

Inhalt

Innere-Punkte-Verfahren und lineare Optimierung über Kegeln

Innere-Punkte-Verfahren

Lineare Optimierung über Kegeln

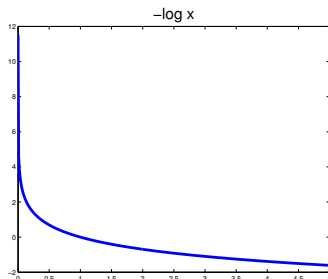
Second-Order-Cone Programme

Semidefinite Optimierung

5.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der **Barriere-Verfahren**, um durch das Innere des Polyeders zu gehen: Ein mit einem **Barriere-Parameter** $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion verhindert das Verlassen des Inneren. Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).



5.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der **Barriere-Verfahren**, um durch das Innere des Polyeders zu gehen: Ein mit einem **Barriere-Parameter** $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion verhindert das Verlassen des Inneren. Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

Bsp: $\min x$ s.t. $x \in [a = 0.5, b = 3.5]$

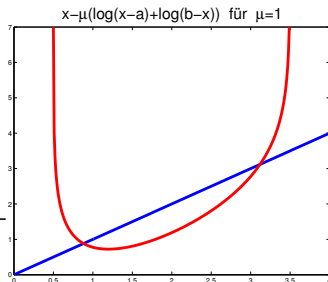
Barriere für $x \geq a$: $-\log(x - a)$

Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ , suche Minimum, etc.



5.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der **Barriere-Verfahren**, um durch das Innere des Polyeders zu gehen: Ein mit einem **Barriere-Parameter** $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion verhindert das Verlassen des Inneren. Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

Bsp: $\min x$ s.t. $x \in [a = 0.5, b = 3.5]$

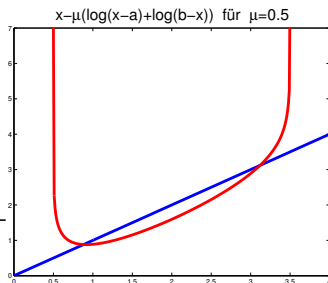
Barriere für $x \geq a$: $-\log(x - a)$

Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ , suche Minimum, etc.



5.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der **Barriere-Verfahren**, um durch das Innere des Polyeders zu gehen: Ein mit einem **Barriere-Parameter** $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion verhindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

Bsp: $\min x$ s.t. $x \in [a = 0.5, b = 3.5]$

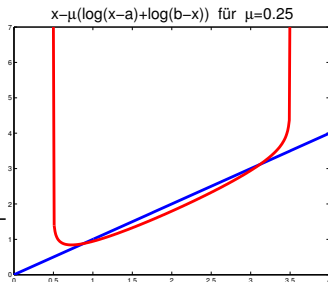
Barriere für $x \geq a$: $-\log(x - a)$

Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ , suche Minimum, etc.



5.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der **Barriere-Verfahren**, um durch das Innere des Polyeders zu gehen: Ein mit einem **Barriere-Parameter** $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion verhindert das Verlassen des Inneren. Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

Bsp: $\min x$ s.t. $x \in [a = 0.5, b = 3.5]$

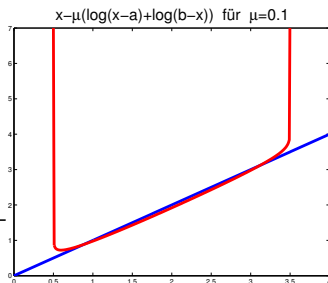
Barriere für $x \geq a$: $-\log(x - a)$

Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ , suche Minimum, etc.



5.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der **Barriere-Verfahren**, um durch das Innere des Polyeders zu gehen: Ein mit einem **Barriere-Parameter** $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion verhindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

Bsp: $\min x$ s.t. $x \in [a = 0.5, b = 3.5]$

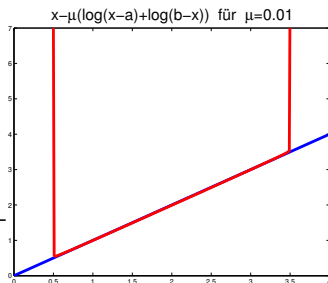
Barriere für $x \geq a$: $-\log(x - a)$

Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ , suche Minimum, etc.



5.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex worst case: exponentiell]

Nutzt Idee der **Barriere-Verfahren**, um durch das Innere des Polyeders zu gehen: Ein mit einem **Barriere-Parameter** $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion verhindert das Verlassen des Inneren. Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

Bsp: $\min x$ s.t. $x \in [a = 0.5, b = 3.5]$

Barriere für $x \geq a$: $-\log(x - a)$

Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

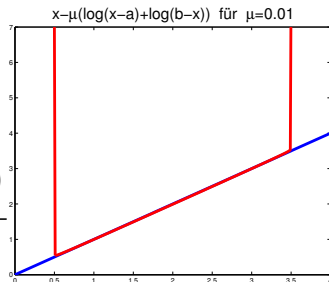
$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ , suche Minimum, etc.

Primales/duales Problem:

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array}$$

$$(D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & z \geq 0 \end{array}$$



[für $x \in \mathbb{R}^n$: $\log x = \sum_{i=1}^n \log x_i$]

5.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der **Barriere-Verfahren**, um durch das Innere des Polyeders zu gehen: Ein mit einem **Barriere-Parameter** $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion verhindert das Verlassen des Inneren. Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

Bsp: $\min x$ s.t. $x \in [a = 0.5, b = 3.5]$

Barriere für $x \geq a$: $-\log(x - a)$

Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

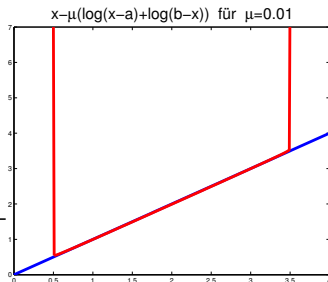
Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ , suche Minimum, etc.

Primales/duales Barriere-Problem:

$$(P_\mu) \quad \begin{array}{ll} \min & c^T x - \mu \log x \\ \text{s.t.} & Ax = b \\ & [x > 0] \end{array}$$

[für $x \in \mathbb{R}^n$: $\log x = \sum_{i=1}^n \log x_i$]

$$(D_\mu) \quad \begin{array}{ll} \max & b^T y + \mu \log z \\ \text{s.t.} & A^T y + z = c \\ & [z > 0] \end{array}$$



Das Primal-Duale KKT-System

Der Name KKT bezieht sich auf die Optimalitätsbedingungen für nichtlineare restringierte Optimierung (näheres dazu später).

$$\begin{array}{ll}
 \min & c^T x - \mu \log x \\
 (P_\mu) \quad \text{s.t.} & Ax = b \\
 & [x > 0]
 \end{array}
 \qquad
 \begin{array}{ll}
 \max & b^T y + \mu \log z \\
 (D_\mu) \quad \text{s.t.} & A^T y + z = c \\
 & [z > 0]
 \end{array}$$

Für die Lagrange-Funktion zum primalen Barriere-Problem (P_μ)

$$L_\mu(x, y) := c^T x + y^T (b - Ax) - \mu \log x$$

sucht man die Nullstellen der Ableitung nach x und y

$$\begin{array}{ll}
 \nabla_x L_\mu = 0 & \rightarrow \quad c - A^T y - \mu x^{-1} = 0 \\
 \nabla_y L_\mu = 0 & \rightarrow \quad b - Ax = 0
 \end{array}
 \qquad
 [x^{-1} := [x_1^{-1}, \dots, x_n^{-1}]^T]$$

Das Primal-Duale KKT-System

Der Name KKT bezieht sich auf die Optimalitätsbedingungen für nichtlineare restringierte Optimierung (näheres dazu später).

$$\begin{array}{ll}
 \min & c^T x - \mu \log x \\
 (P_\mu) \quad \text{s.t.} & Ax = b \\
 & [x > 0]
 \end{array}
 \qquad
 \begin{array}{ll}
 \max & b^T y + \mu \log z \\
 (D_\mu) \quad \text{s.t.} & A^T y + z = c \\
 & [z > 0]
 \end{array}$$

Für die Lagrange-Funktion zum primalen Barriere-Problem (P_μ)

$$L_\mu(x, y) := c^T x + y^T (b - Ax) - \mu \log x$$

sucht man die Nullstellen der Ableitung nach x und y

$$\nabla_x L_\mu = 0 \rightarrow c - A^T y - \mu x^{-1} = 0 \quad [x^{-1} := [x_1^{-1}, \dots, x_n^{-1}]^T]$$

$$\nabla_y L_\mu = 0 \rightarrow b - Ax = 0$$

Setzt man $z = \mu x^{-1}$ erhält man das **Primal-Duale KKT-System**

$$(PD_\mu) \quad \begin{array}{l}
 A^T y + z = c \\
 Ax = b \\
 x \circ z = \mu \mathbf{1}
 \end{array}
 \begin{array}{ll}
 \text{duale Zulässigkeit} & [z > 0] \\
 \text{primale Zulässigkeit} & [x > 0] \\
 \text{perturbierte Komplementarität} & [x \circ z := [x_i z_i]]
 \end{array}$$

Für $\mu = 0$: $x \circ z = 0 \Leftrightarrow x^T z = 0$, mit $x, z \in \mathbb{R}_+^n$ Opt.-Bed. für (P) und (D)!

Das Primal-Duale KKT-System

Der Name KKT bezieht sich auf die Optimalitätsbedingungen für nichtlineare restringierte Optimierung (näheres dazu später).

$$\begin{array}{ll}
 \min & c^T x - \mu \log x \\
 (P_\mu) \quad \text{s.t.} & Ax = b \\
 & [x > 0]
 \end{array}
 \qquad
 \begin{array}{ll}
 \max & b^T y + \mu \log z \\
 (D_\mu) \quad \text{s.t.} & A^T y + z = c \\
 & [z > 0]
 \end{array}$$

Für die Lagrange-Funktion zum primalen Barriere-Problem (P_μ)

$$L_\mu(x, y) := c^T x + y^T (b - Ax) - \mu \log x$$

sucht man die Nullstellen der Ableitung nach x und y

$$\nabla_x L_\mu = 0 \rightarrow c - A^T y - \mu x^{-1} = 0 \quad [x^{-1} := [x_1^{-1}, \dots, x_n^{-1}]^T]$$

$$\nabla_y L_\mu = 0 \rightarrow b - Ax = 0$$

Setzt man $z = \mu x^{-1}$ erhält man das **Primal-Duale KKT-System**

(PD_μ)	$A^T y + z = c$	duale Zulässigkeit	$[z > 0]$
	$Ax = b$	primale Zulässigkeit	$[x > 0]$
	$x \circ z = \mu \mathbf{1}$	perturbierte Komplementarität	$[x \circ z := [x_i z_i]]$

Für $\mu = 0$: $x \circ z = 0 \Leftrightarrow x^T z = 0$, mit $x, z \in \mathbb{R}_+^n$ Opt.-Bed. für (P) und (D)!

Sind (P) und (D) streng zulässig (und hat A vollen Zeilenrang), gibt es für jedes $\mu > 0$ eine eindeutige Lösung $(x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n$, es gilt x_μ ist OL von (P_μ) und (y_μ, z_μ) ist OL von (D_μ) .

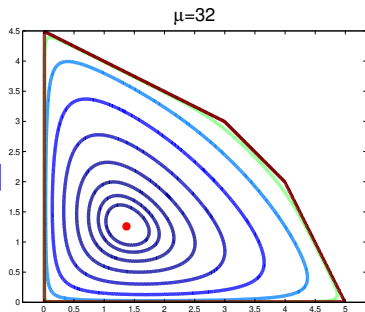
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{rcl} A^T y + z & = & c \\ Ax & = & b \\ x \circ z & = & \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\begin{aligned} \mathcal{Z}(\mu) &= (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) &\in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu). \end{aligned}$$



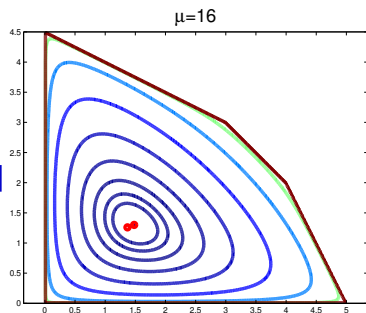
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{rcl} A^T y + z & = & c \\ Ax & = & b \\ x \circ z & = & \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\begin{aligned} \mathcal{Z}(\mu) &= (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) &\in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu). \end{aligned}$$



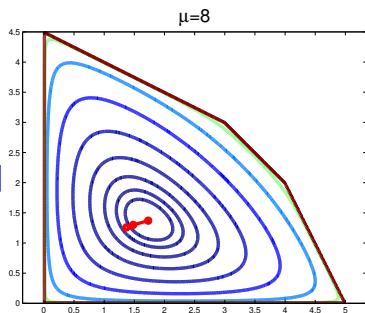
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{l} A^T y + z = c \\ Ax = b \\ x \circ z = \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\begin{aligned} \mathcal{Z}(\mu) &= (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) &\in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu). \end{aligned}$$



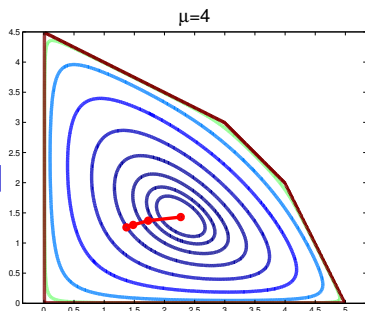
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{l} A^T y + z = c \\ Ax = b \\ x \circ z = \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu).$$



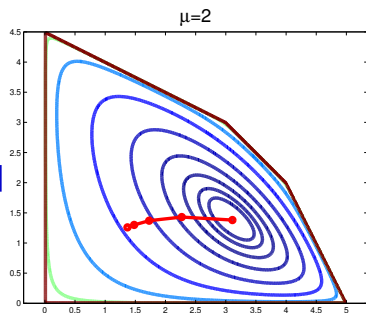
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{rcl} A^T y + z & = & c \\ Ax & = & b \\ x \circ z & = & \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\begin{aligned} \mathcal{Z}(\mu) &= (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) &\in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu). \end{aligned}$$



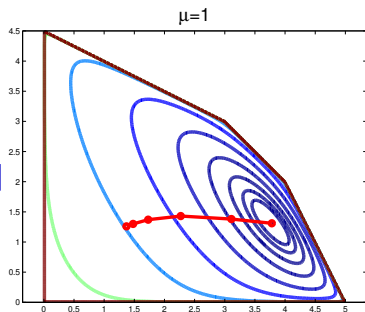
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{l} A^T y + z = c \\ Ax = b \\ x \circ z = \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu).$$



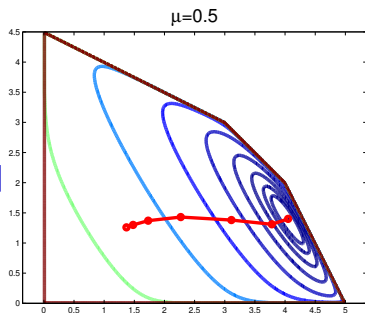
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{l} A^T y + z = c \\ Ax = b \\ x \circ z = \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu).$$



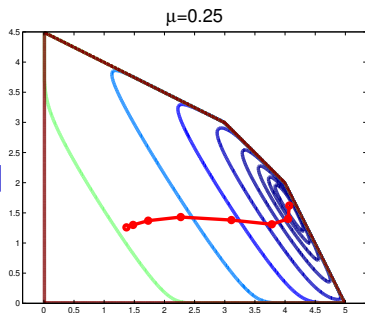
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{rcl} A^T y + z & = & c \\ Ax & = & b \\ x \circ z & = & \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu).$$



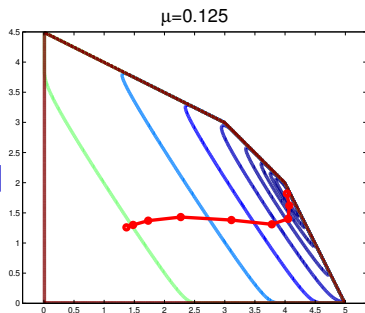
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{l} A^T y + z = c \\ Ax = b \\ x \circ z = \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu).$$



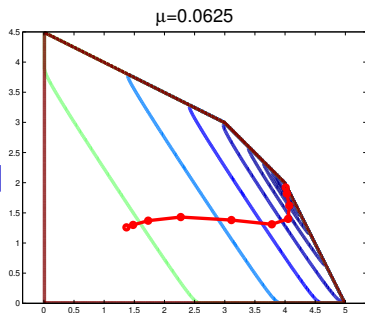
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{l} A^T y + z = c \\ Ax = b \\ x \circ z = \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu).$$



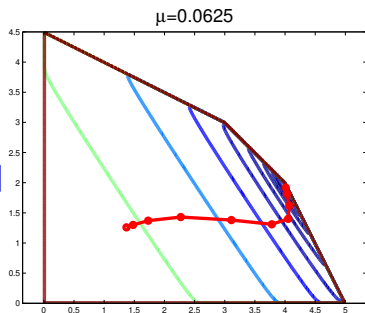
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{l} A^T y + z = c \\ Ax = b \\ x \circ z = \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\begin{aligned} \mathcal{Z}(\mu) &= (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) &\in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu). \end{aligned}$$



Er liegt innerhalb des primalen/dualen Polyeders und konvergiert für $\mu \rightarrow 0$ gegen das **analytische Zentrum** der Opt.-Lösungen

$$(x^*, y^*, z^*) := \lim_{\mu \rightarrow 0} (x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n.$$

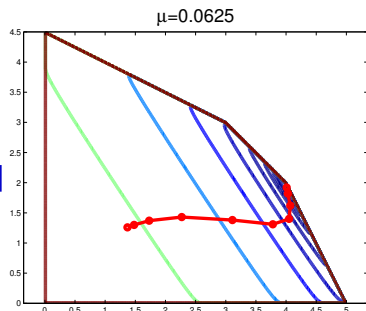
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{cases} A^T y + z = c & [z > 0] \\ Ax = b & [x > 0] \\ x \circ z = \mu \mathbf{1} \end{cases}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\begin{aligned} \mathcal{Z}(\mu) &= (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) &\in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu). \end{aligned}$$



Er liegt innerhalb des primalen/dualen Polyeders und konvergiert für $\mu \rightarrow 0$ gegen das **analytische Zentrum** der Opt.-Lösungen

$$(x^*, y^*, z^*) := \lim_{\mu \rightarrow 0} (x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n.$$

x^* und (y^*, z^*) sind OL von (P) und (D) und **streng komplementär**, d.h., sie erfüllen $x_i z_i = 0$ mit entweder $x_i > 0$ oder $z_i > 0$.

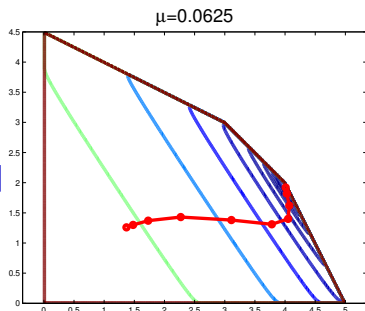
Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu) \quad \begin{array}{l} A^T y + z = c \\ Ax = b \\ x \circ z = \mu \mathbf{1} \end{array} \quad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$[(P), (D) \text{ str. zul.}, \text{Rang}(A) = m]$

Der **zentrale Pfad** ist die glatte Kurve

$$\begin{aligned} \mathcal{Z}(\mu) &= (x_\mu, y_\mu, z_\mu), \quad \mu > 0, \\ (x_\mu, y_\mu, z_\mu) &\in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n \text{ löst } (PD_\mu). \end{aligned}$$



Er liegt innerhalb des primalen/dualen Polyeders und konvergiert für $\mu \rightarrow 0$ gegen das **analytische Zentrum** der Opt.-Lösungen

$$(x^*, y^*, z^*) := \lim_{\mu \rightarrow 0} (x_\mu, y_\mu, z_\mu) \in \mathbb{R}_+^n \times \mathbb{R}^m \times \mathbb{R}_+^n.$$

x^* und (y^*, z^*) sind OL von (P) und (D) und **streng komplementär**, d.h., sie erfüllen $x_i z_i = 0$ mit entweder $x_i > 0$ oder $z_i > 0$.

Nur falls die Optimallösung von (P) bzw. (D) eindeutig ist, ist x^* bzw. (y^*, z^*) auch eine Ecke des zulässigen Polyeders von (P) bzw. (D)!

Lösung des Primal-Dualen Systems

$\mathcal{Z}(\mu)$ nähert man iterativ mit dem Newton-Verfahren für nichtlineare Gleichungen an (s. später): Bestimme für gegebenes $\bar{x} > 0$, \bar{y} , $\bar{z} > 0$ den nächsten Punkt $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ aus der Linearisierung:

$$\begin{array}{rcl}
 A^T y + z = c & & \text{I.} \quad A^T(\bar{y} + \Delta y) + \bar{z} + \Delta z = c \\
 Ax = b & \longrightarrow & \text{II.} \quad A(\bar{x} + \Delta x) = b \\
 x \circ z = \mu \mathbf{1} & & \text{III.} \quad \bar{x} \circ \bar{z} + \Delta x \circ \bar{z} + \bar{x} \circ \Delta z = \mu \mathbf{1}
 \end{array}$$

Lösung des Primal-Dualen Systems

$\mathcal{Z}(\mu)$ nähert man iterativ mit dem Newton-Verfahren für nichtlineare Gleichungen an (s. später): Bestimme für gegebenes $\bar{x} > 0$, \bar{y} , $\bar{z} > 0$ den nächsten Punkt $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ aus der Linearisierung:

$$\begin{array}{rcl}
 A^T y + z & = & c \\
 Ax & = & b \\
 x \circ z & = & \mu \mathbf{1}
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{rcl}
 \text{I.} & A^T(\bar{y} + \Delta y) + \bar{z} + \Delta z & = c \\
 \text{II.} & A(\bar{x} + \Delta x) & = b \\
 \text{III.} & \bar{x} \circ \bar{z} + \Delta x \circ \bar{z} + \bar{x} \circ \Delta z & = \mu \mathbf{1}
 \end{array}$$

- aus I.: $\Delta z = c - A^T \bar{y} - \bar{z} - A^T \Delta y$
- aus III.: $\Delta x = \mu \bar{z}^{-1} - \bar{x} - \bar{x} \circ \bar{z}^{-1} \circ \Delta z \stackrel{(1.)}{=} \bar{x} \circ \bar{z}^{-1} \circ A^T \Delta y + \dots$
- in II.: $A \text{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \Delta y = b - A \bar{x} + \dots$

Lösung des Primal-Dualen Systems

$\mathcal{Z}(\mu)$ nähert man iterativ mit dem Newton-Verfahren für nichtlineare Gleichungen an (s. später): Bestimme für gegebenes $\bar{x} > 0$, \bar{y} , $\bar{z} > 0$ den nächsten Punkt $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ aus der Linearisierung:

$$\begin{array}{rcl}
 A^T y + z = c & & \text{I.} \quad A^T(\bar{y} + \Delta y) + \bar{z} + \Delta z = c \\
 Ax = b & \longrightarrow & \text{II.} \quad A(\bar{x} + \Delta x) = b \\
 x \circ z = \mu \mathbf{1} & & \text{III.} \quad \bar{x} \circ \bar{z} + \Delta x \circ \bar{z} + \bar{x} \circ \Delta z = \mu \mathbf{1}
 \end{array}$$

1. aus I.: $\Delta z = c - A^T \bar{y} - \bar{z} - A^T \Delta y$

2. aus III.: $\Delta x = \mu \bar{z}^{-1} - \bar{x} - \bar{x} \circ \bar{z}^{-1} \circ \Delta z \stackrel{(1.)}{=} \bar{x} \circ \bar{z}^{-1} \circ A^T \Delta y + \dots$

3. in II.: $A \text{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \Delta y = b - A \bar{x} + \dots$

Wegen $x > 0, z > 0$ ist die Matrix $M := A \text{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \in \mathbb{R}^{m \times m}$ symmetrisch positiv semidefinit und positiv definit, falls $\text{Rang}(A) = m$.

\Rightarrow Das Gleichungssystem $M \Delta y = \dots$ kann in etwa $m^3/3$ arithmetischen Operationen mit dem **Cholesky-Verfahren** gelöst werden.

Dieses „faktoriert“ $M = LL^T$ mit L untere Dreiecksmatrix.

Lösung des Primal-Dualen Systems

$\mathcal{Z}(\mu)$ nähert man iterativ mit dem Newton-Verfahren für nichtlineare Gleichungen an (s. später): Bestimme für gegebenes $\bar{x} > 0$, \bar{y} , $\bar{z} > 0$ den nächsten Punkt $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ aus der Linearisierung:

$$\begin{array}{lcl} A^T y + z = c & & \text{I.} \quad A^T(\bar{y} + \Delta y) + \bar{z} + \Delta z = c \\ Ax = b & \longrightarrow & \text{II.} \quad A(\bar{x} + \Delta x) = b \\ x \circ z = \mu \mathbf{1} & & \text{III.} \quad \bar{x} \circ \bar{z} + \Delta x \circ \bar{z} + \bar{x} \circ \Delta z = \mu \mathbf{1} \end{array}$$

1. aus I.: $\Delta z = c - A^T \bar{y} - \bar{z} - A^T \Delta y$

2. aus III.: $\Delta x = \mu \bar{z}^{-1} - \bar{x} - \bar{x} \circ \bar{z}^{-1} \circ \Delta z \stackrel{(1.)}{=} \bar{x} \circ \bar{z}^{-1} \circ A^T \Delta y + \dots$

3. in II.: $A \text{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \Delta y = b - A \bar{x} + \dots$

Wegen $x > 0, z > 0$ ist die Matrix $M := A \text{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \in \mathbb{R}^{m \times m}$ symmetrisch positiv semidefinit und positiv definit, falls $\text{Rang}(A) = m$.

\Rightarrow Das Gleichungssystem $M \Delta y = \dots$ kann in etwa $m^3/3$ arithmetischen Operationen mit dem **Cholesky-Verfahren** gelöst werden.

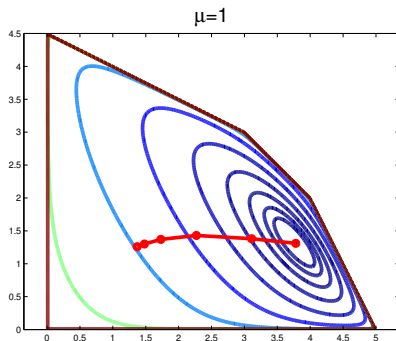
Dieses „faktoriert“ $M = LL^T$ mit L untere Dreiecksmatrix.

Wichtiger Unterschied zum Simplex für die Praxis:

A ist meist nur sehr dünn besetzt, für Simplex in $A_B x_B = b$ gut nutzbar!
Trotz vieler Tricks ist L meist sehr viel dichter besetzt als A

\rightarrow braucht viel mehr Speicher, jede Iteration sehr langsam, dafür wenige!

Grobe Näherung für $\mathcal{Z}(\mu)$ genügt



Für $\bar{\mu} > \mu > 0$ mit $\bar{\mu} - \mu$ klein, ist $\|\mathcal{Z}(\bar{\mu}) - \mathcal{Z}(\mu)\|$ klein. Ist $(\bar{x}, \bar{y}, \bar{z})$ gute Näherung für $\mathcal{Z}(\bar{\mu})$ mit $\bar{\mu} = \bar{x}^T \bar{z} / n$ und $\mu = \sigma \bar{\mu}$ mit $\sigma < 1$ nahe bei 1, dann genügt ein Newton-Schritt, damit $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ eine gute Näherung für $\mathcal{Z}(\sigma \bar{\mu})$ ist mit $\sigma \bar{\mu} = (\bar{x} + \Delta x)^T (\bar{z} + \Delta z) / n$.

Algorithmisches Schema für Innere-Punkte-Verfahren

Idee: Folge dem zentralen Pfad \mathcal{Z} mit Newton-Schritten für $\mu \rightarrow 0$.

Input: $A, b, c, \bar{x} > 0, \bar{y}, \bar{z} > 0, \varepsilon > 0, \sigma \in (0, 1)$

1. Setze $\mu \leftarrow \sigma \frac{\bar{x}^T \bar{z}}{n}$.
2. Bestimme $(\Delta x, \Delta y, \Delta z)$ als Newton-Schritt für (PD_μ) .
3. Bestimme großes $\alpha \in (0, 1]$ mit $\bar{x} + \alpha \Delta x > 0, \bar{z} + \alpha \Delta z > 0$
4. setze $(\bar{x}, \bar{y}, \bar{z}) \leftarrow (\bar{x}, \bar{y}, \bar{z}) + \alpha(\Delta x, \Delta y, \Delta z)$
5. Ist $\|b - A\bar{x}\| < \varepsilon$ und $\|c - A^T \bar{y} - \bar{z}\| < \varepsilon$ und $\bar{x}^T \bar{z} < \varepsilon$ STOP, sonst gehe zu 1.

Algorithmisches Schema für Innere-Punkte-Verfahren

Idee: Folge dem zentralen Pfad \mathcal{Z} mit Newton-Schritten für $\mu \rightarrow 0$.

Input: $A, b, c, \bar{x} > 0, \bar{y}, \bar{z} > 0, \varepsilon > 0, \sigma \in (0, 1)$

1. Setze $\mu \leftarrow \sigma \frac{\bar{x}^T \bar{z}}{n}$.
2. Bestimme $(\Delta x, \Delta y, \Delta z)$ als Newton-Schritt für (PD_μ) .
3. Bestimme großes $\alpha \in (0, 1]$ mit $\bar{x} + \alpha \Delta x > 0, \bar{z} + \alpha \Delta z > 0$
4. setze $(\bar{x}, \bar{y}, \bar{z}) \leftarrow (\bar{x}, \bar{y}, \bar{z}) + \alpha(\Delta x, \Delta y, \Delta z)$
5. Ist $\|b - A\bar{x}\| < \varepsilon$ und $\|c - A^T \bar{y} - \bar{z}\| < \varepsilon$ und $\bar{x}^T \bar{z} < \varepsilon$ STOP, sonst gehe zu 1.

Sind (P) und (D) zulässig und $\text{Rang}(A) = m$, kann man für ein leicht umformuliertes Problem einen zulässigen Startpunkt $\bar{x}, \bar{y}, \bar{z}$ und ein $\sigma < 1$ angeben, sodass der Algorithmus nach höchstens $\gamma \sqrt{n \log \frac{\bar{x}^T \bar{z}}{\varepsilon}}$ Iterationen (für eine problemunabhängige Konstante $\gamma > 0$) endet.

Algorithmisches Schema für Innere-Punkte-Verfahren

Idee: Folge dem zentralen Pfad \mathcal{Z} mit Newton-Schritten für $\mu \rightarrow 0$.

Input: $A, b, c, \bar{x} > 0, \bar{y}, \bar{z} > 0, \varepsilon > 0, \sigma \in (0, 1)$

1. Setze $\mu \leftarrow \sigma \frac{\bar{x}^T \bar{z}}{n}$.
2. Bestimme $(\Delta x, \Delta y, \Delta z)$ als Newton-Schritt für (PD_μ) .
3. Bestimme großes $\alpha \in (0, 1]$ mit $\bar{x} + \alpha \Delta x > 0, \bar{z} + \alpha \Delta z > 0$
4. setze $(\bar{x}, \bar{y}, \bar{z}) \leftarrow (\bar{x}, \bar{y}, \bar{z}) + \alpha(\Delta x, \Delta y, \Delta z)$
5. Ist $\|b - A\bar{x}\| < \varepsilon$ und $\|c - A^T \bar{y} - \bar{z}\| < \varepsilon$ und $\bar{x}^T \bar{z} < \varepsilon$ STOP, sonst gehe zu 1.

Sind (P) und (D) zulässig und $\text{Rang}(A) = m$, kann man für ein leicht umformuliertes Problem einen zulässigen Startpunkt $\bar{x}, \bar{y}, \bar{z}$ und ein $\sigma < 1$ angeben, sodass der Algorithmus nach höchstens $\gamma \sqrt{n \log \frac{\bar{x}^T \bar{z}}{\varepsilon}}$ Iterationen (für eine problemunabhängige Konstante $\gamma > 0$) endet.

- Das Ergebnis ist eine Näherung und keine exakte Lösung!
- Für ε klein genug kann für diese in einem „**cross-over**“-Schritt eine optimale Basis bestimmt werden (benötigt oft viele Simplex-Iterationen).
- Erkennen von Unzulässigkeit/Unbeschränktheit ist deutlich schwerer.

Zusammenfassung der Eigenschaften

- Innere-Punkte-Verfahren haben polynomiale Laufzeit.
- Für große Probleme sind sie oft schneller als Simplex, Degeneriertheit der Polyeder ist kaum ein Problem.
- Erkennen von Unzulässigkeit/Unbeschränktheit ist schwerer.
- Da Schnittebenen den zentralen Pfad großräumig verändern, ist kein guter Warmstart von der alten OL aus möglich.
→ erst cross-over, dann dualer Simplex
- Der gleiche Ansatz funktioniert sehr gut für lineare Programme über gutartigen nichtlinearen Kegeln!

Inhalt

Innere-Punkte-Verfahren und lineare Optimierung über Kegeln

Innere-Punkte-Verfahren

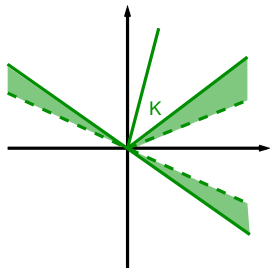
Lineare Optimierung über Kegeln

Second-Order-Cone Programme

Semidefinite Optimierung

5.2 Lineare Optimierung über Kegeln: Konvexe Kegel, dualer Kegel

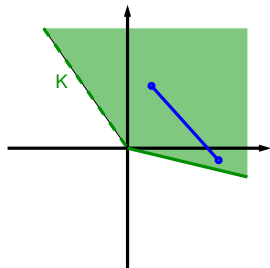
Eine Menge $K \subseteq \mathbb{R}^n$ heißt **Kegel**,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.



5.2 Lineare Optimierung über Kegeln: Konvexer Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **Kegel**,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **konvexer Kegel**,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.

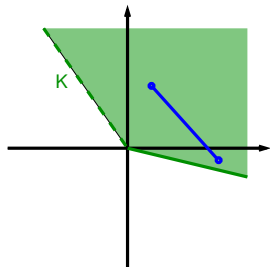


5.2 Lineare Optimierung über Kegeln: Konvexer Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **Kegel**,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **konvexer Kegel**,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.

Bspe: \emptyset , $\{0\}$, \mathbb{R}^n , \mathbb{R}_+^n ,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.

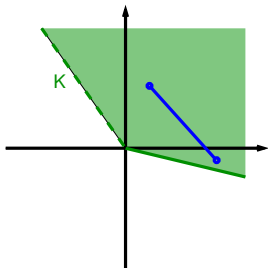


5.2 Lineare Optimierung über Kegeln: Konvexer Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **Kegel**,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **konvexer Kegel**,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.

Bspe: \emptyset , $\{0\}$, \mathbb{R}^n , \mathbb{R}_+^n ,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.



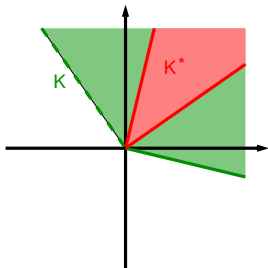
5.2 Lineare Optimierung über Kegeln: Konvexer Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **Kegel**,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **konvexer Kegel**,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.

Bspe: \emptyset , $\{0\}$, \mathbb{R}^n , \mathbb{R}_+^n ,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.

Ist $K \subseteq \mathbb{R}^n$ konvexer Kegel, nennt man
 $K^* := \{z \in \mathbb{R}^n : z^T x \geq 0 \forall x \in K\}$ den zu K **dualen Kegel**.



5.2 Lineare Optimierung über Kegeln: Konvexer Kegel, dualer Kegel

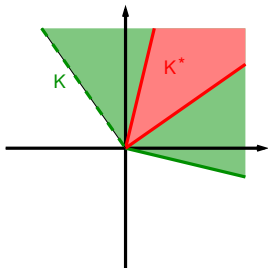
Eine Menge $K \subseteq \mathbb{R}^n$ heißt **Kegel**,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **konvexer Kegel**,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.

Bspe: \emptyset , $\{0\}$, \mathbb{R}^n , \mathbb{R}_+^n ,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.

Ist $K \subseteq \mathbb{R}^n$ konvexer Kegel, nennt man
 $K^* := \{z \in \mathbb{R}^n : z^T x \geq 0 \forall x \in K\}$ den zu K **dualen Kegel**.

Bspe: $(\mathbb{R}_+^n)^* = \mathbb{R}_+^n$, $\{0\}^* = \mathbb{R}^n$, $(\mathbb{R}^n)^* = \{0\}$, $(K_1 \times K_2)^* = K_1^* \times K_2^*$.



5.2 Lineare Optimierung über Kegeln: Konvexer Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **Kegel**,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

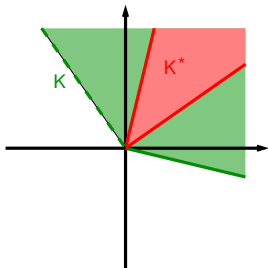
Eine Menge $K \subseteq \mathbb{R}^n$ heißt **konvexer Kegel**,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.

Bspe: \emptyset , $\{0\}$, \mathbb{R}^n , \mathbb{R}_+^n ,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.

Ist $K \subseteq \mathbb{R}^n$ konvexer Kegel, nennt man
 $K^* := \{z \in \mathbb{R}^n : z^T x \geq 0 \ \forall x \in K\}$ den zu K **dualen Kegel**.

Bspe: $(\mathbb{R}_+^n)^* = \mathbb{R}_+^n$, $\{0\}^* = \mathbb{R}^n$, $(\mathbb{R}^n)^* = \{0\}$, $(K_1 \times K_2)^* = K_1^* \times K_2^*$.
Für die Optimierung wichtig ist die Eigenschaft

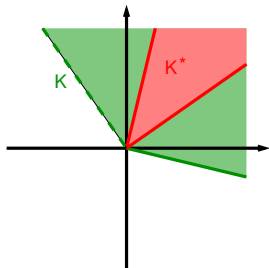
$$\inf_{x \in K} z^T x = \begin{cases} 0 & \Leftrightarrow z \in K^*, \\ -\infty & \text{sonst.} \end{cases}$$



5.2 Lineare Optimierung über Kegeln: Konvexer Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **Kegel**,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt **konvexer Kegel**,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.



Bspe: \emptyset , $\{0\}$, \mathbb{R}^n , \mathbb{R}_+^n ,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.

Ist $K \subseteq \mathbb{R}^n$ konvexer Kegel, nennt man
 $K^* := \{z \in \mathbb{R}^n : z^T x \geq 0 \forall x \in K\}$ den zu K **dualen Kegel**.

Bspe: $(\mathbb{R}_+^n)^* = \mathbb{R}_+^n$, $\{0\}^* = \mathbb{R}^n$, $(\mathbb{R}^n)^* = \{0\}$, $(K_1 \times K_2)^* = K_1^* \times K_2^*$.
Für die Optimierung wichtig ist die Eigenschaft

$$\inf_{x \in K} z^T x = \begin{cases} 0 & \Leftrightarrow z \in K^*, \\ -\infty & \text{sonst.} \end{cases}$$

Vorsicht! $(K^*)^* = K$ gilt NUR, wenn K ein abgeschlossener konvexer Kegel ist!

Primale/Duale lineare Programme über Kegeln

Ersetze in der linearen Optimierung \mathbb{R}_+^n durch konvexen Kegel $K \subseteq \mathbb{R}^n$:

$$\begin{array}{lll} \min & c^T x & \min & c^T x & \min & c^T x \\ \text{s.t.} & Ax = b & \Leftrightarrow & Ax = b & \rightarrow & Ax = b \\ & x \geq 0 & & x \in \mathbb{R}_+^n & & x \in K \end{array}$$

Primale/Duale lineare Programme über Kegeln

Ersetze in der linearen Optimierung \mathbb{R}_+^n durch konvexen Kegel $K \subseteq \mathbb{R}^n$:

$$\begin{array}{lll} \min & c^T x & \min & c^T x & \min & c^T x \\ \text{s.t.} & Ax = b & \Leftrightarrow & Ax = b & \rightarrow & Ax = b \\ & x \geq 0 & & x \in \mathbb{R}_+^n & & x \in K \end{array}$$

Definiere die Lagrange-Funktion

$$L(x, y) := c^T x + y^T (b - Ax) \quad \text{für } (x, y) \in K \times \mathbb{R}^m.$$

Für $y \in \mathbb{R}^m$ und $Ax = b$ ist $(b - Ax)^T y = 0$, daher gilt

$$\text{für alle } y \in \mathbb{R}^m : \quad \inf_{x \in K} L(x, y) \leq \inf \{c^T x : Ax = b, x \in K\}.$$

Primale/Duale lineare Programme über Kegeln

Ersetze in der linearen Optimierung \mathbb{R}_+^n durch konvexen Kegel $K \subseteq \mathbb{R}^n$:

$$\begin{array}{lll} \min & c^T x & \min & c^T x & \min & c^T x \\ \text{s.t.} & Ax = b & \Leftrightarrow & Ax = b & \rightarrow & Ax = b \\ & x \geq 0 & & x \in \mathbb{R}_+^n & & x \in K \end{array}$$

Definiere die Lagrange-Funktion

$$L(x, y) := c^T x + y^T (b - Ax) \quad \text{für } (x, y) \in K \times \mathbb{R}^m.$$

Für $y \in \mathbb{R}^m$ und $Ax = b$ ist $(b - Ax)^T y = 0$, daher gilt

$$\text{für alle } y \in \mathbb{R}^m : \quad \inf_{x \in K} L(x, y) \leq \inf \{c^T x : Ax = b, x \in K\}.$$

Die beste untere Schranke (Lagrange-Relaxation) liefert

$$\sup_{y \in \mathbb{R}^m} \inf_{x \in K} L(x, y) = \sup_{y \in \mathbb{R}^m} [b^T y + \inf_{x \in K} x^T (c - A^T y)]$$

Primale/Duale lineare Programme über Kegeln

Ersetze in der linearen Optimierung \mathbb{R}_+^n durch konvexen Kegel $K \subseteq \mathbb{R}^n$:

$$\begin{array}{lll} \min & c^T x & \min & c^T x & \min & c^T x \\ \text{s.t.} & Ax = b & \Leftrightarrow & Ax = b & \rightarrow & \text{s.t.} & Ax = b \\ & x \geq 0 & & x \in \mathbb{R}_+^n & & & x \in K \end{array}$$

Definiere die Lagrange-Funktion

$$L(x, y) := c^T x + y^T (b - Ax) \quad \text{für } (x, y) \in K \times \mathbb{R}^m.$$

Für $y \in \mathbb{R}^m$ und $Ax = b$ ist $(b - Ax)^T y = 0$, daher gilt

$$\text{für alle } y \in \mathbb{R}^m : \quad \inf_{x \in K} L(x, y) \leq \inf \{c^T x : Ax = b, x \in K\}.$$

Die beste untere Schranke (Lagrange-Relaxation) liefert

$$\sup_{y \in \mathbb{R}^m} \inf_{x \in K} L(x, y) = \sup_{y \in \mathbb{R}^m} [b^T y + \inf_{x \in K} x^T (c - A^T y)]$$

Das inf ist nur für $z = c - A^T y \in K^*$ endlich, das duale Programm lautet

$$\begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \in K^* \end{array}$$

Schwache und starke Dualität

Sei $K \subseteq \mathbb{R}^n$ ein konvexer Kegel.

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \in K \end{array} \quad (P) \qquad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \in K^* \end{array} \quad (D)$$

Nach Konstruktion gilt immer **schwache Dualität**, $v(P) \geq v(D)$,
Gleichheit gilt keineswegs immer (Beispiel später)!

Für starke Dualität braucht es Zusatzbedingungen:

Schwache und starke Dualität

Sei $K \subseteq \mathbb{R}^n$ ein konvexer Kegel.

$$\begin{array}{ll}
 \min & c^T x \\
 \text{s.t.} & Ax = b \\
 & x \in K
 \end{array}
 \quad (P)
 \qquad
 \begin{array}{ll}
 \max & b^T y \\
 \text{s.t.} & A^T y + z = c \\
 & y \in \mathbb{R}^m, z \in K^*
 \end{array}
 \quad (D)$$

Nach Konstruktion gilt immer **schwache Dualität**, $v(P) \geq v(D)$,
Gleichheit gilt keineswegs immer (Beispiel später)!

Für starke Dualität braucht es Zusatzbedingungen:

Ein primal zulässiges \bar{x} heißt **streng zulässig** für (P) , wenn \bar{x} im Inneren von K liegt ($\exists \rho > 0 : B_\rho(\bar{x}) := \{x \in \mathbb{R}^n : \|x - \bar{x}\| \leq \rho\} \subseteq K$). Gibt es so ein \bar{x} , heißt auch (P) **streng zulässig**.

Schwache und starke Dualität

Sei $K \subseteq \mathbb{R}^n$ ein konvexer Kegel.

$$\begin{array}{ll}
 \min & c^T x \\
 \text{(P)} \quad \text{s.t.} & Ax = b \\
 & x \in K
 \end{array}
 \qquad
 \begin{array}{ll}
 \max & b^T y \\
 \text{(D)} \quad \text{s.t.} & A^T y + z = c \\
 & y \in \mathbb{R}^m, z \in K^*
 \end{array}$$

Nach Konstruktion gilt immer **schwache Dualität**, $v(P) \geq v(D)$, Gleichheit gilt keineswegs immer (Beispiel später)!

Für starke Dualität braucht es Zusatzbedingungen:

Ein primal zulässiges \bar{x} heißt **streng zulässig** für (P), wenn \bar{x} im Inneren von K liegt ($\exists \rho > 0 : B_\rho(\bar{x}) := \{x \in \mathbb{R}^n : \|x - \bar{x}\| \leq \rho\} \subseteq K$). Gibt es so ein \bar{x} , heißt auch (P) **streng zulässig**.

Ein dual zulässiges (\bar{y}, \bar{z}) heißt **streng zulässig** für (D), wenn \bar{z} im Inneren von K^* liegt ($\exists \rho > 0 : B_\rho(\bar{z}) \subseteq K^*$). Gibt es so ein (\bar{y}, \bar{z}) , heißt auch (D) **streng zulässig**.

Satz (Starke Dualität)

Ist (P) streng zulässig, wird das duale Optimum $v(D)$ angenommen.

Ist (D) streng zulässig, wird das primale Optimum $v(P)$ angenommen.

In beiden Fällen gilt $v(P) = v(D)$.

Selbstduale Kegel, Innere-Punkte-Verfahren

Hier werden nur drei spezielle Kegelarten K verwendet:

- $K = \mathbb{R}_+^n$, der nichtnegative Orthant
- $K = \mathcal{Q}^n$, der quadratische Kegel
- $K = S_+^n$ der Kegel der positiv semidefiniten Matrizen

Die genauen Definitionen von \mathcal{Q}^n und S_+^n folgen demnächst.

Selbstduale Kegel, Innere-Punkte-Verfahren

Hier werden nur drei spezielle Kegelarten K verwendet:

- $K = \mathbb{R}_+^n$, der nichtnegative Orthant
- $K = \mathcal{Q}^n$, der quadratische Kegel
- $K = S_+^n$ der Kegel der positiv semidefiniten Matrizen

Die genauen Definitionen von \mathcal{Q}^n und S_+^n folgen demnächst.

Die wichtigsten Eigenschaften dieser drei sind:

- Sie sind **selbstdual**, d.h., $K = K^*$.
- Es sind geeignete Barriere-Funktionen für sie bekannt.
- Gute Innere-Punkte-Codes, die die Verwendung der drei gleichzeitig erlauben, sind für MATLAB verfügbar (SeDuMi, SDPT3).
- Sie ermöglichen viele wichtige Anwendungen zu modellieren.

Selbstduale Kegel, Innere-Punkte-Verfahren

Hier werden nur drei spezielle Kegelarten K verwendet:

- $K = \mathbb{R}_+^n$, der nichtnegative Orthant
- $K = \mathcal{Q}^n$, der quadratische Kegel
- $K = S_+^n$ der Kegel der positiv semidefiniten Matrizen

Die genauen Definitionen von \mathcal{Q}^n und S_+^n folgen demnächst.

Die wichtigsten Eigenschaften dieser drei sind:

- Sie sind **selbstdual**, d.h., $K = K^*$.
- Es sind geeignete Barriere-Funktionen für sie bekannt.
- Gute Innere-Punkte-Codes, die die Verwendung der drei gleichzeitig erlauben, sind für MATLAB verfügbar (SeDuMi, SDPT3).
- Sie ermöglichen viele wichtige Anwendungen zu modellieren.

In den Anwendung ist K oft aus mehreren Einzelkegeln zusammengesetzt,

$$K = \mathbb{R}_+^{n_1} \times \mathcal{Q}^{m_1} \times \dots \times \mathcal{Q}^{m_k} \times S_+^{n_1} \times \dots \times S_+^{n_h}$$

Das wird sich aber natürlich ergeben und $K = K^*$ gilt für diese Kombinationen immer.

Inhalt

Innere-Punkte-Verfahren und lineare Optimierung über Kegeln

Innere-Punkte-Verfahren

Lineare Optimierung über Kegeln

Second-Order-Cone Programme

SOC Anwendung: Regularisierung

SOC Anwendung: Klassifizierung, Support-Vektor

SOC Anwendung: Das Markowitz Modell, Chance Constraints

Semidefinite Optimierung

5.3 Der quadratische Kegel (Second Order Cone)

Der **quadratische Kegel**
oder **Second-Order-Cone (SOC)**

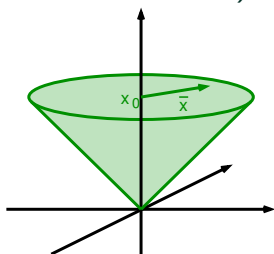
$$\mathcal{Q}^n = \left\{ \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \in \mathbb{R}^{n+1} : x_0 \geq \|\bar{x}\| \right\}$$

ist ein konvexer Kegel,

denn für $x, y \in \mathcal{Q}^n$, $\alpha \geq 0$ ist

$$\|\alpha(\bar{x} + \bar{y})\| \leq \alpha\|\bar{x}\| + \alpha\|\bar{y}\| \leq \alpha(x_0 + y_0).$$

\mathcal{Q}^n ist **selbstdual**, $(\mathcal{Q}^n)^* = \mathcal{Q}^n$.



5.3 Der quadratische Kegel (Second Order Cone)

Der **quadratische Kegel**
oder **Second-Order-Cone (SOC)**

$$\mathcal{Q}^n = \left\{ \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \in \mathbb{R}^{n+1} : x_0 \geq \|\bar{x}\| \right\}$$

ist ein konvexer Kegel,

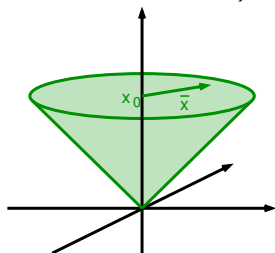
denn für $x, y \in \mathcal{Q}^n$, $\alpha \geq 0$ ist

$$\|\alpha(\bar{x} + \bar{y})\| \leq \alpha\|\bar{x}\| + \alpha\|\bar{y}\| \leq \alpha(x_0 + y_0).$$

\mathcal{Q}^n ist **selbstdual**, $(\mathcal{Q}^n)^* = \mathcal{Q}^n$.

Statt $x \in \mathcal{Q}^n$ schreiben wir auch $x \geq_{\mathcal{Q}} 0$. Für $a, b \in \mathbb{R}^{n+1}$ ist $a \geq_{\mathcal{Q}} b$ gleichbedeutend mit $a - b \geq_{\mathcal{Q}} 0$, also $a - b \in \mathcal{Q}^n$.

Innere-Punkte-Verfahren verwenden $-\log(x_0^2 - \bar{x}^T \bar{x})$ als Barriere.



5.3 Der quadratische Kegel (Second Order Cone)

Der **quadratische Kegel**
oder **Second-Order-Cone** (SOC)

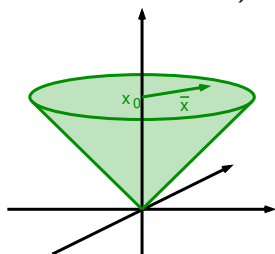
$$Q^n = \left\{ \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \in \mathbb{R}^{n+1} : x_0 \geq \|\bar{x}\| \right\}$$

ist ein konvexer Kegel,

denn für $x, y \in Q^n$, $\alpha \geq 0$ ist

$$\|\alpha(\bar{x} + \bar{y})\| \leq \alpha\|\bar{x}\| + \alpha\|\bar{y}\| \leq \alpha(x_0 + y_0).$$

Q^n ist **selbstdual**, $(Q^n)^* = Q^n$.



Statt $x \in Q^n$ schreiben wir auch $x \geq_Q 0$. Für $a, b \in \mathbb{R}^{n+1}$ ist $a \geq_Q b$ gleichbedeutend mit $a - b \geq_Q 0$, also $a - b \in Q^n$.

Innere-Punkte-Verfahren verwenden $-\log(x_0^2 - \bar{x}^T \bar{x})$ als Barriere.

Lineare Programme, die als Kegel nur \mathbb{R}_+^n und mindestens ein Q^n einsetzen, heißen **Second-Order-Cone Programme** (kurz SOCP).

Ein SOCP mit nur einem Q^n liest sich

$$\begin{array}{ll}
 \min & c^T x \\
 \text{(P)} & \text{s.t. } Ax = b \\
 & x \geq_Q 0
 \end{array}
 \qquad
 \begin{array}{ll}
 \max & b^T y \\
 \text{(D)} & \text{s.t. } A^T y + z = c \\
 & y \in \mathbb{R}^m, z \geq_Q 0
 \end{array}$$

[Für nur einen einzelnen SOC ist es sogar explizit lösbar.]

Ein SOCP-Beispiel mit Dualitätslücke

$$(P) \quad \begin{array}{ll} \min & -x_2 \\ \text{s.t.} & x_0 - x_1 = 0 \\ & x_0 \geq \sqrt{x_1^2 + x_2^2} \end{array}$$

$$(D) \quad \begin{array}{ll} \max & 0 \\ \text{s.t.} & y_1 + z_0 = 0 \\ & -y_1 + z_1 = 0 \\ & z_2 = -1 \\ & z_0 \geq \sqrt{z_1^2 + z_2^2} \end{array}$$

Ein SOCP-Beispiel mit Dualitätslücke

$$\begin{array}{ll}
 \min & -x_2 \\
 \text{s.t.} & x_0 - x_1 = 0 \\
 & x_0 \geq \sqrt{x_1^2 + x_2^2}
 \end{array}
 \quad (P)
 \qquad
 \begin{array}{ll}
 \max & 0 \\
 \text{s.t.} & y_1 + z_0 = 0 \\
 & -y_1 + z_1 = 0 \\
 & z_2 = -1 \\
 & z_0 \geq \sqrt{z_1^2 + z_2^2}
 \end{array}
 \quad (D)$$

In (P) folgt aus $x_0 = x_1$ und $x \in Q^2$, dass $x_2 = 0$, also $v(P) = 0$.

Ein SOCP-Beispiel mit Dualitätslücke

$$\begin{array}{ll}
 \min & -x_2 \\
 \text{s.t.} & x_0 - x_1 = 0 \\
 & x_0 \geq \sqrt{x_1^2 + x_2^2}
 \end{array}
 \quad (P)
 \qquad
 \begin{array}{ll}
 \max & 0 \\
 \text{s.t.} & y_1 + z_0 = 0 \\
 & -y_1 + z_1 = 0 \\
 & z_2 = -1 \\
 & z_0 \geq \sqrt{z_1^2 + z_2^2}
 \end{array}
 \quad (D)$$

In (P) folgt aus $x_0 = x_1$ und $x \in \mathcal{Q}^2$, dass $x_2 = 0$, also $v(P) = 0$.

In (D) folgt aus $z_0 = -y_1 = -z_1$ und $z \in \mathcal{Q}^2$, dass $z_2 = 0$,
in Widerspruch zu $z_2 = -1$, also $v(D) = -\infty$.

Ein SOCP-Beispiel mit Dualitätslücke

$$\begin{array}{ll}
 \min & -x_2 \\
 \text{s.t.} & x_0 - x_1 = 0 \\
 & x_0 \geq \sqrt{x_1^2 + x_2^2}
 \end{array}
 \quad (P)
 \qquad
 \begin{array}{ll}
 \max & 0 \\
 \text{s.t.} & y_1 + z_0 = 0 \\
 & -y_1 + z_1 = 0 \\
 & z_2 = -1 \\
 & z_0 \geq \sqrt{z_1^2 + z_2^2}
 \end{array}
 \quad (D)$$

In (P) folgt aus $x_0 = x_1$ und $x \in \mathcal{Q}^2$, dass $x_2 = 0$, also $v(P) = 0$.

In (D) folgt aus $z_0 = -y_1 = -z_1$ und $z \in \mathcal{Q}^2$, dass $z_2 = 0$,
in Widerspruch zu $z_2 = -1$, also $v(D) = -\infty$.

Beachte: $v(P) \neq v(D)$ geht nur, wenn beide, Primales und Duales, keinen streng zulässigen Punkt haben.

Ein SOCP-Beispiel mit Dualitätslücke

$$\begin{array}{ll}
 \min & -x_2 \\
 \text{s.t.} & x_0 - x_1 = 0 \\
 & x_0 \geq \sqrt{x_1^2 + x_2^2}
 \end{array}
 \quad (P)
 \qquad
 \begin{array}{ll}
 \max & 0 \\
 \text{s.t.} & y_1 + z_0 = 0 \\
 & -y_1 + z_1 = 0 \\
 & z_2 = -1 \\
 & z_0 \geq \sqrt{z_1^2 + z_2^2}
 \end{array}
 \quad (D)$$

In (P) folgt aus $x_0 = x_1$ und $x \in \mathcal{Q}^2$, dass $x_2 = 0$, also $v(P) = 0$.

In (D) folgt aus $z_0 = -y_1 = -z_1$ und $z \in \mathcal{Q}^2$, dass $z_2 = 0$,
in Widerspruch zu $z_2 = -1$, also $v(D) = -\infty$.

Beachte: $v(P) \neq v(D)$ geht nur, wenn beide, Primales und Duales, keinen streng zulässigen Punkt haben.

In Anwendungen hat eine SOC-Nebenbedingung meist die Struktur

$$\begin{array}{ll}
 \min & c^T x \\
 \text{s.t.} & g^T x + d \geq \|Ax - b\| \\
 & x \in \mathbb{R}^n
 \end{array}$$

Ein SOCP-Beispiel mit Dualitätslücke

$$\begin{array}{ll}
 \min & -x_2 \\
 \text{s.t.} & x_0 - x_1 = 0 \\
 & x_0 \geq \sqrt{x_1^2 + x_2^2}
 \end{array}
 \quad (P)
 \qquad
 \begin{array}{ll}
 \max & 0 \\
 \text{s.t.} & y_1 + z_0 = 0 \\
 & -y_1 + z_1 = 0 \\
 & z_2 = -1 \\
 & z_0 \geq \sqrt{z_1^2 + z_2^2}
 \end{array}
 \quad (D)$$

In (P) folgt aus $x_0 = x_1$ und $x \in \mathcal{Q}^2$, dass $x_2 = 0$, also $v(P) = 0$.

In (D) folgt aus $z_0 = -y_1 = -z_1$ und $z \in \mathcal{Q}^2$, dass $z_2 = 0$,
in Widerspruch zu $z_2 = -1$, also $v(D) = -\infty$.

Beachte: $v(P) \neq v(D)$ geht nur, wenn beide, Primales und Duales, keinen streng zulässigen Punkt haben.

In Anwendungen hat eine SOC-Nebenbedingung meist die Struktur

$$\begin{array}{ll}
 \min & c^T x \\
 \text{s.t.} & g^T x + d \geq \|Ax - b\| \\
 & x \in \mathbb{R}^n
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{ll}
 \min & c^T x \\
 \text{s.t.} & x_0 = g^T x + d \\
 & \bar{x} = Ax - b \\
 & \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \geq_{\mathcal{Q}} 0, x \in \mathbb{R}^n
 \end{array}$$

5.3.1 SOC Anwendung: Regularisierte Approximation

Ein Datenvektor $b \in \mathbb{R}^m$ soll einerseits möglichst gut durch eine Linearkombination $\sum_i a^{(i)} x_i$ von Grundfunktionen $a^{(i)} \in \mathbb{R}^m$ dargestellt werden ($\rightarrow \|Ax - b\|$), andererseits möchte man die „Größe“ von x möglichst klein halten oder Eigenschaften von Ax kontrollieren ($\rightarrow f(x)$).

5.3.1 SOC Anwendung: Regularisierte Approximation

Ein Datenvektor $b \in \mathbb{R}^m$ soll einerseits möglichst gut durch eine Linearkombination $\sum_i a^{(i)} x_i$ von Grundfunktionen $a^{(i)} \in \mathbb{R}^m$ dargestellt werden ($\rightarrow \|Ax - b\|$), andererseits möchte man die „Größe“ von x möglichst klein halten oder Eigenschaften von Ax kontrollieren ($\rightarrow f(x)$).

Dieses bikriterielle Problem wird mit Parameter $\gamma > 0$ **skalarisiert** zu

$$\min_{x \in \mathbb{R}_+^n} \|Ax - b\| + \gamma f(x)$$

5.3.1 SOC Anwendung: Regularisierte Approximation

Ein Datenvektor $b \in \mathbb{R}^m$ soll einerseits möglichst gut durch eine Linearkombination $\sum_i a^{(i)} x_i$ von Grundfunktionen $a^{(i)} \in \mathbb{R}^m$ dargestellt werden ($\rightarrow \|Ax - b\|$), andererseits möchte man die „Größe“ von x möglichst klein halten oder Eigenschaften von Ax kontrollieren ($\rightarrow f(x)$).

Dieses bikriterielle Problem wird mit Parameter $\gamma > 0$ **skalarisiert** zu

$$\min_{x \in \mathbb{R}_+^n} \|Ax - b\| + \gamma f(x)$$

Für Anwendungen in der Signalrekonstruktion/-glättung/-entstörung ist

- $b_j = h(t_j)$ das empfangene Signal zum Zeitpunkt t_j
- $a_j^{(i)} = h_i(t_j)$ der Wert der Basisfunktion h_i zum Zeitpunkt t_j
- x_i der zu ermittelnde Anteil der Basisfunktion h_i am Signal

Die Wahl von f unterscheidet sich je nach Zielen, hier:

- Glättung bei Sprunganteilen
- Reduktion der Zahl verwendeter Basisfunktionen

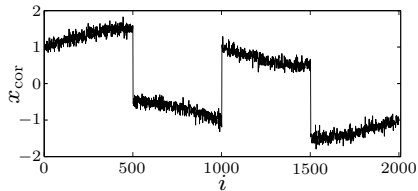
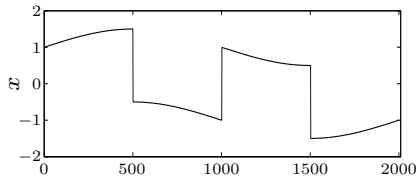
Glättung bei Sprunganteilen

Bei Glättung bestraft f normalerweise starke Änderungen in Ax , bei quadratischer Glättung wäre $f(x) = \sum_{j=1}^{m-1} ([Ax]_{j+1} - [Ax]_j)^2$, aber dadurch werden Sprünge „herausgeglättet“.

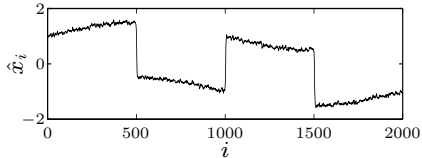
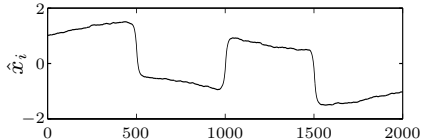
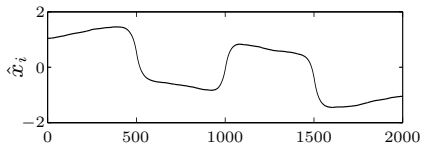
[Folie von Boyd, <http://www.stanford.edu/~boyd/cvxbook/>, 25.10.2009]

Andere Notation!

total variation reconstruction example



original signal x and noisy
signal x_{cor}



three solutions on trade-off curve
 $\|\hat{x} - x_{\text{cor}}\|_2$ versus $\phi_{\text{quad}}(\hat{x})$

Glättung bei Sprunganteilen

Bei Glättung bestraft f normalerweise starke Änderungen in Ax , bei quadratischer Glättung wäre $f(x) = \sum_{j=1}^{m-1} ([Ax]_{j+1} - [Ax]_j)^2$, aber dadurch werden Sprünge „herausgeglättet“.

Sollen Sprunganteile erhalten bleiben, verwendet man besser die 1-Norm,

$$f(x) = \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j|,$$

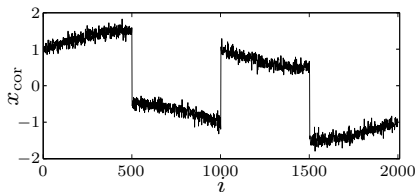
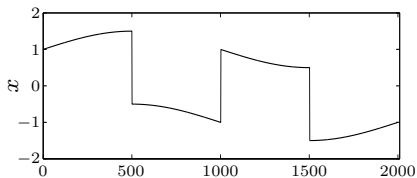
für diese dürfen gerne große Terme an wenigen Stellen auftreten, solange dadurch die Summe insgesamt nicht größer wird.

$$\min_{x \in \mathbb{R}_+^n} \|Ax - b\| + \gamma \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j|$$

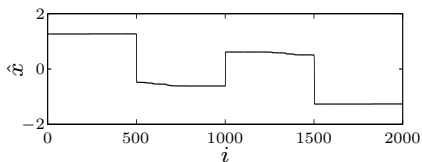
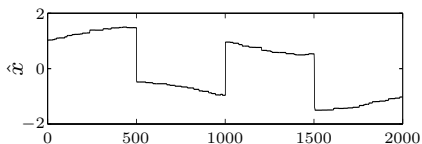
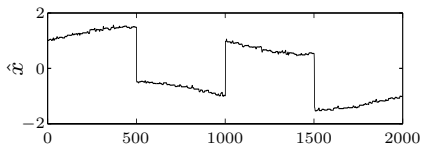
ist ein SOCP:

[Folie von Boyd, <http://www.stanford.edu/~boyd/cvxbook/>, 25.10.2009]

Andere Notation!



original signal x and noisy
signal x_{cor}



three solutions on trade-off curve
 $\|\hat{x} - x_{\text{cor}}\|_2$ versus $\phi_{\text{tv}}(\hat{x})$

Glättung bei Sprunganteilen

Bei Glättung bestraft f normalerweise starke Änderungen in Ax , bei quadratischer Glättung wäre $f(x) = \sum_{j=1}^{m-1} ([Ax]_{j+1} - [Ax]_j)^2$, aber dadurch werden Sprünge „herausgeglättet“.

Sollen Sprunganteile erhalten bleiben, verwendet man besser die 1-Norm,

$$f(x) = \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j|,$$

für diese dürfen gerne große Terme an wenigen Stellen auftreten, solange dadurch die Summe insgesamt nicht größer wird.

$$\min_{x \in \mathbb{R}_+^n} \|Ax - b\| + \gamma \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j|$$

ist ein SOCP:

Glättung bei Sprunganteilen

Bei Glättung bestraft f normalerweise starke Änderungen in Ax , bei quadratischer Glättung wäre $f(x) = \sum_{j=1}^{m-1} ([Ax]_{j+1} - [Ax]_j)^2$, aber dadurch werden Sprünge „herausgeglättet“.

Sollen Sprunganteile erhalten bleiben, verwendet man besser die 1-Norm,

$$f(x) = \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j|,$$

für diese dürfen gerne große Terme an wenigen Stellen auftreten, solange dadurch die Summe insgesamt nicht größer wird.

$$\min_{x \in \mathbb{R}^n_+} \|Ax - b\| + \gamma \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j|$$

ist ein SOCP:

[Notation: $A_{j,\bullet}$... Zeile j von A]

$$\begin{aligned} \min \quad & \xi_0 + \gamma \sum_{j=1}^{m-1} s_j \\ \text{s.t.} \quad & \bar{\xi} - Ax = -b \\ & -s_j \leq (A_{j+1,\bullet} - A_{j,\bullet})x \leq s_j \quad j = 1, \dots, m-1 \\ & \begin{bmatrix} \xi_0 \\ \bar{\xi} \end{bmatrix} \geq_{\mathcal{Q}} 0, x \geq 0, s \geq 0 \end{aligned}$$

Reduktion der Zahl verwendeter Basisfunktionen

Wenn man weiß, dass sich das Signal nur aus wenigen Basisfunktionen zusammensetzt oder zur Datenkompression nur wenige verwenden will, möchte man ein gemischt-ganzzahliges nichtlineares Problem lösen:

$$\min_{x \in \mathbb{R}_+^n} \|Ax - b\| + \gamma |\{i = 1, \dots, n : x_i > 0\}|.$$

Reduktion der Zahl verwendeter Basisfunktionen

Wenn man weiß, dass sich das Signal nur aus wenigen Basisfunktionen zusammensetzt oder zur Datenkompression nur wenige verwenden will, möchte man ein gemischt-ganzzahliges nichtlineares Problem lösen:

$$\min_{x \in \mathbb{R}_+^n} \|Ax - b\| + \gamma |\{i = 1, \dots, n : x_i > 0\}|.$$

Dies ist zu aufwendig, daher gibt man sich mit der 1-Norm zufrieden,

$$\min_{x \in \mathbb{R}_+^n} \|Ax - b\| + \gamma \sum_{i=1}^n |x_i|,$$

und hofft, dass diese den Wert auf nur wenige Koordinaten konzentriert.

Übung: Erstelle das SOCP zur 1-Norm Variante.

Reduktion der Zahl verwendeter Basisfunktionen

Wenn man weiß, dass sich das Signal nur aus wenigen Basisfunktionen zusammensetzt oder zur Datenkompression nur wenige verwenden will, möchte man ein gemischt-ganzzahliges nichtlineares Problem lösen:

$$\min_{x \in \mathbb{R}_+^n} \|Ax - b\| + \gamma |\{i = 1, \dots, n : x_i > 0\}|.$$

Dies ist zu aufwendig, daher gibt man sich mit der 1-Norm zufrieden,

$$\min_{x \in \mathbb{R}_+^n} \|Ax - b\| + \gamma \sum_{i=1}^n |x_i|,$$

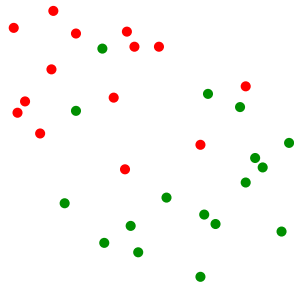
und hofft, dass diese den Wert auf nur wenige Koordinaten konzentriert.

Übung: Erstelle das SOCP zur 1-Norm Variante.

Bei bikriteriellen Aufgaben bietet es sich an, das Problem für mehrere Parameterwerte $\gamma > 0$ zu lösen und die jeweils erzeugten Funktionswertpaare in einer Graphik miteinander zu vergleichen.

5.3.2 Klassifizierung, Support-Vektor

Für Datenpunkte im \mathbb{R}^n , die eine bestimmte Eigenschaft haben bzw. nicht haben, soll eine Ebene gefunden werden, die die Punkttypen „möglichst gut“ voneinander trennt (Ziel: neue Punkte klassifizieren).



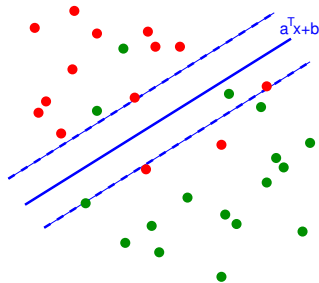
5.3.2 Klassifizierung, Support-Vektor

Für Datenpunkte im \mathbb{R}^n , die eine bestimmte Eigenschaft haben bzw. nicht haben, soll eine Ebene gefunden werden, die die Punkttypen „möglichst gut“ voneinander trennt (Ziel: neue Punkte klassifizieren).

Gegeben zwei disjunkte endliche Mengen $G, R \subset \mathbb{R}^n$, finde $a^T x + b$ (mit Variablen a und b) mit „möglichst“ $a^T x + b \geq 1$ für $x \in G$ und $a^T x + b \leq -1$ für $x \in R$.

Schwierigkeiten:

- Für gute Trennung soll $\|a\|$ klein sein.
- Was tun, wenn Fehlklassifizierungen unvermeidbar sind?



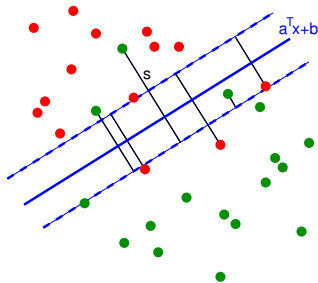
5.3.2 Klassifizierung, Support-Vektor

Für Datenpunkte im \mathbb{R}^n , die eine bestimmte Eigenschaft haben bzw. nicht haben, soll eine Ebene gefunden werden, die die Punkttypen „möglichst gut“ voneinander trennt (Ziel: neue Punkte klassifizieren).

Gegeben zwei disjunkte endliche Mengen $G, R \subset \mathbb{R}^n$, finde $a^T x + b$ (mit Variablen a und b) mit „möglichst“ $a^T x + b \geq 1$ für $x \in G$ und $a^T x + b \leq -1$ für $x \in R$.

Schwierigkeiten:

- Für gute Trennung soll $\|a\|$ klein sein.
- Was tun, wenn Fehlklassifizierungen unvermeidbar sind?



Ein Ansatz: Minimiere gleichzeitig $\|a\|$ und die Summe der Verletzungen der Ungleichungsbedingungen, skalarisiert mit Parameter $\gamma > 0$,

$$\begin{aligned}
 \min \quad & \|a\| + \gamma \sum_{x \in G \cup R} s_x \\
 \text{s.t.} \quad & x^T a - b \geq 1 - s_x \quad x \in G \\
 & x^T a - b \leq s_x - 1 \quad x \in R \\
 & a \in \mathbb{R}^n, b \in \mathbb{R}, s \in \mathbb{R}_+^{G \cup R}
 \end{aligned}$$

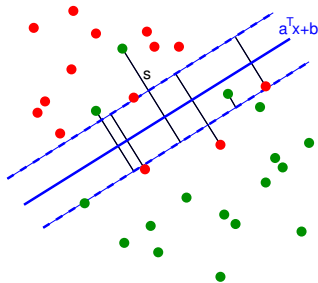
5.3.2 Klassifizierung, Support-Vektor

Für Datenpunkte im \mathbb{R}^n , die eine bestimmte Eigenschaft haben bzw. nicht haben, soll eine Ebene gefunden werden, die die Punkttypen „möglichst gut“ voneinander trennt (Ziel: neue Punkte klassifizieren).

Gegeben zwei disjunkte endliche Mengen $G, R \subset \mathbb{R}^n$, finde $a^T x + b$ (mit Variablen a und b) mit „möglichst“ $a^T x + b \geq 1$ für $x \in G$ und $a^T x + b \leq -1$ für $x \in R$.

Schwierigkeiten:

- Für gute Trennung soll $\|a\|$ klein sein.
- Was tun, wenn Fehlklassifizierungen unvermeidbar sind?



Ein Ansatz: Minimiere gleichzeitig $\|a\|$ und die Summe der Verletzungen der Ungleichungsbedingungen, skalarisiert mit Parameter $\gamma > 0$,

$$\begin{array}{ll}
 \min & \|a\| + \gamma \sum_{x \in G \cup R} s_x \\
 \text{s.t.} & x^T a - b \geq 1 - s_x \quad x \in G \\
 & x^T a - b \leq s_x - 1 \quad x \in R \\
 & a \in \mathbb{R}^n, b \in \mathbb{R}, s \in \mathbb{R}_+^{G \cup R}
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{ll}
 \min & a_0 + \gamma \sum_{x \in G \cup R} s_x \\
 \text{s.t.} & x^T a - b \geq 1 - s_x \quad x \in G \\
 & x^T a - b \leq s_x - 1 \quad x \in R \\
 & \begin{bmatrix} a_0 \\ a \end{bmatrix} \succeq_Q 0, b \in \mathbb{R}, s \geq 0
 \end{array}$$

5.3.3 Das Markowitz Modell

Im Markowitz-Modell der Portfolio-Optimierung soll gegebenes Kapital so mit einem erwarteten Mindestgewinn investiert werden, dass das Risiko minimiert wird.

5.3.3 Das Markowitz Modell

Im Markowitz-Modell der Portfolio-Optimierung soll gegebenes Kapital so mit einem erwarteten Mindestgewinn investiert werden, dass das Risiko minimiert wird.

$x \in \mathbb{R}_+^n$ mit $\mathbf{1}^T x = 1$ gibt den Anteil des Budgets an, der in Aktien $1, \dots, n$ investiert wird. Der Gewinn g pro Anteil ist eine Zufallsvariable mit Erwartungswert $\bar{g} \in \mathbb{R}^n$ und Kovarianzmatrix $G \in S_+^n$ ($n \times n$, positiv semidefinit), $s \in \mathbb{R}$ sei ein gegebener Schwellwert. Als Maß für das Risiko sieht das Markowitz-Modell $x^T G x$ vor. [Andere Maße möglich!]

$$\begin{aligned} \min \quad & x^T G x \\ \text{s.t.} \quad & \bar{g}^T x \geq s \\ & \mathbf{1}^T x = 1 \\ & x \in \mathbb{R}_+^n \end{aligned}$$

5.3.3 Das Markowitz Modell

Im Markowitz-Modell der Portfolio-Optimierung soll gegebenes Kapital so mit einem erwarteten Mindestgewinn investiert werden, dass das Risiko minimiert wird.

$x \in \mathbb{R}_+^n$ mit $\mathbf{1}^T x = 1$ gibt den Anteil des Budgets an, der in Aktien $1, \dots, n$ investiert wird. Der Gewinn g pro Anteil ist eine Zufallsvariable mit Erwartungswert $\bar{g} \in \mathbb{R}^n$ und Kovarianzmatrix $G \in S_+^n$ ($n \times n$, positiv semidefinit), $s \in \mathbb{R}$ sei ein gegebener Schwellwert. Als Maß für das Risiko sieht das Markowitz-Modell $x^T G x$ vor. [Andere Maße möglich!]

$$\begin{aligned} \min \quad & x^T G x \\ \text{s.t.} \quad & \bar{g}^T x \geq s \\ & \mathbf{1}^T x = 1 \\ & x \in \mathbb{R}_+^n \end{aligned}$$

Wegen G positiv semidefinit ist das ein konvex quadratisches Problem. [Das bikriterielle Ziel Gewinn gegen Risiko wird hier durch eine Nebenbedingung an eines der Kriterien umgesetzt.]

Modellierung als SOCP?

Quadratische Nebenbedingungen mit SOCP

Sei $Q \in S_+^n$ positiv definit, $q \in \mathbb{R}^n$, $d \in \mathbb{R}$. Die konvex quadratische Nebenbedingung

$$\frac{1}{2}x^T Qx + q^T x + d \leq 0$$

lässt sich mit $Q = Q^{\frac{1}{2}} Q^{\frac{1}{2}}$ über

$$\|Q^{\frac{1}{2}}x + Q^{-\frac{1}{2}}q\| \leq \sqrt{q^T Q^{-1}q - 2d}$$

als SOC-Nebenbedingung schreiben (Beweis: beide Seiten quadrieren).

[Eigenwertzerlegung $Q = P\Lambda P^T \rightarrow Q^{\frac{1}{2}} = P\Lambda^{\frac{1}{2}}P^T$, $\Lambda = \text{Diag}(\lambda_1, \dots, \lambda_n) \geq 0$]

Quadratische Nebenbedingungen mit SOCP

Sei $Q \in S_+^n$ positiv definit, $q \in \mathbb{R}^n$, $d \in \mathbb{R}$. Die konvex quadratische Nebenbedingung

$$\frac{1}{2}x^T Qx + q^T x + d \leq 0$$

lässt sich mit $Q = Q^{\frac{1}{2}} Q^{\frac{1}{2}}$ über

$$\|Q^{\frac{1}{2}}x + Q^{-\frac{1}{2}}q\| \leq \sqrt{q^T Q^{-1}q - 2d}$$

als SOC-Nebenbedingung schreiben (Beweis: beide Seiten quadrieren).

[Eigenwertzerlegung $Q = P\Lambda P^T \rightarrow Q^{\frac{1}{2}} = P\Lambda^{\frac{1}{2}}P^T$, $\Lambda = \text{Diag}(\lambda_1, \dots, \lambda_n) \geq 0$]

Für das Markowitz-Modell ist es einfacher, $x_0 \geq \|G^{\frac{1}{2}}x\|$ reicht aus

$$\begin{aligned} \min \quad & x_0 \\ \text{s.t.} \quad & \bar{x} = G^{\frac{1}{2}}x \\ & \bar{g}^T x \geq s \\ & \mathbf{1}^T x = 1 \\ & \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \geq_Q 0, x \geq 0 \end{aligned}$$

Probabilistische Nebenbedingung, Chance Constraint

Unter der Annahme, dass g mit Varianz G um \bar{g} normalverteilt ist, sollen nun zusätzlich nur Investitionspläne erlaubt werden, für die die Wahrscheinlichkeit, dass der Gewinn über einem Schwellwert $\underline{s} < s$ bleibt, mindestens $\eta \in (0, 1)$ ist,

$$\mathbb{P}(g^T x \geq \underline{s}) \geq \eta$$

Probabilistische Nebenbedingung, Chance Constraint

Unter der Annahme, dass g mit Varianz G um \bar{g} normalverteilt ist, sollen nun zusätzlich nur Investitionspläne erlaubt werden, für die die Wahrscheinlichkeit, dass der Gewinn über einem Schwellwert $\underline{s} < s$ bleibt, mindestens $\eta \in (0, 1)$ ist,

$$\mathbb{P}(g^T x \geq \underline{s}) \geq \eta \quad \rightarrow \quad \begin{array}{ll} \min & x^T G x \\ \text{s.t.} & \bar{g}^T x \geq s \\ & \mathbb{P}(g^T x \geq \underline{s}) \geq \eta \\ & \mathbf{1}^T x = 1 \\ & x \in \mathbb{R}_+^n \end{array}$$

Dies modelliert man mit einer Technik der robusten Optimierung: $g^T x \geq \underline{s}$ wird als Ungleichung mit unsicheren Koeffizienten aufgefasst.

Nebenbedingungen mit unsicheren Koeffizienten

Ist in einer Ungleichung $a^T x \leq b$ über die Koeffizienten nur bekannt, dass $a \in \{\bar{a} + Hu : \|u\| = 1\}$ für gegebenes $H \in S_+^n$ (pos. semidef), und soll x diese Ungleichung für all diese a erfüllen, so muss

$$\max_{\|u\|=1} \bar{a}^T x + u^T Hx = \bar{a}^T x + \|Hx\| \leq b$$

gelten.

Nebenbedingungen mit unsicheren Koeffizienten

Ist in einer Ungleichung $a^T x \leq b$ über die Koeffizienten nur bekannt, dass $a \in \{\bar{a} + Hu : \|u\| = 1\}$ für gegebenes $H \in S_+^n$ (pos. semidef), und soll x diese Ungleichung für all diese a erfüllen, so muss

$$\max_{\|u\|=1} \bar{a}^T x + u^T Hx = \bar{a}^T x + \|Hx\| \leq b$$

gelten. Letztere Ungleichung ist als SOC darstellbar in der Form

$$\begin{aligned} \xi_0 &= b - \bar{a}^T x \\ \xi &= Hx \\ \begin{pmatrix} \xi_0 \\ \xi \end{pmatrix} &\in SOC_n = \left\{ \begin{pmatrix} \xi_0 \\ \xi \end{pmatrix} : \|\xi\| \leq \xi_0, \xi \in \mathbb{R}^n \right\} \end{aligned}$$

Nebenbedingungen mit unsicheren Koeffizienten

Ist in einer Ungleichung $a^T x \leq b$ über die Koeffizienten nur bekannt, dass $a \in \{\bar{a} + Hu : \|u\| = 1\}$ für gegebenes $H \in S_+^n$ (pos. semidef), und soll x diese Ungleichung für all diese a erfüllen, so muss

$$\max_{\|u\|=1} \bar{a}^T x + u^T Hx = \bar{a}^T x + \|Hx\| \leq b$$

gelten. Letztere Ungleichung ist als SOC darstellbar in der Form

$$\begin{aligned} \xi_0 &= b - \bar{a}^T x \\ \xi &= Hx \\ \begin{pmatrix} \xi_0 \\ \xi \end{pmatrix} &\in SOC_n = \left\{ \begin{pmatrix} \xi_0 \\ \xi \end{pmatrix} : \|\xi\| \leq \xi_0, \xi \in \mathbb{R}^n \right\} \end{aligned}$$

Für die probabilistische Interpretation sei g normalverteilt um \bar{g} mit Kovarianzmatrix $G = H^2$ und $g^T x \geq \underline{s}$ soll mit Wahrscheinlichkeit $0 < \eta < 1$ erfüllt sein, dann entspricht $\mathbb{P}(g^T x \geq \underline{s}) \geq \eta$ der Nebenbedingung $-\bar{g}^T x + \Phi^{-1}(\eta)\|Hx\| \leq -\underline{s}$. [Φ ... Normalverteilung]

Inhalt

Innere-Punkte-Verfahren und lineare Optimierung über Kegeln

Innere-Punkte-Verfahren

Lineare Optimierung über Kegeln

Second-Order-Cone Programme

Semidefinite Optimierung

SDP Anwendung: Robuste Stabilität dynamischer Systeme

SDP Anwendung: Entwurf von Experimenten

(SDP Anwendung: Graphenpartition)

(SDP Anwendung: geometrische Einbettungen)

5.4 Semidefinite Optimierung: Positiv semidefinite Matrizen

Eine symmetrische Matrix $A \in S^n := \{A \in \mathbb{R}^{n \times n} : A = A^T\}$ heißt **positiv semidefinit**, falls $v^T A v \geq 0 \quad \forall v \in \mathbb{R}^n$; wir schreiben $A \in S_+^n$ oder $A \succeq 0$.

Sie heißt **positiv definit** ($A \in S_{++}^n, A \succ 0$), falls $v^T A v > 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$.

[Für $A \succeq 0$ ($\succ 0$) und $J \subseteq \{1, \dots, n\}$ ist $A_{J,J} \succeq 0$ ($\succ 0$).]

5.4 Semidefinite Optimierung: Positiv semidefinite Matrizen

Eine symmetrische Matrix $A \in S^n := \{A \in \mathbb{R}^{n \times n} : A = A^T\}$ heißt **positiv semidefinit**, falls $v^T A v \geq 0 \quad \forall v \in \mathbb{R}^n$; wir schreiben $A \in S_+^n$ oder $A \succeq 0$.

Sie heißt **positiv definit** ($A \in S_{++}^n$, $A \succ 0$), falls $v^T A v > 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$.

[Für $A \succeq 0$ ($\succ 0$) und $J \subseteq \{1, \dots, n\}$ ist $A_{J,J} \succeq 0$ ($\succ 0$).]

$\lambda \in \mathbb{R}$ heißt **Eigenwert** und $v \in \mathbb{R}^n \setminus \{0\}$ **Eigenvektor** von A , falls $Av = \lambda v$.

Für jedes $A \in S^n$ gibt es eine **Eigenwertzerlegung** $A = P \Lambda P^T$ mit reellem $\Lambda = \text{Diag}(\lambda_1, \dots, \lambda_n)$ und $P \in \mathbb{R}^{n \times n}$ orthogonal (d.h., $P^T P = I$).

Für $P = [v_1, \dots, v_n]$ ist $A = P \Lambda P^T = \sum_{i=1}^n \lambda_i v_i v_i^T$.

5.4 Semidefinite Optimierung: Positiv semidefinite Matrizen

Eine symmetrische Matrix $A \in S^n := \{A \in \mathbb{R}^{n \times n} : A = A^T\}$ heißt **positiv semidefinit**, falls $v^T A v \geq 0 \quad \forall v \in \mathbb{R}^n$; wir schreiben $A \in S_+^n$ oder $A \succeq 0$.

Sie heißt **positiv definit** ($A \in S_{++}^n$, $A \succ 0$), falls $v^T A v > 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$.

[Für $A \succeq 0$ ($\succ 0$) und $J \subseteq \{1, \dots, n\}$ ist $A_{J,J} \succeq 0$ ($\succ 0$).]

$\lambda \in \mathbb{R}$ heißt **Eigenwert** und $v \in \mathbb{R}^n \setminus \{0\}$ **Eigenvektor** von A , falls $Av = \lambda v$.

Für jedes $A \in S^n$ gibt es eine Eigenwertzerlegung $A = P \Lambda P^T$ mit reellem $\Lambda = \text{Diag}(\lambda_1, \dots, \lambda_n)$ und $P \in \mathbb{R}^{n \times n}$ orthogonal (d.h., $P^T P = I$).

Für $P = [v_1, \dots, v_n]$ ist $A = P \Lambda P^T = \sum_{i=1}^n \lambda_i v_i v_i^T$.

Für $A, B \in S^n$ verwenden wir als inneres Produkt

$$\langle A, B \rangle := \sum_{1 \leq i, j \leq n} A_{ij} B_{ij} \quad [= \text{vec}(A)^T \text{vec}(B)]$$

5.4 Semidefinite Optimierung: Positiv semidefinite Matrizen

Eine symmetrische Matrix $A \in S^n := \{A \in \mathbb{R}^{n \times n} : A = A^T\}$ heißt **positiv semidefinit**, falls $v^T A v \geq 0 \quad \forall v \in \mathbb{R}^n$; wir schreiben $A \in S_+^n$ oder $A \succeq 0$.

Sie heißt **positiv definit** ($A \in S_{++}^n$, $A \succ 0$), falls $v^T A v > 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$.

[Für $A \succeq 0$ ($\succ 0$) und $J \subseteq \{1, \dots, n\}$ ist $A_{J,J} \succeq 0$ ($\succ 0$).]

$\lambda \in \mathbb{R}$ heißt **Eigenwert** und $v \in \mathbb{R}^n \setminus \{0\}$ **Eigenvektor** von A , falls $Av = \lambda v$.

Für jedes $A \in S^n$ gibt es eine Eigenwertzerlegung $A = P \Lambda P^T$ mit reellem $\Lambda = \text{Diag}(\lambda_1, \dots, \lambda_n)$ und $P \in \mathbb{R}^{n \times n}$ orthogonal (d.h., $P^T P = I$).

Für $P = [v_1, \dots, v_n]$ ist $A = P \Lambda P^T = \sum_{i=1}^n \lambda_i v_i v_i^T$.

Für $A, B \in S^n$ verwenden wir als inneres Produkt

$$\langle A, B \rangle := \sum_{1 \leq i, j \leq n} A_{ij} B_{ij} \quad [= \text{vec}(A)^T \text{vec}(B)]$$

Satz

Für $A \in S^n$ sind folgende Bedingungen äquivalent:

- $A \succeq 0$,
- $\lambda_i(A) \geq 0$, $i = 1, \dots, n$, [$\Rightarrow \det(A) \geq 0$]
- $A = C^T C$ für ein $C \in \mathbb{R}^{k \times n}$, [Es gilt: $\text{Rang}(A) = \text{Rang}(C)$]
- $\langle A, B \rangle \geq 0 \quad \forall B \succeq 0$.

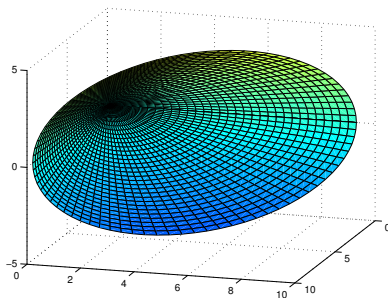
Der Kegel der positiv semidefiniten Matrizen

Die positiv semidefiniten Matrizen S_+^n bilden einen konvexen Kegel, denn für $X, Y \in S_+^n$, $\alpha \geq 0$ ist $\forall v \in \mathbb{R}^n$

$$v^T(\alpha(X + Y))v = \alpha(v^T X v + v^T Y v) \geq 0.$$

Aus $A \in S_+^n \Leftrightarrow \langle A, B \rangle \geq 0 \forall B \succeq 0$ folgt:
 S_+^n ist **selbstdual**, $(S_+^n)^* = S_+^n$.

Bild rechts: $S_+^2 = \left[\begin{array}{cc} x & z \\ z & y \end{array} \right] \succeq 0$.



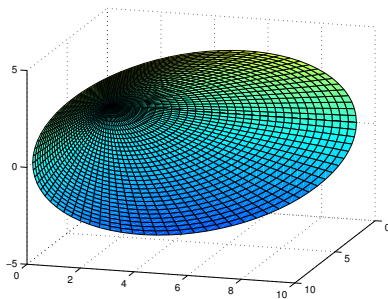
Der Kegel der positiv semidefiniten Matrizen

Die positiv semidefiniten Matrizen S_+^n bilden einen konvexen Kegel, denn für $X, Y \in S_+^n$, $\alpha \geq 0$ ist $\forall v \in \mathbb{R}^n$

$$v^T(\alpha(X + Y))v = \alpha(v^T X v + v^T Y v) \geq 0.$$

Aus $A \in S_+^n \Leftrightarrow \langle A, B \rangle \geq 0 \forall B \succeq 0$ folgt:
 S_+^n ist **selbstdual**, $(S_+^n)^* = S_+^n$.

Bild rechts: $S_+^2 = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0$.



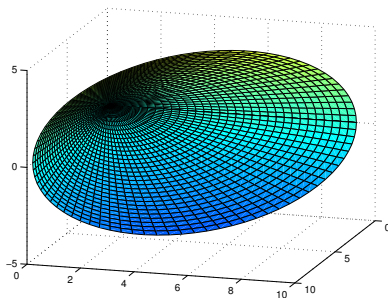
• Ist $B \in \mathbb{R}^{n \times n}$ regulär (=invertierbar), gilt $X \succeq 0 \Leftrightarrow B^T X B \succeq 0$.

Der Kegel der positiv semidefiniten Matrizen

Die positiv semidefiniten Matrizen S_+^n bilden einen konvexen Kegel, denn für $X, Y \in S_+^n$, $\alpha \geq 0$ ist $\forall v \in \mathbb{R}^n$
 $v^T(\alpha(X + Y))v = \alpha(v^T X v + v^T Y v) \geq 0$.

Aus $A \in S_+^n \Leftrightarrow \langle A, B \rangle \geq 0 \forall B \succeq 0$ folgt:
 S_+^n ist **selbstdual**, $(S_+^n)^* = S_+^n$.

Bild rechts: $S_+^2 = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0$.



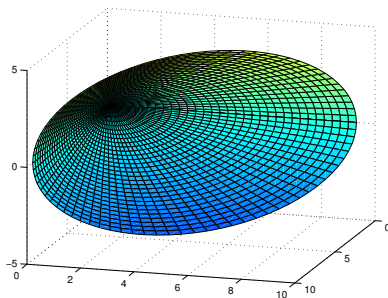
- Ist $B \in \mathbb{R}^{n \times n}$ regulär (=invertierbar), gilt $X \succeq 0 \Leftrightarrow B^T X B \succeq 0$.
- Für $A - B \succeq 0$ schreiben wir auch $A \succeq B$.

Der Kegel der positiv semidefiniten Matrizen

Die positiv semidefiniten Matrizen S_+^n bilden einen konvexen Kegel, denn für $X, Y \in S_+^n$, $\alpha \geq 0$ ist $\forall v \in \mathbb{R}^n$
 $v^T(\alpha(X + Y))v = \alpha(v^T X v + v^T Y v) \geq 0$.

Aus $A \in S_+^n \Leftrightarrow \langle A, B \rangle \geq 0 \forall B \succeq 0$ folgt:
 S_+^n ist **selbstdual**, $(S_+^n)^* = S_+^n$.

Bild rechts: $S_+^2 = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0$.



- Ist $B \in \mathbb{R}^{n \times n}$ regulär (=invertierbar), gilt $X \succeq 0 \Leftrightarrow B^T X B \succeq 0$.
- Für $A - B \succeq 0$ schreiben wir auch $A \succeq B$.

In der Modellierung semidefiniter Programme besonders nützlich:

Satz (Schur-Komplement)

Für $A \in S_{++}^m$, $C \in S_+^n$ und $B \in \mathbb{R}^{m \times n}$ gilt

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0 \quad (\text{bzw. } \succ 0) \iff C \succeq B^T A^{-1} B \quad (\text{bzw. } \succ 0)$$

LP \leftrightarrow Semidefinites Programm

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

$$\begin{aligned} \min \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \mathcal{A}X = b \\ & X \succeq 0 \end{aligned}$$

$$\begin{aligned} x &\in \mathbb{R}_+^n \\ c^T x &= \sum_i c_i x_i \\ Ax &= \begin{pmatrix} a_1^T x \\ \vdots \\ a_m^T x \end{pmatrix} \\ A^T y &= \sum_i a_i y_i \end{aligned}$$

$$\begin{aligned} X &\in S_+^n \\ \langle C, X \rangle &= \sum_{i,j} C_{ij} X_{ij} \\ \mathcal{A}X &= \begin{pmatrix} \langle A_1, X \rangle \\ \vdots \\ \langle A_m, X \rangle \end{pmatrix} \\ \mathcal{A}^T y &= \sum_i A_i y_i \end{aligned}$$

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{aligned}$$

$$\begin{aligned} \max \quad & b^T y \\ \text{s.t.} \quad & \mathcal{A}^T y + Z = C \\ & y \in \mathbb{R}^m, Z \succeq 0 \end{aligned}$$

Semidefinites Programm (SDP) in Normalform

$$\begin{array}{ll} \min & \langle C, X \rangle \\ (P) \quad \text{s.t.} & \mathcal{A}X = b \\ & X \succeq 0 \end{array} \quad \begin{array}{ll} \max & b^T y \\ (D) \quad \text{s.t.} & \mathcal{A}^T y + Z = C \\ & y \in \mathbb{R}^n, Z \succeq 0 \end{array}$$

Ist eines der beiden streng zulässig, gilt $v(P) = v(D)$.

Semidefinites Programm (SDP) in Normalform

$$\begin{array}{ll}
 \min & \langle C, X \rangle \\
 (P) \quad \text{s.t.} & \mathcal{A}X = b \\
 & X \succeq 0
 \end{array}
 \quad
 \begin{array}{ll}
 \max & b^T y \\
 (D) \quad \text{s.t.} & \mathcal{A}^T y + Z = C \\
 & y \in \mathbb{R}^n, Z \succeq 0
 \end{array}$$

Ist eines der beiden streng zulässig, gilt $v(P) = v(D)$.

In der Praxis gibt es oft mehrere $X_i \succeq 0$, für die Theorie reicht eine:

$$X_1 \succeq 0, X_2 \succeq 0, \dots, X_k \succeq 0 \quad \Leftrightarrow \quad \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & X_k \end{bmatrix} \succeq 0.$$

\Rightarrow Semidefinite Optimierung enthält Lineare Optimierung ($X_i \in S_+^1$).

Semidefinites Programm (SDP) in Normalform

$$\begin{array}{ll}
 \min & \langle C, X \rangle \\
 (P) \quad \text{s.t.} & \mathcal{A}X = b \\
 & X \succeq 0
 \end{array}
 \quad
 \begin{array}{ll}
 \max & b^T y \\
 (D) \quad \text{s.t.} & \mathcal{A}^T y + Z = C \\
 & y \in \mathbb{R}^n, Z \succeq 0
 \end{array}$$

Ist eines der beiden streng zulässig, gilt $v(P) = v(D)$.

In der Praxis gibt es oft mehrere $X_i \succeq 0$, für die Theorie reicht eine:

$$X_1 \succeq 0, X_2 \succeq 0, \dots, X_k \succeq 0 \quad \Leftrightarrow \quad \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & X_k \end{bmatrix} \succeq 0.$$

\Rightarrow Semidefinite Optimierung enthält Lineare Optimierung ($X_i \in S_+^1$).

Mit semidefiniter Optimierung sind auch SOC-Bedingungen darstellbar:

$$\begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \succeq_{\mathcal{Q}} 0 \quad \stackrel{x_0 > 0}{\Leftrightarrow} \quad x_0 \geq \frac{1}{x_0} \bar{x}^T / \bar{x} \quad \stackrel{\text{Schur}}{\Leftrightarrow} \quad \begin{bmatrix} x_0 & \bar{x}^T \\ \bar{x} & x_0 I \end{bmatrix} \succeq 0.$$

[für $x_0 = 0$ direkt nachprüfen]

Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$$\Rightarrow x \geq 0, y \geq 0, xy - z^2 \geq 0$$

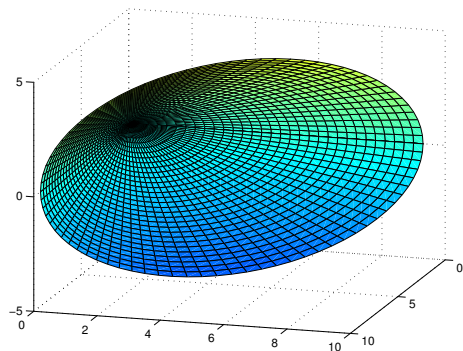


Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$$\Rightarrow x \geq 0, y \geq 0, xy - z^2 \geq 0$$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta = 0 \rightarrow z = 0$$

$$\Leftrightarrow \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}_+^2, \text{ wie LP}$$

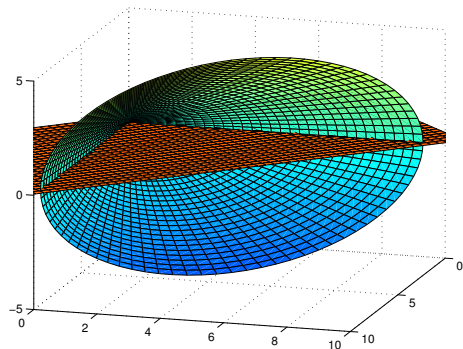


Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$$\Rightarrow x \geq 0, y \geq 0, xy - z^2 \geq 0$$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta = 0 \rightarrow z = 0$$

$$\Leftrightarrow \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}_+^2, \text{ wie LP}$$

$$A \succ 0, \beta > 0$$

→ „Normalfall Ellipse“

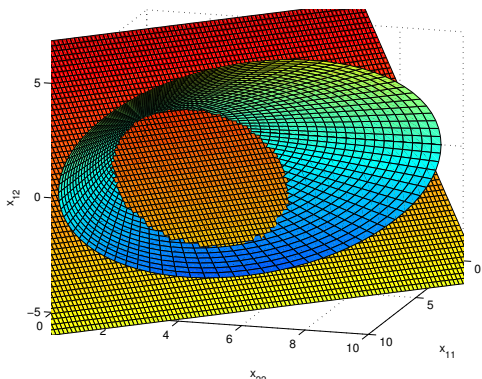


Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$$\Rightarrow x \geq 0, y \geq 0, xy - z^2 \geq 0$$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta = 0 \rightarrow z = 0$$

$$\Leftrightarrow \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}_+^2, \text{ wie LP}$$

$$A \succ 0, \beta > 0$$

→ „Normalfall Ellipse“

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta < 0 \rightarrow z = \frac{1}{2}\beta$$

$$\Leftrightarrow xy \geq \frac{1}{4}\beta^2, \text{ Hyperbel}$$

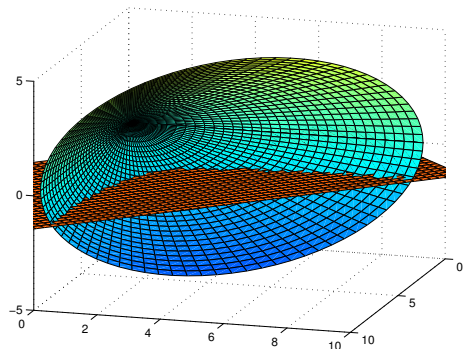


Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$$\Rightarrow x \geq 0, y \geq 0, xy - z^2 \geq 0$$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta = 0 \rightarrow z = 0$$

$$\Leftrightarrow \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}_+^2, \text{ wie LP}$$

$$A \succ 0, \beta > 0$$

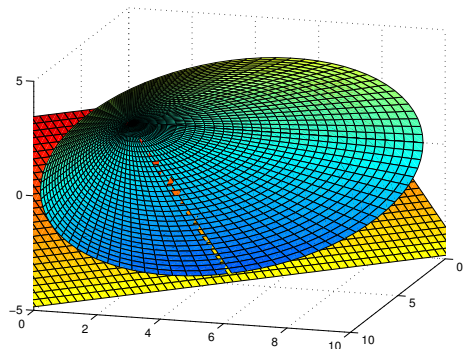
→ „Normalfall Ellipse“

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta < 0 \rightarrow z = \frac{1}{2}\beta$$

$$\Leftrightarrow xy \geq \frac{1}{4}\beta^2, \text{ Hyperbel}$$

$$A = vv^T, \beta = 0 \rightarrow v \text{ EV zu } \lambda_1 = 0$$

Nur Randpunkte, numerisch schwer!



Beispiel mit Dualitätslücke

$$\begin{array}{ll} \min & x_{12} \\ \text{s.t.} & \begin{bmatrix} 0 & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1 + x_{12} \end{bmatrix} \succeq 0 \end{array} \quad \begin{array}{ll} \max & y_1 \\ \text{s.t.} & Z = \begin{bmatrix} -y_2 & \frac{1+y_1}{2} & -y_3 \\ \frac{1+y_1}{2} & 0 & -y_4 \\ -y_3 & -y_4 & -y_1 \end{bmatrix} \succeq 0 \end{array}$$

Entsprechende Matrizen:

$$\begin{array}{l} C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ A_1 = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ A_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$

$$\begin{array}{l} \max \quad \langle C, X \rangle \\ \langle A_1, X \rangle = 1 \\ \langle A_2, X \rangle = 0 \\ \langle A_3, X \rangle = 0 \\ \langle A_4, X \rangle = 0 \\ X \succeq 0 \end{array}$$

$$x_{11} = 0 \Rightarrow x_{12} = 0, \quad \text{primale Optimallösung ist } 0.$$

$$z_{22} = 0 \Rightarrow \frac{1+y_1}{2} = 0, \quad \text{duale Optimallösung ist } -1.$$

Problem: Primales Problem instabil

$$\begin{array}{ll} \min & x_{12} \\ \text{s.t.} & \begin{bmatrix} \varepsilon & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1 + x_{12} \end{bmatrix} \succeq 0 \end{array} \quad \begin{array}{ll} \max & y_1 + \varepsilon y_2 \\ \text{s.t.} & Z = \begin{bmatrix} -y_2 & \frac{1+y_1}{2} & -y_3 \\ \frac{1+y_1}{2} & 0 & -y_4 \\ -y_3 & -y_4 & -y_1 \end{bmatrix} \succeq 0 \end{array}$$

Entsprechende Matrizen:

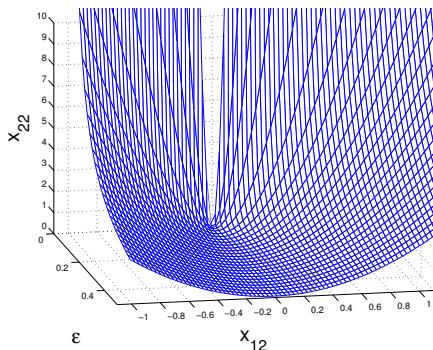
$$\begin{array}{l} C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ A_1 = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\ A_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$

$$\begin{array}{l} \max \quad \langle C, X \rangle \\ \langle A_1, X \rangle = 1 \\ \langle A_2, X \rangle = \varepsilon \\ \langle A_3, X \rangle = 0 \\ \langle A_4, X \rangle = 0 \\ X \succeq 0 \end{array}$$

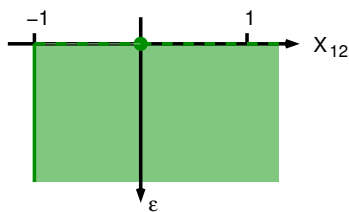
$$x_{33} \geq 0 \Rightarrow x_{12} \geq -1, \quad x_{22} \geq \frac{x_{12}^2}{\varepsilon}, \quad \text{primale Optimallösung ist } -1.$$

$$z_{22} = 0 \Rightarrow \frac{1+y_1}{2} = 0, \quad y_2 = 0, \quad \text{duale Optimallösung ist } -1.$$

$$\begin{bmatrix} \varepsilon & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1 + x_{12} \end{bmatrix} \succeq 0 \Leftrightarrow \begin{cases} \varepsilon > 0 & x_{12} \geq -1 & x_{22} \geq \frac{x_{12}^2}{\varepsilon} \\ \varepsilon = 0 & x_{12} = 0 & x_{22} \geq 0 \end{cases}$$



Projektion auf (ε, x_{12}) -Ebene:



Für $\varepsilon > 0$ ist $x_{12} \in [-1, -\infty)$,
für $\varepsilon = 0$ nur noch $x_{12} \in \{0\}$
zulässig!

Mathem. Grund: die Menge $\left\{ \begin{bmatrix} \langle C, X \rangle \\ AX \end{bmatrix} : X \succeq 0 \right\}$ ist nicht abgeschlossen.

SDP und Eigenwertoptimierung

Für $A \in S^n$ bezeichne $\lambda_{\min}(A) := \lambda_1(A) \leq \dots \leq \lambda_n(A) =: \lambda_{\max}(A)$. Es gilt $\lambda_i(A + y_0 I) = \lambda_i(A) + y_0$ für $i = 1, \dots, n$ und $y_0 \in \mathbb{R}$.

SDP und Eigenwertoptimierung

Für $A \in S^n$ bezeichne $\lambda_{\min}(A) := \lambda_1(A) \leq \dots \leq \lambda_n(A) =: \lambda_{\max}(A)$. Es gilt $\lambda_i(A + y_0 I) = \lambda_i(A) + y_0$ für $i = 1, \dots, n$ und $y_0 \in \mathbb{R}$.

In der Optimalsteuerung ist die Stabilität eines Systems gewährleistet, wenn man für die von Steuerparametern $y \in \mathbb{R}^m$ abhängige Systemmatrix $A(y)$ nachweisen kann, dass $\lambda_{\max}(A(y)) < 0$.

Ist $A(y)$ affin, etwa $A(y) := C - \sum_{i=1}^m y_i A_i$ mit $C, A_i \in S^n$, führt das auf

$$\min_{y \in \mathbb{R}^m} \lambda_{\max}(C - \mathcal{A}^T y)$$

SDP und Eigenwertoptimierung

Für $A \in S^n$ bezeichne $\lambda_{\min}(A) := \lambda_1(A) \leq \dots \leq \lambda_n(A) =: \lambda_{\max}(A)$. Es gilt $\lambda_i(A + y_0 I) = \lambda_i(A) + y_0$ für $i = 1, \dots, n$ und $y_0 \in \mathbb{R}$.

In der Optimalsteuerung ist die Stabilität eines Systems gewährleistet, wenn man für die von Steuerparametern $y \in \mathbb{R}^m$ abhängige Systemmatrix $A(y)$ nachweisen kann, dass $\lambda_{\max}(A(y)) < 0$.

Ist $A(y)$ affin, etwa $A(y) := C - \sum_{i=1}^m y_i A_i$ mit $C, A_i \in S^n$, führt das auf

$$\min_{y \in \mathbb{R}^m} \lambda_{\max}(C - \mathcal{A}^T y)$$

Zur Modellierung als SDP: $\lambda_{\max}(A) = -\lambda_{\min}(-A)$ und

$$y_0 \geq \lambda_{\max}(C - \mathcal{A}^T y) \Leftrightarrow y_0 + \lambda_{\min}(\mathcal{A}^T y - C) \geq 0 \Leftrightarrow \lambda_{\min}(y_0 I + \mathcal{A}^T y - C) \geq 0$$

Wegen $Z \succeq 0 \Leftrightarrow \lambda_{\min}(Z) \geq 0$ gilt daher

$$\min_{y \in \mathbb{R}^m} \lambda_{\max}(C - \mathcal{A}^T y) \Leftrightarrow \begin{array}{ll} \min & y_0 \\ \text{s.t.} & Z = y_0 I + \mathcal{A}^T y - C \\ & y \in \mathbb{R}^m, Z \succeq 0 \end{array}$$

Lineare Matrix Ungleichungen (LMI)

Eine Bedingung der Form

$$y_1 A_1 + y_2 A_2 + \cdots + y_m A_m \preceq C$$

mit $A_i, C \in S^n$ heißt **lineare Matrix Ungleichung** (Linear Matrix Inequality).

Zulässige $y \in \mathbb{R}^m$ sind SDP-darstellbar, $\{y \in \mathbb{R}^m : \mathcal{A}^T y + Z = C, Z \succeq 0\}$.

Lineare Matrix Ungleichungen (LMI)

Eine Bedingung der Form

$$y_1 A_1 + y_2 A_2 + \cdots + y_m A_m \preceq C$$

mit $A_i, C \in S^n$ heißt **lineare Matrix Ungleichung** (Linear Matrix Inequality).

Zulässige $y \in \mathbb{R}^m$ sind SDP-darstellbar, $\{y \in \mathbb{R}^m : \mathcal{A}^T y + Z = C, Z \succeq 0\}$.

Bsp: Die **Lyapunov Ungleichung** fordert für festes $P = [p_1, \dots, p_n] \in \mathbb{R}^{n \times n}$

$$P^T X + X P \prec 0, \quad X \succ 0.$$

In LMI-Darstellung wäre $y = [x_{11}, x_{12}, \dots, x_{1n}, x_{22}, x_{23}, \dots, x_{nn}]^T$, aber es wäre umständlich/sinnlos, die A_i für diese Ungleichungen anzugeben, die Struktur lässt sich im SDP besser direkt nutzen.

Lineare Matrix Ungleichungen (LMI)

Eine Bedingung der Form

$$y_1 A_1 + y_2 A_2 + \dots + y_m A_m \preceq C$$

mit $A_i, C \in S^n$ heißt **lineare Matrix Ungleichung** (Linear Matrix Inequality).

Zulässige $y \in \mathbb{R}^m$ sind SDP-darstellbar, $\{y \in \mathbb{R}^m : \mathcal{A}^T y + Z = C, Z \succeq 0\}$.

Bsp: Die **Lyapunov Ungleichung** fordert für festes $P = [p_1, \dots, p_n] \in \mathbb{R}^{n \times n}$

$$P^T X + X P \prec 0, \quad X \succ 0.$$

In LMI-Darstellung wäre $y = [x_{11}, x_{12}, \dots, x_{1n}, x_{22}, x_{23}, \dots, x_{nn}]^T$, aber es wäre umständlich/sinnlos, die A_i für diese Ungleichungen anzugeben, die Struktur lässt sich im SDP besser direkt nutzen.

Um LMIs zu erkennen, reicht es festzustellen, dass die Matrizen linear von den jeweiligen Variablen abhängen:

Die Matrix-Multiplikation $P^T X$ (bzw. XP) ist linear in X .

Wie erzwingt man positive Definitheit?

Die semidefinite Barrierefunktion – $\log \det X$

Wegen $\det X = \prod_{k=1}^n \lambda_k(X)$ ist

$$-\log \det X = -\log \prod_{k=1}^n \lambda_k(X) = -\sum_{k=1}^n \log \lambda_k(X)$$

eine Barrierefunktion für $X \succeq 0$ ($\Leftrightarrow \lambda_i(X) \geq 0$) und nur für $X \succ 0$ ($\Leftrightarrow \lambda_i(X) > 0$) erklärt. [vergl. Barriere – $\sum \log x_i$ für $x \geq 0$ in LP]

Innere-Punkte-Verfahren für SDP nutzen diese Barrierefunktion. [wie LP!]

Die semidefinite Barrierefunktion – $\log \det X$

Wegen $\det X = \prod_{k=1}^n \lambda_k(X)$ ist

$$-\log \det X = -\log \prod_{k=1}^n \lambda_k(X) = -\sum_{k=1}^n \log \lambda_k(X)$$

eine Barrierefunktion für $X \succeq 0$ ($\Leftrightarrow \lambda_i(X) \geq 0$) und nur für $X \succ 0$ ($\Leftrightarrow \lambda_i(X) > 0$) erklärt. [vergl. Barriere – $\sum \log x_i$ für $x \geq 0$ in LP]

Innere-Punkte-Verfahren für SDP nutzen diese Barrierefunktion. [wie LP!]

$-\log \det X$ wird auch genutzt, um streng zulässige Lösungen zu finden, z.B.

$$\begin{aligned} \min \quad & -\log \det X - \log \det Z \\ \text{s.t.} \quad & Z = P^T X + P X \\ & X \succ 0, Z \succ 0 \end{aligned}$$

→ Innere-Punkte-Verfahren für konstanten Barriereparameter $\mu = 1$.

Das Maximieren der Determinante hat zahlreiche Anwendungen, insbesondere im Entwurf von Experimenten (*experiment design*)

Anwendungen der Semidefiniten Optimierung

- Optimalsteuerung und Kontrolltheorie
- Signalverarbeitung
- Kombinatorische Optimierung
- Globale Optimierung über Polynomen
- Robuster Entwurf von Stabkonstruktionen (truss topology design)
- Entwurf von Materialien (free material design)
- Robuste Optimierung
- Momenten-Probleme in der Wahrscheinlichkeitstheorie
- Entwurf von Experimenten in der Statistik
- Eigenwert-Optimierung
- Optimierung (trust-region Bestimmung, quadratische Relaxationen)

5.4.1 Robuste Stabilität dynamischer Systeme

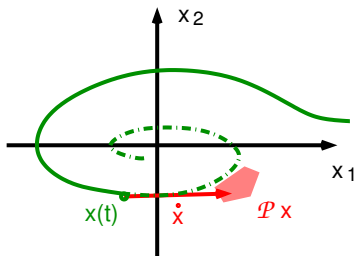
In einem (homogenen linearen) dynamischen System mit unsicheren Daten,

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \text{conv}\{P_1, \dots, P_k\} \subset \mathbb{R}^{n \times n},$$

- beschreibt
- $x(t)$... Zustand des Systems zur Zeit t .
 - $\dot{x} := \frac{d}{dt}x(t)$... (infinitesimale) Veränderung von $x(\cdot)$
 - $P(t)$... unsichere Übergangsmatrix zur Zeit t .

(DS) heißt **stabil**, wenn $x(t) \rightarrow 0$ für $t \rightarrow \infty$ und beliebige $P(t) \in \mathcal{P}$.

[In der Regelungstechnik würde \mathcal{P} die möglichen Auswirkungen der Regelung/Steuerung umfassen. Man möchte wissen, ob diese auch bei unsauberer Realisierung in der Praxis den Zweck erfüllt.]



5.4.1 Robuste Stabilität dynamischer Systeme

In einem (homogenen linearen) dynamischen System mit unsicheren Daten,

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \text{conv}\{P_1, \dots, P_k\} \subset \mathbb{R}^{n \times n},$$

beschreibt

- $x(t)$... Zustand des Systems zur Zeit t .
- $\dot{x} := \frac{d}{dt}x(t)$... (infinitesimale) Veränderung von $x(\cdot)$
- $P(t)$... unsichere Übergangsmatrix zur Zeit t .

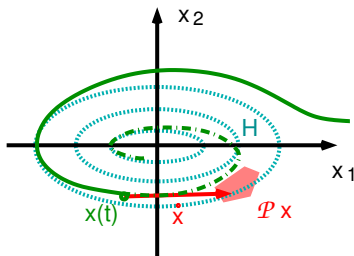
(DS) heißt **stabil**, wenn $x(t) \rightarrow 0$ für $t \rightarrow \infty$ und beliebige $P(t) \in \mathcal{P}$.

Hinreichend: Es gibt eine Norm

$$\|x\|_H := \sqrt{x^T H x} \quad \text{mit } H \succ 0$$

mit $\frac{d}{dt} \|x(t)\|_H^2 < 0$ auf allen Trajektorien

(das System heißt dann **quadratisch stabil**,
 $x^T H x$ **quadratische Lyapunov Funktion**).



Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \text{conv}\{P_1, \dots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\frac{d}{dt} \|x(t)\|_H^2 < 0$.

Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \text{conv}\{P_1, \dots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\frac{d}{dt} \|x(t)\|_H^2 < 0$.

$$\frac{d}{dt} \|x(t)\|_H^2 = \frac{d}{dt} x^T H x = \dot{x}^T H x + x^T H \dot{x} = x^T (P(t)^T H + H P(t)) x$$

Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \text{conv}\{P_1, \dots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\frac{d}{dt} \|x(t)\|_H^2 < 0$.

$$\frac{d}{dt} \|x(t)\|_H^2 = \frac{d}{dt} x^T H x = \dot{x}^T H x + x^T H \dot{x} = x^T (P(t)^T H + H P(t)) x$$

Falls $A := P^T H + H P \prec 0$ (negativ definit), gilt $v^T A v < 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$.

Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \text{conv}\{P_1, \dots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\frac{d}{dt} \|x(t)\|_H^2 < 0$.

$$\frac{d}{dt} \|x(t)\|_H^2 = \frac{d}{dt} x^T H x = \dot{x}^T H x + x^T H \dot{x} = x^T (P(t)^T H + H P(t)) x$$

Falls $A := P^T H + H P \prec 0$ (negativ definit), gilt $v^T A v < 0 \forall v \in \mathbb{R}^n \setminus \{0\}$.
 \Rightarrow Das System ist quadratisch stabil, falls

$$H \succ 0, \quad P_i^T H + H P_i \prec 0 \quad \text{für } i = 1, \dots, k$$

zulässig lösbar ist, denn für so ein H erfüllt auch jede Konvexkombination $P \in \mathcal{P}$ die Bedingung $P^T H + H P \prec 0$.

Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \text{conv}\{P_1, \dots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\frac{d}{dt} \|x(t)\|_H^2 < 0$.

$$\frac{d}{dt} \|x(t)\|_H^2 = \frac{d}{dt} x^T H x = \dot{x}^T H x + x^T H \dot{x} = x^T (P(t)^T H + H P(t)) x$$

Falls $A := P^T H + H P \prec 0$ (negativ definit), gilt $v^T A v < 0 \forall v \in \mathbb{R}^n \setminus \{0\}$.
 \Rightarrow Das System ist quadratisch stabil, falls

$$H \succ 0, \quad P_i^T H + H P_i \prec 0 \quad \text{für } i = 1, \dots, k$$

zulässig lösbar ist, denn für so ein H erfüllt auch jede Konvexkombination $P \in \mathcal{P}$ die Bedingung $P^T H + H P \prec 0$.

Suche H über Determinanten-Maximierung oder Eigenwert-Optimierung:

$$\max \lambda \quad \text{s.t. } H \succeq \lambda I, \quad P_i^T H + H P_i \preceq -\lambda I \quad \text{für } i = 1, \dots, k.$$

5.4.2 Entwurf von Experimenten

Um die Werte eines Parametervektors $\xi \in \mathbb{R}^p$ zu schätzen, stehen $\mathcal{R} = \{r_i \in \mathbb{R}^p : i = 1, \dots, n\}$ mögliche Experimente zur Verfügung. Experiment i liefert pro Durchführung einen Messwert $r_i^T \xi + \rho_i$ mit unabhängig $(\mu = 0, \sigma^2 = 1)$ -normalverteiltem Messfehler ρ_i .

5.4.2 Entwurf von Experimenten

Um die Werte eines Parametervektors $\xi \in \mathbb{R}^p$ zu schätzen, stehen $\mathcal{R} = \{r_i \in \mathbb{R}^p : i = 1, \dots, n\}$ mögliche Experimente zur Verfügung. Experiment i liefert pro Durchführung einen Messwert $r_i^T \xi + \rho_i$ mit unabhängig ($\mu = 0, \sigma^2 = 1$)-normalverteiltem Messfehler ρ_i .

Werden m Experimente $a_j \in \mathcal{R}$ (Wiederholungen sind erlaubt) mit Ergebnissen $\eta_j = a_j^T \xi + \rho_j$ durchgeführt, ergibt der Maximum-Likelihood-Schätzer bei $\text{Rang}[a_1, \dots, a_m] = n$ ein geschätztes

$$\hat{\xi} = G \sum_{j=1}^m \eta_j a_j \quad \text{mit} \quad G = \left(\sum_{j=1}^m a_j a_j^T \right)^{-1},$$

dessen Fehlerverteilung Erwartungswert 0 und Kovarianzmatrix G hat.

5.4.2 Entwurf von Experimenten

Um die Werte eines Parametervektors $\xi \in \mathbb{R}^p$ zu schätzen, stehen $\mathcal{R} = \{r_i \in \mathbb{R}^p : i = 1, \dots, n\}$ mögliche Experimente zur Verfügung. Experiment i liefert pro Durchführung einen Messwert $r_i^T \xi + \rho_i$ mit unabhängig ($\mu = 0, \sigma^2 = 1$)-normalverteiltem Messfehler ρ_i .

Werden m Experimente $a_j \in \mathcal{R}$ (Wiederholungen sind erlaubt) mit Ergebnissen $\eta_j = a_j^T \xi + \rho_j$ durchgeführt, ergibt der Maximum-Likelihood-Schätzer bei $\text{Rang}[a_1, \dots, a_m] = n$ ein geschätztes

$$\hat{\xi} = G \sum_{j=1}^m \eta_j a_j \quad \text{mit} \quad G = \left(\sum_{j=1}^m a_j a_j^T \right)^{-1},$$

dessen Fehlerverteilung Erwartungswert 0 und Kovarianzmatrix G hat.

Sind G und G' zwei Kovarianzmatrizen dieser Art und gilt $G \preceq G'$, dann ist die zu G gehörende Experimentfolge besser, weil die Varianz des Schätzfehlers kleiner ist.

→ Finde die bzgl. \preceq minimalen Elemente von

$$\left\{ G = \left(\sum_{i=1}^n m_i r_i r_i^T \right)^{-1} : m_i \in \mathbb{N}_0, \sum_i m_i = m \right\}.$$

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^n \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^n \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Es gibt mehrere Ansätze, ein bzgl. \preceq minimales G zu finden. Interpretiere G dazu als ein „Konfidenzellipsoid“ mit Halbachsenlängen $\lambda_j(G)$,

$$\mathcal{E} = \{ \zeta : (\zeta - \hat{\xi})^T G^{-1} (\zeta - \hat{\xi}) \leq \beta \}.$$

D -optimales Design: Minimiere das Volumen des Konfidenzellipsoids.

E -optimales Design: Minimiere die längste Halbachse.

A -optimales Design: Minimiere die Summe der Halbachsen.

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^n \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Es gibt mehrere Ansätze, ein bzgl. \preceq minimales G zu finden. Interpretiere G dazu als ein „Konfidenzellipsoid“ mit Halbachsenlängen $\lambda_j(G)$,

$$\mathcal{E} = \{ \zeta : (\zeta - \hat{\xi})^T G^{-1} (\zeta - \hat{\xi}) \leq \beta \}.$$

D -optimales Design: Minimiere das Volumen des Konfidenzellipsoids.

E -optimales Design: Minimiere die längste Halbachse.

A -optimales Design: Minimiere die Summe der Halbachsen.

D -optimales Design. Das Volumen ist zu $\det G = \prod \lambda_j(G)$ proportional. Wegen $\det(G^{-1}) = \det(G)^{-1} \Leftrightarrow$ maximiere die Determinante von G^{-1} ,

$$\begin{aligned} \min \quad & -\log \det X \\ \text{s.t.} \quad & X = \sum_{i=1}^n \alpha_i r_i r_i^T \\ & \mathbf{1}^T \alpha = 1 \\ & \alpha \geq 0, [X \succ 0] \end{aligned}$$

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^n \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Es gibt mehrere Ansätze, ein bzgl. \preceq minimales G zu finden. Interpretiere G dazu als ein „Konfidenzellipsoid“ mit Halbachsenlängen $\lambda_j(G)$,

$$\mathcal{E} = \{ \zeta : (\zeta - \hat{\xi})^T G^{-1} (\zeta - \hat{\xi}) \leq \beta \}.$$

D -optimales Design: Minimiere das Volumen des Konfidenzellipsoids.

E -optimales Design: Minimiere die längste Halbachse.

A -optimales Design: Minimiere die Summe der Halbachsen.

E -optimales Design. Die längste Halbachse ist $\lambda_{\max}(G)$.

Wegen $\lambda_{\min}(G^{-1}) = \lambda_{\max}(G)^{-1} \Leftrightarrow$ maximiere $\lambda_{\min}(G^{-1})$,

$$\begin{aligned} \max \quad & -\lambda \\ \text{s.t.} \quad & \sum_{j=1}^n \alpha_j r_j r_j^T \succeq \lambda I \\ & \mathbf{1}^T \alpha = 1 \\ & \alpha \geq 0, \lambda \in \mathbb{R} \end{aligned}$$

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^n \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Es gibt mehrere Ansätze, ein bzgl. \preceq minimales G zu finden. Interpretiere G dazu als ein „Konfidenzellipsoid“ mit Halbachsenlängen $\lambda_j(G)$,

$$\mathcal{E} = \{ \zeta : (\zeta - \hat{\xi})^T G^{-1} (\zeta - \hat{\xi}) \leq \beta \}.$$

D -optimales Design: Minimiere das Volumen des Konfidenzellipsoids.

E -optimales Design: Minimiere die längste Halbachse.

A -optimales Design: Minimiere die Summe der Halbachsen.

A -optimales Design. $\sum_{j=1}^p \lambda_j(G) = \sum_{j=1}^p G_{jj} = \sum_{j=1}^p e_j^T G e_j$.

Für jedes j ist die Unglg. $u_j \succeq e_j^T G e_j$ über Schur-Komplement darstellbar:

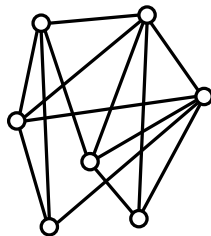
$$\begin{aligned} \min \quad & \mathbf{1}^T u \\ \text{s.t.} \quad & \begin{bmatrix} \sum_{i=1}^n \alpha_i r_i r_i^T & e_j \\ e_j^T & u_j \end{bmatrix} \succeq 0, \quad j = 1, \dots, p \\ & \mathbf{1}^T \alpha = 1, \alpha \geq 0, u \in \mathbb{R}^p \end{aligned}$$

(5.4.3 Graphenpartition: Max-Cut)

Gegeben: Graph $G = (V, E)$, $V = \{1, \dots, n\}$,
 $E \subseteq \{ij : i, j \in V, i < j\}$, Kantengewichte a_{ij}

Gesucht: $S \subset V$ mit gewichtsmaximalem
 Schnitt $\delta(S) := \{ij \in E : i \in S, j \in V \setminus S\}$

$$(MC) \quad \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} \quad [NP\text{-vollst.}]$$

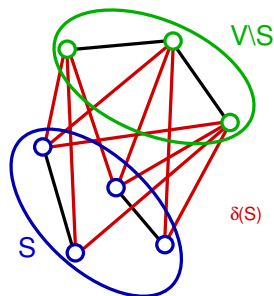


(5.4.3 Graphenpartition: Max-Cut)

Gegeben: Graph $G = (V, E)$, $V = \{1, \dots, n\}$,
 $E \subseteq \{ij : i, j \in V, i < j\}$, Kantengewichte a_{ij}

Gesucht: $S \subset V$ mit gewichtsmaximalem
 Schnitt $\delta(S) := \{ij \in E : i \in S, j \in V \setminus S\}$

$$(MC) \quad \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} \quad [NP\text{-vollst.}]$$



(5.4.3 Graphenpartition: Max-Cut)

Gegeben: Graph $G = (V, E)$, $V = \{1, \dots, n\}$,
 $E \subseteq \{ij : i, j \in V, i < j\}$, Kantengewichte a_{ij}

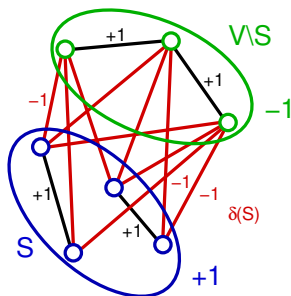
Gesucht: $S \subset V$ mit gewichtsmaximalem
 Schnitt $\delta(S) := \{ij \in E : i \in S, j \in V \setminus S\}$

$$(MC) \quad \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} \quad [NP\text{-vollst.}]$$

Modellierung: Repräsentiere die Partition durch

$$x \in \{-1, 1\}^n \quad \text{mit} \quad x_i = \begin{cases} 1 & i \in S \\ -1 & i \in V \setminus S \end{cases}$$

$$\text{Dann ist } x_i x_j = \begin{cases} -1 & ij \in \delta(S) \\ 1 & \text{sonst} \end{cases}, \quad \text{bzw.} \quad \frac{1 - x_i x_j}{2} = \begin{cases} 1 & ij \in \delta(S) \\ 0 & \text{sonst.} \end{cases}$$



(5.4.3 Graphenpartition: Max-Cut)

Gegeben: Graph $G = (V, E)$, $V = \{1, \dots, n\}$,
 $E \subseteq \{ij : i, j \in V, i < j\}$, Kantengewichte a_{ij}

Gesucht: $S \subset V$ mit gewichtsmaximalem
 Schnitt $\delta(S) := \{ij \in E : i \in S, j \in V \setminus S\}$

$$(MC) \quad \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} \quad [NP\text{-vollst.}]$$

Modellierung: Repräsentiere die Partition durch

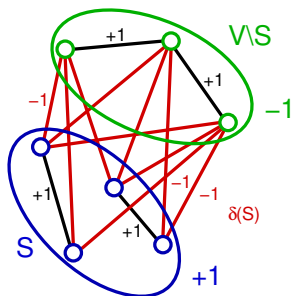
$$x \in \{-1, 1\}^n \quad \text{mit} \quad x_i = \begin{cases} 1 & i \in S \\ -1 & i \in V \setminus S \end{cases}$$

$$\text{Dann ist } x_i x_j = \begin{cases} -1 & ij \in \delta(S) \\ 1 & \text{sonst} \end{cases}, \quad \text{bzw.} \quad \frac{1 - x_i x_j}{2} = \begin{cases} 1 & ij \in \delta(S) \\ 0 & \text{sonst.} \end{cases}$$

$$\max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} = \max_{x \in \{-1, 1\}^n} \sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2} \quad \rightarrow \quad \max_{x \in \{-1, 1\}^n} x^T C x$$

$[C \in S^n: C_{ii} = \frac{1}{4} \sum_{j: ij \in E} a_{ij}$ (für $i \in V$), $C_{ij} = -\frac{1}{4} a_{ij}$ (für $ij \in E$), 0 sonst]

Äquivalent zu quadratischer 0-1 Optimierung!



Semidefinite Max-Cut Relaxation

Beachte: $x^T C x = \langle Cx, x \rangle = \langle C, xx^T \rangle$

Eigenschaften von $xx^T = [x_i x_j]$ für $x \in \{-1, 1\}^n$:

- $x_i^2 = 1 \Rightarrow \text{diag}(xx^T) = \mathbf{1}$
- xx^T ist positiv semidefinit, $xx^T \succeq 0$
- $\text{Rang}(xx^T) = 1$

Semidefinite Max-Cut Relaxation

Beachte: $x^T C x = \langle C x, x \rangle = \langle C, x x^T \rangle$

Eigenschaften von $x x^T = [x_i x_j]$ für $x \in \{-1, 1\}^n$:

- $x_i^2 = 1 \Rightarrow \text{diag}(x x^T) = \mathbf{1}$
- $x x^T$ ist positiv semidefinit, $x x^T \succeq 0$
- $\text{Rang}(x x^T) = 1$

Relaxationsidee: Ersetze $x x^T$ durch eine positiv semidefinite Matrix X .

$$\max_{x \in \{-1, 1\}^n} x^T C x \leq$$

$$\begin{array}{ll} \max & \langle C, X \rangle \\ \text{s.t.} & \text{diag}(X) = \mathbf{1} \\ & X \succeq 0 \\ & [\text{Rang}(X) = 1] \end{array}$$

[mit Rang 1 \Leftrightarrow (MC), NP-vollst.]

Semidefinite Max-Cut Relaxation

Beachte: $x^T C x = \langle C x, x \rangle = \langle C, x x^T \rangle$

Eigenschaften von $x x^T = [x_i x_j]$ für $x \in \{-1, 1\}^n$:

- $x_i^2 = 1 \Rightarrow \text{diag}(x x^T) = \mathbf{1}$
- $x x^T$ ist positiv semidefinit, $x x^T \succeq 0$
- $\text{Rang}(x x^T) = 1$

Relaxationsidee: Ersetze $x x^T$ durch eine positiv semidefinite Matrix X .

$$\max_{x \in \{-1, 1\}^n} x^T C x \leq$$

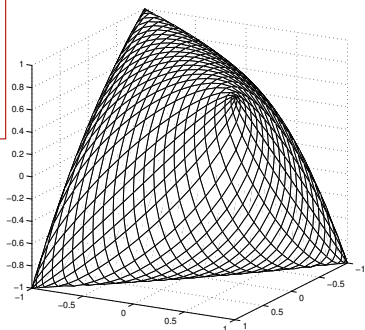
$$\begin{array}{ll} \max & \langle C, X \rangle \\ \text{s.t.} & \text{diag}(X) = \mathbf{1} \\ & X \succeq 0 \\ & [\text{Rang}(X) = 1] \end{array}$$

[mit Rang 1 \Leftrightarrow (MC), NP-vollst.]

Illustration für $n = 3$:

Rand wird beschrieben von

$$\det \begin{bmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{bmatrix} = 0.$$



Geometrische Interpretation und Rundung

Interpretiere OL X^* als **Gram-Matrix** $X^* = R^T R$ von $R = [r_1, \dots, r_n] \in \mathbb{R}^{k \times n}$,

$$X_{ij}^* = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2)$$

Geometrische Interpretation und Rundung

Interpretiere OL X^* als **Gram-Matrix** $X^* = R^T R$ von $R = [r_1, \dots, r_n] \in \mathbb{R}^{k \times n}$,

$$X_{ij}^* = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

$\|r_i\| = 1$ wegen $\text{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X_{ij}^*}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Geometrische Interpretation und Rundung

Interpretiere OL X^* als **Gram-Matrix** $X^* = R^T R$ von $R = [r_1, \dots, r_n] \in \mathbb{R}^{k \times n}$,

$$X_{ij}^* = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

$\|r_i\| = 1$ wegen $\text{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X_{ij}^*}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Ist $r_i^T r_j$ nahe bei -1 (großer Winkel $\angle(r_1, r_2)$), sollte man i und j trennen.

Geht das für alle gleichzeitig?

Geometrische Interpretation und Rundung

Interpretiere OL X^* als **Gram-Matrix** $X^* = R^T R$ von $R = [r_1, \dots, r_n] \in \mathbb{R}^{k \times n}$,

$$X_{ij}^* = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

$\|r_i\| = 1$ wegen $\text{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

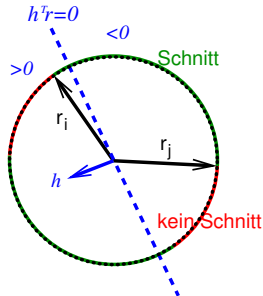
$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X_{ij}^*}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Ist $r_i^T r_j$ nahe bei -1 (großer Winkel $\angle(r_1, r_2)$), sollte man i und j trennen.

Geht das für alle gleichzeitig? „**Zufälliges Runden mit Hyperebenen**“

Runde zu $x \in \{-1, 1\}^n$ über einen normalv. Zufallsvektor $h \in \mathbb{R}^k$ durch

$$x_i = \begin{cases} 1 & h^T r_i \geq 0 \\ -1 & \text{sonst} \end{cases} \rightarrow H = \sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2}$$



Geometrische Interpretation und Rundung

Interpretiere OL X^* als **Gram-Matrix** $X^* = R^T R$ von $R = [r_1, \dots, r_n] \in \mathbb{R}^{k \times n}$,

$$X_{ij}^* = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

$\|r_i\| = 1$ wegen $\text{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X_{ij}^*}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Ist $r_i^T r_j$ nahe bei -1 (großer Winkel $\angle(r_1, r_2)$), sollte man i und j trennen.

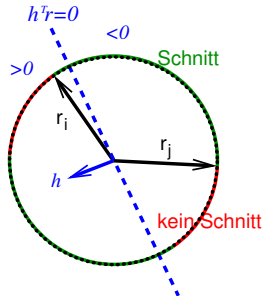
Geht das für alle gleichzeitig? **„Zufälliges Runden mit Hyperebenen“**

Runde zu $x \in \{-1, 1\}^n$ über einen normalv. Zufallsvektor $h \in \mathbb{R}^k$ durch

$$x_i = \begin{cases} 1 & h^T r_i \geq 0 \\ -1 & \text{sonst} \end{cases} \rightarrow H = \sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2}$$

$\mathbb{E}(H)$ errechnet sich pro $ij \in E$ durch Projektion von h auf die $\{r_i, r_j\}$ -Ebene:

$$\mathbb{P}(x_i x_j = -1) = \arccos(r_i^T r_j) / \pi$$



Geometrische Interpretation und Rundung

Interpretiere OL X^* als **Gram-Matrix** $X^* = R^T R$ von $R = [r_1, \dots, r_n] \in \mathbb{R}^{k \times n}$,

$$X_{ij}^* = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

$\|r_i\| = 1$ wegen $\text{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X_{ij}^*}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Ist $r_i^T r_j$ nahe bei -1 (großer Winkel $\angle(r_1, r_2)$), sollte man i und j trennen.

Geht das für alle gleichzeitig? **„Zufälliges Runden mit Hyperebenen“**

Runde zu $x \in \{-1, 1\}^n$ über einen normalv. Zufallsvektor $h \in \mathbb{R}^k$ durch

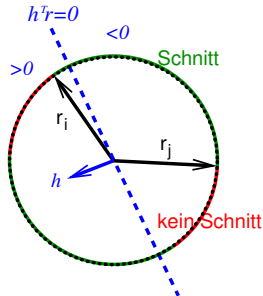
$$x_i = \begin{cases} 1 & h^T r_i \geq 0 \\ -1 & \text{sonst} \end{cases} \rightarrow H = \sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2}$$

$\mathbb{E}(H)$ errechnet sich pro $ij \in E$ durch Projektion von h auf die $\{r_i, r_j\}$ -Ebene:

$$\mathbb{P}(x_i x_j = -1) = \arccos(r_i^T r_j) / \pi$$

Es gilt $\arccos(t) / \pi \geq 0.878 \frac{1-t}{2}$

und wenn $a_{ij} \geq 0 \Rightarrow \mathbb{E}(H) \geq 0.878 \langle C, X_* \rangle$



„0.878 Approximationsalgorithmus“ von Goemans und Williamson

(5.4.4 SDP: Gram-Matrix und geometrische Einbettung)

Sind n Punkte $r_i \in \mathbb{R}^k$ gegeben und ist $R = [r_1, r_2, \dots, r_n]$, dann ist die **Gram-Matrix** $X = R^T R \succeq 0$ positiv semidefinit mit Rang k , und erfüllt

$$X_{ij} = r_i^T r_j,$$
$$\|r_i - r_j\|^2 = r_i^T r_i - 2r_i^T r_j + r_j^T r_j = X_{ii} - 2X_{ij} + X_{jj} = \langle E_{ij}, X \rangle.$$

$$E_{ij} \in \mathbb{R}^{n \times n}, E_{ii} = E_{jj} = 1, E_{ij} = E_{ji} = -1, \text{ und } 0 \text{ sonst: } E_{ij} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{matrix} i \\ j \end{matrix}$$

(5.4.4 SDP: Gram-Matrix und geometrische Einbettung)

Sind n Punkte $r_i \in \mathbb{R}^k$ gegeben und ist $R = [r_1, r_2, \dots, r_n]$, dann ist die **Gram-Matrix** $X = R^T R \succeq 0$ positiv semidefinit mit Rang k , und erfüllt

$$X_{ij} = r_i^T r_j,$$

$$\|r_i - r_j\|^2 = r_i^T r_i - 2r_i^T r_j + r_j^T r_j = X_{ii} - 2X_{ij} + X_{jj} = \langle E_{ij}, X \rangle.$$

$$E_{ij} \in \mathbb{R}^{n \times n}, E_{ii} = E_{jj} = 1, E_{ij} = E_{ji} = -1, \text{ und } 0 \text{ sonst: } E_{ij} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{matrix} i \\ j \end{matrix}$$

Ist umgekehrt die Lage der n Punkte nicht bekannt, sondern nur einige Distanzen $d_{ij} \in \mathbb{R}_+$ zwischen Paaren $ij \in E \subseteq \{ij : 1 \leq i < j \leq n\}$, sucht man eine niedrigdimensionale Einbettung der Punkte mit diesen Distanzen:

$$\begin{aligned} \min \quad & \text{Rang}(X) \\ \text{s.t.} \quad & \langle E_{ij}, X \rangle = d_{ij} \quad ij \in E \\ & X \succeq 0 \end{aligned}$$

(5.4.4 SDP: Gram-Matrix und geometrische Einbettung)

Sind n Punkte $r_i \in \mathbb{R}^k$ gegeben und ist $R = [r_1, r_2, \dots, r_n]$, dann ist die **Gram-Matrix** $X = R^T R \succeq 0$ positiv semidefinit mit Rang k , und erfüllt

$$X_{ij} = r_i^T r_j,$$

$$\|r_i - r_j\|^2 = r_i^T r_i - 2r_i^T r_j + r_j^T r_j = X_{ii} - 2X_{ij} + X_{jj} = \langle E_{ij}, X \rangle.$$

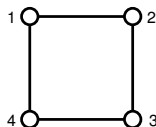
$$E_{ij} \in \mathbb{R}^{n \times n}, E_{ii} = E_{jj} = 1, E_{ij} = E_{ji} = -1, \text{ und } 0 \text{ sonst: } E_{ij} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{matrix} i \\ j \end{matrix}$$

Ist umgekehrt die Lage der n Punkte nicht bekannt, sondern nur einige Distanzen $d_{ij} \in \mathbb{R}_+$ zwischen Paaren $ij \in E \subseteq \{ij : 1 \leq i < j \leq n\}$, sucht man eine niedrigdimensionale Einbettung der Punkte mit diesen Distanzen:

$$\begin{aligned} \min \quad & \text{Rang}(X) \\ \text{s.t.} \quad & \langle E_{ij}, X \rangle = d_{ij} \quad ij \in E \\ & X \succeq 0 \end{aligned}$$

Zwei Schwierigkeiten:

- Das geht gar nicht für beliebige d_{ij} !
Bsp: Kanten-Abstand im Kreis der Länge 4:
 $n = 4, d_{12} = d_{14} = d_{23} = d_{34} = 1, d_{13} = d_{24} = 2$
- Rang-Minimierung ist nicht konvex und *NP*-schwer.



Einbettung mit kleiner Verzerrung

Seien für n Punkte alle paarweisen Distanzen $d_{ij} > 0$ gegeben, und die Dreiecksungleichung $d_{ij} \leq d_{ih} + d_{hj}$ sei für alle i, j, h erfüllt. Eine Einbettung $\rho : \{1, \dots, n\} \rightarrow \mathbb{R}^k$ hat **Verzerrung (distortion)** $D > 0$ falls

$$d_{ij} \leq \|\rho(i) - \rho(j)\| \leq Dd_{ij} \quad \forall 1 \leq i < j \leq n$$

Einbettung mit kleiner Verzerrung

Seien für n Punkte alle paarweisen Distanzen $d_{ij} > 0$ gegeben, und die Dreiecksungleichung $d_{ij} \leq d_{ih} + d_{hj}$ sei für alle i, j, h erfüllt. Eine Einbettung $\rho : \{1, \dots, n\} \rightarrow \mathbb{R}^k$ hat **Verzerrung (distortion)** $D > 0$ falls

$$d_{ij} \leq \|\rho(i) - \rho(j)\| \leq Dd_{ij} \quad \forall 1 \leq i < j \leq n$$

Eine Einbettung geringster Verzerrung in \mathbb{R}^n ist per SDP konstruierbar:

$$\begin{array}{ll} \min & \delta \\ \text{s.t.} & d_{ij} \leq \langle E_{ij}, X \rangle \leq \delta d_{ij} \quad 1 \leq i < j \leq n \\ & X \succeq 0, \delta \in \mathbb{R} \quad [\delta = D^2] \end{array}$$

Faktorisiere X^* zum Beispiel über die Eigenwertzerlegung:

$$X^* = P\Lambda^*P^T = P(\Lambda^*)^{\frac{1}{2}} \underbrace{(\Lambda^*)^{\frac{1}{2}}P^T}_{=:R} = R^T R$$

Der Rang von R ist mit geringem Verlust in D auf $\sim \log n$ reduzierbar.