360063e+05 | 1.7e-07 | 0.51 |
| 10 | 4.0e-08 | 1.8e-05 | 6.3e-02 | 1.00e+00 | 5.781868331e+05 | 5.781559827e+05 | 3.5e-08 | 0.56 |
| 11 | 7.7e-09 | 3.4e-06 | 2.8e-02 | 1.00e+00 | 5.781658349e+05 | 5.781598509e+05 | 6.8e-09 | 0.61 |
| 12 | 1.4e-09 | 6.4e-07 | 1.3e-02 | 1.00e+00 | 5.781616976e+05 | 5.781605773e+05 | 1.3e-09 | 0.66 |
| 13 | 1.5e-10 | 6.5e-08 | 4.1e-03 | 1.00e+00 | 5.781608410e+05 | 5.781607258e+05 | 1.3e-10 | 0.71 |
| 14 | 1.4e-11 | 6.3e-09 | 1.3e-03 | 1.00e+00 | 5.781607522e+05 | 5.781607411e+05 | 1.3e-11 | 0.76 |

Solved in less than a second using MOSEK.

# Computing the tightest perspective relaxation

The dual of the perspective relaxation can be expressed as

$$\begin{array}{ll} \text{maximize} & \nu + \gamma r - \lambda^T e - \alpha \\ \text{subject to} & \begin{pmatrix} Q - \mathbf{Diag}(d) & y \\ y^T & \alpha \end{pmatrix} \succeq 0 \\ & y = (1/2)(2z + \nu e + \gamma\mu + (U+L)\beta) \\ & ((1/2)(d_i + \beta_i), \lambda_i + u_i l_i \beta_i, z_i) \in \mathcal{Q}_r \\ & \gamma \geq 0, \lambda \geq 0, \beta \geq 0. \end{array}$$

- Quadratic constraint written as semidefinite constraint.
- The problem is linear in $d$.
- Maximizing over $d > 0$ gives the tightest perspective relaxation.

Zheng, Sun & Li: `www.optimization-online.org/DB_HTML/2010/11/2797.html`

# Computing the tightest perspective relaxation

The dual of the perspective relaxation can be expressed as

$$
\begin{array}{ll}
\text{maximize} & \nu + \gamma r - \lambda^T e - \alpha \\
\text{subject to} & \begin{pmatrix} Q - \mathbf{Diag}(d) & y \\ y^T & \alpha \end{pmatrix} \succeq 0 \\
& y = (1/2)(2z + \nu e + \gamma\mu + (U + L)\beta) \\
& ((1/2)(d_i + \beta_i), \lambda_i + u_i l_i \beta_i, z_i) \in \mathcal{Q}_r \\
& \gamma \geq 0, \lambda \geq 0, \beta \geq 0.
\end{array}
$$

- Quadratic constraint written as semidefinite constraint.
- The problem is linear in $d$.
- Maximizing over $d > 0$ gives the tightest perspective relaxation.

Zheng, Sun & Li: `www.optimization-online.org/DB_HTML/2010/11/2797.html`

# Computing the tightest perspective relaxation

The dual of the perspective relaxation can be expressed as

$$
\begin{array}{ll}
\text{maximize} & \nu + \gamma r - \lambda^T e - \alpha \\
\text{subject to} & \begin{pmatrix} Q - \mathbf{Diag}(d) & y \\ y^T & \alpha \end{pmatrix} \succeq 0 \\
& y = (1/2)(2z + \nu e + \gamma\mu + (U + L)\beta) \\
& ((1/2)(d_i + \beta_i), \lambda_i + u_i l_i \beta_i, z_i) \in \mathcal{Q}_r \\
& \gamma \geq 0, \lambda \geq 0, \beta \geq 0.
\end{array}
$$

- Quadratic constraint written as semidefinite constraint.
- The problem is linear in $d$.
- Maximizing over $d > 0$ gives the tightest perspective relaxation.

Zheng, Sun & Li: `www.optimization-online.org/DB_HTML/2010/11/2797.html`

# Computing the tightest perspective relaxation

Optimizing over $d > 0$ and dualizing gives us:

$$
\begin{array}{ll}
\text{minimize} & \langle Q, X \rangle \\
\text{subject to} & e^T x = 1 \\
& \mu^T x \geq r \\
& ((1/2)\phi_i, y_i, x_i) \in \mathcal{Q}_r \\
& \phi_i - (u_i + l_i)x_i + u_i l_i y_i \leq 0 \\
& \mathbf{diag}(X) \geq \phi \\
& X \succeq xx^T \\
& y \leq e,
\end{array}
$$

which should be compared to the perspective reformulation with fixed $d$,

$$
\begin{array}{ll}
\text{minimize} & x^T(Q - \mathbf{Diag}(d))x + d^T \phi \\
\text{subject to} & e^T x = 1 \\
& \mu^T x \geq r \\
& ((1/2)\phi_i, y_i, x_i) \in \mathcal{Q}_r \\
& \phi_i - (u_i + l_i)x_i + u_i l_i y_i \leq 0 \\
& y \leq e.
\end{array}
$$

# Computing the tightest perspective relaxation

```
def perspective_tight(Q, mu, r, l, u):
  with Model('MaxSum') as M:
      n = len(mu)
      a   = M.variable('a',   1, Domain.greaterThan(0.0))
      x   = M.variable('x',   n, Domain.greaterThan(0.0))
      phi = M.variable('phi', n, Domain.greaterThan(0.0))
      y   = M.variable('y',   n, Domain.inRange(0.0, 1.0))
      X   = M.variable('X',      Domain.inPSDCone(n+1))

      # (0.5*phi_i, y_i, x_i) \in Qr,  sum(x) == 1,  mu'*x >= r,  phi - (L+U)*x + L*U*y <= 0
      M.constraint(Expr.hstack(Expr.mul(0.5, phi), y, x), Domain.inRotatedQCone())
      M.constraint(Expr.sum(x), Domain.equalsTo(1.0))
      M.constraint(Expr.dot(mu,x), Domain.greaterThan(r))
      M.constraint(Expr.add(Expr.sub(phi, Expr.mulElm(l+u,x)), Expr.mulElm(l*u,y)), Domain.lessThan(0.0))

      # Diag(X) >= phi
      for j in range(n):
          M.constraint(Expr.sub(X.index(j,j), phi.index(j)), Domain.greaterThan(0.0))

      # [X x; x' 1] >= 0
      M.constraint(Expr.sub(X.slice([0,n],[n,n+1]), x), Domain.equalsTo(0.0))
      M.constraint(X.index(n,n), Domain.equalsTo(1.0))

      # minimize dot(Q,X)
      M.objective('obj', ObjectiveSense.Minimize, Expr.dot(Q, X.slice([0,0],[n,n])))
      M.setLogHandler(sys.stdout)
      M.solve()
      xo, Z = x.level(), X.dual()

      return (xo, [ Q[i][i] - Z[i*(n+2)] for i in range(n) ])
```

# Sum-of-squares relaxations

- $f$: multivariate polynomial of degree $2d$.
- $v_d = (1, x_1, x_2, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_n^2, \ldots, x_n^d)$.
  Vector of monomials of degree $d$ or less.

**Sum-of-squares representation**

$f$ is a sum-of-squares (SOS) iff

$$f(x_1, \ldots, x_n) = v_d^T Q v_d, \quad Q \succeq 0.$$

If $Q = LL^T$ then

$$f(x_1, \ldots, x_n) = v_d^T LL^T v_d = \sum_{i=1}^{m} (l_i^T v_d)^2.$$

Is obviously **sufficient** for $f(x_1, \ldots, x_n) \geq 0$.

# Sum-of-squares relaxations

- $f$: multivariate polynomial of degree $2d$.
- $v_d = (1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_n^2, \ldots, x_n^d)$.
  Vector of monomials of degree $d$ or less.

## Sum-of-squares representation

$f$ is a sum-of-squares (SOS) iff

$$f(x_1, \ldots, x_n) = v_d^T Q v_d, \quad Q \succeq 0.$$

If $Q = LL^T$ then

$$f(x_1, \ldots, x_n) = v_d^T LL^T v_d = \sum_{i=1}^{m} (l_i^T v_d)^2.$$

Is obviously **sufficient** for $f(x_1, \ldots, x_n) \geq 0$.

# A simple example

Consider

$$f(x,z) = 2x^4 + 2x^3z - x^2z^2 + 5z^4,$$

homogeneous of degree 4, so we only need

$$v = \begin{pmatrix} x^2 & xz & z^2 \end{pmatrix}.$$

Comparing cofficients of $f(x,z)$ and $v^TQv = \langle Q, vv^T \rangle$,

$$\langle Q, vv^T \rangle = \left\langle \begin{pmatrix} q_{00} & q_{01} & q_{02} \\ q_{10} & q_{11} & q_{12} \\ q_{20} & q_{21} & q_{22} \end{pmatrix}, \begin{pmatrix} x^4 & x^3z & x^2z^2 \\ x^3z & x^2z^2 & xz^3 \\ x^2z^2 & xz^3 & z^4 \end{pmatrix} \right\rangle$$

we see that $f(x,z)$ is SOS iff $Q \succeq 0$ and

$$q_{00} = 2, \quad 2q_{10} = 2, \quad 2q_{20} + q_{11} = -1, \quad 2q_{21} = 0, \quad q_{22} = 5.$$

# A simple example

Consider

$$f(x,z) = 2x^4 + 2x^3 z - x^2 z^2 + 5z^4,$$

homogeneous of degree 4, so we only need

$$v = \begin{pmatrix} x^2 & xz & z^2 \end{pmatrix}.$$

Comparing cofficients of $f(x,z)$ and $v^T Q v = \langle Q, vv^T \rangle$,

$$\langle Q, vv^T \rangle = \langle \begin{pmatrix} q_{00} & q_{01} & q_{02} \\ q_{10} & q_{11} & q_{12} \\ q_{20} & q_{21} & q_{22} \end{pmatrix}, \begin{pmatrix} x^4 & x^3 z & x^2 z^2 \\ x^3 z & x^2 z^2 & xz^3 \\ x^2 z^2 & xz^3 & z^4 \end{pmatrix} \rangle$$

we see that $f(x,z)$ is SOS iff $Q \succeq 0$ and

$$q_{00} = 2, \quad 2q_{10} = 2, \quad 2q_{20} + q_{11} = -1, \quad 2q_{21} = 0, \quad q_{22} = 5.$$

# Applications in polynomial optimization

$$f(x,z) = 4x^2 - \frac{21}{10}x^4 + \frac{1}{3}x^6 + xz - 4z^2 + 4z^4$$

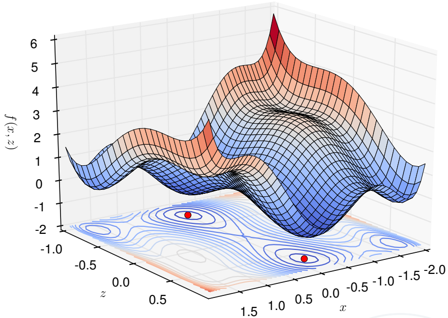
## Global lower bound

Replace non-tractable problem,

$$\text{minimize} \quad f(x,z)$$

by a tractable lower bound

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & f(x,z) - t \text{ is SOS.} \end{array}$$




Relaxation finds the global optimum $t = -1.031$.

Essentially due to Shor, 1987.

$$f(x,z) - t = 4x^2 - \frac{21}{10}x^4 + \frac{1}{3}x^6 + xz - 4z^2 + 4z^4 - t$$

$$vv^T = \begin{pmatrix} 1 & x & z & x^2 & xz & z^2 & x^3 & x^2z & xz^2 & z^3 \\ x & x^2 & xz & x^3 & x^2z & xz^2 & x^4 & x^3z & x^2z^2 & xz^3 \\ z & xz & z^2 & x^2z & xz^2 & z^3 & x^3z & x^2z^2 & xz^3 & z^4 \\ x^2 & x^3 & x^2z & x^4 & x^3z & x^2z^2 & x^5 & x^4z & x^3z^2 & x^2z^3 \\ xz & x^2z & xz^2 & x^3z & x^2z^2 & xz^3 & x^4z & x^3z^2 & x^2z^3 & xz^4 \\ z^2 & xz^2 & z^3 & x^2z^2 & xz^3 & z^4 & x^3z^2 & x^2z^3 & xz^4 & y^5 \\ x^3 & x^4 & x^3z & x^5 & x^4z & x^3z^2 & x^6 & x^5z & x^4z^2 & x^3z^3 \\ x^2z & x^3z & x^2z^2 & x^4z & x^3z^2 & x^2z^3 & x^5z & x^4z^2 & x^3z^3 & x^2z^4 \\ xz^2 & x^2z^2 & xz^3 & x^3z^2 & x^2z^3 & xz^4 & x^4z^2 & x^3z^3 & x^2z^4 & xz^5 \\ z^3 & xz^3 & z^4 & x^2z^3 & xz^4 & z^5 & x^3z^3 & x^2z^4 & xz^5 & z^6 \end{pmatrix}$$

By comparing cofficients of $v^TQv$ and $f(x,z) - t$:

$$q_{00} = -t, \quad (2q_{30} + q_{11}) = 4, \quad (2q_{72} + q_{44}) = -\frac{21}{10}, \quad q_{77} = \frac{1}{3}$$

$$2(q_{51} + q_{32}) = 1, \quad (2q_{61} + q_{33}) = -4, \quad (2q_{10,3} + q_{66}) = 4$$

$$2q_{10} = 0, \quad 2q_{20} = 0, \quad 2(q_{71} + q_{42}) = 0, \quad \dots$$

A standard SDP with a $10 \times 10$ variable and 28 constraints.

# Nonnegative polynomials

- Univariate polynomial of degree $2n$:

$$f(x) = c_0 + c_1 x + \cdots + c_{2n} x^{2n}.$$

- Nonnegativity is **equivalent** to SOS, i.e.,

$$f(x) \geq 0 \qquad \Longleftrightarrow \qquad f(x) = v^T Q v, \quad Q \succeq 0$$

with $v = (1, x, \ldots, x^n)$.

- Simple extensions for nonnegativity on a subinterval $I \subset \mathbb{R}$.

# Nonnegative polynomials

- Univariate polynomial of degree $2n$:

$$f(x) = c_0 + c_1 x + \cdots + c_{2n} x^{2n}.$$

- Nonnegativity is **equivalent** to SOS, i.e.,

$$f(x) \geq 0 \qquad \Longleftrightarrow \qquad f(x) = v^T Q v, \quad Q \succeq 0$$

  with $v = (1, x, \ldots, x^n)$.

- Simple extensions for nonnegativity on a subinterval $I \subset \mathbb{R}$.

# Nonnegative polynomials

- Univariate polynomial of degree $2n$:

$$f(x) = c_0 + c_1 x + \cdots + c_{2n} x^{2n}.$$

- Nonnegativity is **equivalent** to SOS, i.e.,

$$f(x) \geq 0 \qquad \Longleftrightarrow \qquad f(x) = v^T Q v, \quad Q \succeq 0$$

  with $v = (1, x, \ldots, x^n)$.
- Simple extensions for nonnegativity on a subinterval $I \subset \mathbb{R}$.

# Polynomial interpolation

Fit a polynomial of degree $n$ to a set of points $(x_j, y_j)$,

$$f(x_j) = y_j, \quad j = 1, \ldots, m,$$

i.e., linear equality constraints in $c$,

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \ldots & x_m^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

Semidefinite shape constraints:

- Nonnegativity $f(x) \geq 0$.
- Monotonicity $f'(x) \geq 0$.
- Convexity $f''(x) \geq 0$.

# Polynomial interpolation

Fit a polynomial of degree $n$ to a set of points $(x_j, y_j)$,

$$f(x_j) = y_j, \quad j = 1, \ldots, m,$$

i.e., linear equality constraints in $c$,

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^n \\ 1 & x_2 & x_2^2 & \ldots & x_2^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_m & x_m^2 & \ldots & x_m^n \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

Semidefinite shape constraints:

- Nonnegativity $f(x) \geq 0$.
- Monotonicity $f'(x) \geq 0$.
- Convexity $f''(x) \geq 0$.

# Polynomial interpolation

## A specific example

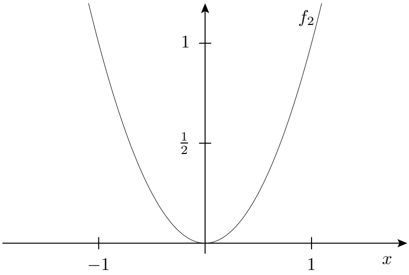
### Smooth interpolation

Minimize largest derivative,

$$\begin{array}{ll} \text{minimize} & \displaystyle\max_{x\in[-1,1]} |f'(x)| \\ \text{subject to} & f(-1) = 1 \\ & f(0) = 0 \\ & f(1) = 1 \end{array}$$

or equivalently

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & -z \leq f'(x) \leq z \\ & f(-1) = 1 \\ & f(0) = 0 \\ & f(1) = 1. \end{array}$$




$f_2(x) = x^2 \qquad f_2'(1) = 2$

# Polynomial interpolation

## A specific example

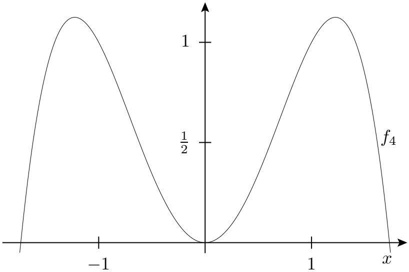
### Smooth interpolation

Minimize largest derivative,

$$\begin{array}{ll} \text{minimize} & \max_{x\in[-1,1]} |f'(x)| \\ \text{subject to} & f(-1) = 1 \\ & f(0) = 0 \\ & f(1) = 1 \end{array}$$

or equivalently

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & -z \leq f'(x) \leq z \\ & f(-1) = 1 \\ & f(0) = 0 \\ & f(1) = 1. \end{array}$$




$$f_2(x) = x^2 \qquad f_2'(1) = 2$$

$$f_4(x) = \frac{3}{2}x^2 - \frac{1}{2}x^4 \qquad f_4'(\frac{1}{\sqrt{2}}) = \sqrt{2}$$

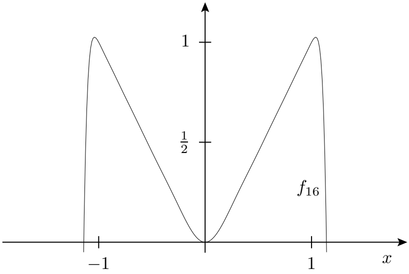
### Smooth interpolation

Minimize largest derivative,

$$\begin{array}{ll} \text{minimize} & \max_{x\in[-1,1]} |f'(x)| \\ \text{subject to} & f(-1) = 1 \\ & f(0) = 0 \\ & f(1) = 1 \end{array}$$

or equivalently

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & -z \le f'(x) \le z \\ & f(-1) = 1 \\ & f(0) = 0 \\ & f(1) = 1. \end{array}$$




$$f_2(x) = x^2 \qquad f_2'(1) = 2$$

$$f_4(x) = \frac{3}{2}x^2 - \frac{1}{2}x^4 \qquad f_4'\left(\frac{1}{\sqrt{2}}\right) = \sqrt{2}$$

# Polynomial interpolation

## A specific example

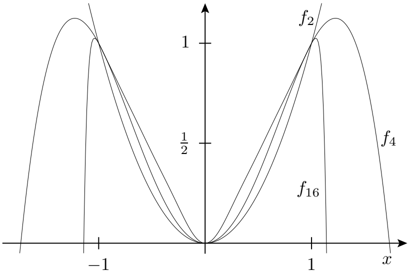
### Smooth interpolation

Minimize largest derivative,

$$\begin{array}{ll} \text{minimize} & \max_{x\in[-1,1]} |f'(x)| \\ \text{subject to} & f(-1) = 1 \\ & f(0) = 0 \\ & f(1) = 1 \end{array}$$

or equivalently

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & -z \le f'(x) \le z \\ & f(-1) = 1 \\ & f(0) = 0 \\ & f(1) = 1. \end{array}$$




$$f_2(x) = x^2 \qquad f_2'(1) = 2$$

$$f_4(x) = \frac{3}{2}x^2 - \frac{1}{2}x^4 \qquad f_4'\left(\frac{1}{\sqrt{2}}\right) = \sqrt{2}$$

More generally, we can form relaxations for polynomial problems

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \geq 0, \quad i = 1, \dots, m \\ & x \in \mathbb{R}^n \end{array}$$

with real polynomials $f, g_i : \mathbb{R}^n \mapsto \mathbb{R}$.

Modeling packages for Matlab:

- **GloptiPoly**, standard moment relaxations.
- **SparsePoP**, sparse moment relaxations.
- **SOSTools**, general sum-of-squares problems.
- **Yalmip**, general sums-of-squares and polynomial optimization.

Rudimentary Julia package by MOSEK:
`https://github.com/MOSEK/Polyopt.jl`

**Thank you!**

Joachim Dahl

www.mosek.com