Semidefinite programming

The what, why and how.

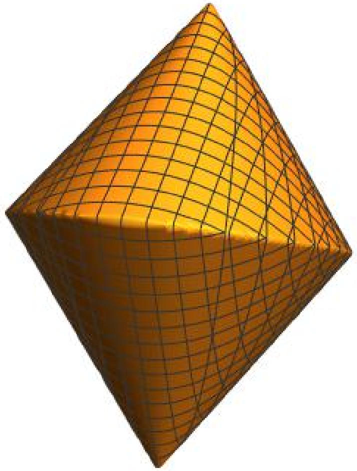
PETER BROWN -- TÉLÉCOM PARIS / INRIA

QSI SCHOOL ON QUANTUM CRYPTOGRAPHY - 31.01.24

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & \sum_i \lambda_i \omega_i \\ \text{s.t.} \quad & \sum_i \lambda_i F_i - C \succeq 0 \\ & \lambda_i \geq 0 \quad \forall\, i \end{aligned}$$

The what



SDPs - A first example

This is an SDP

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x + y \\ x + y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

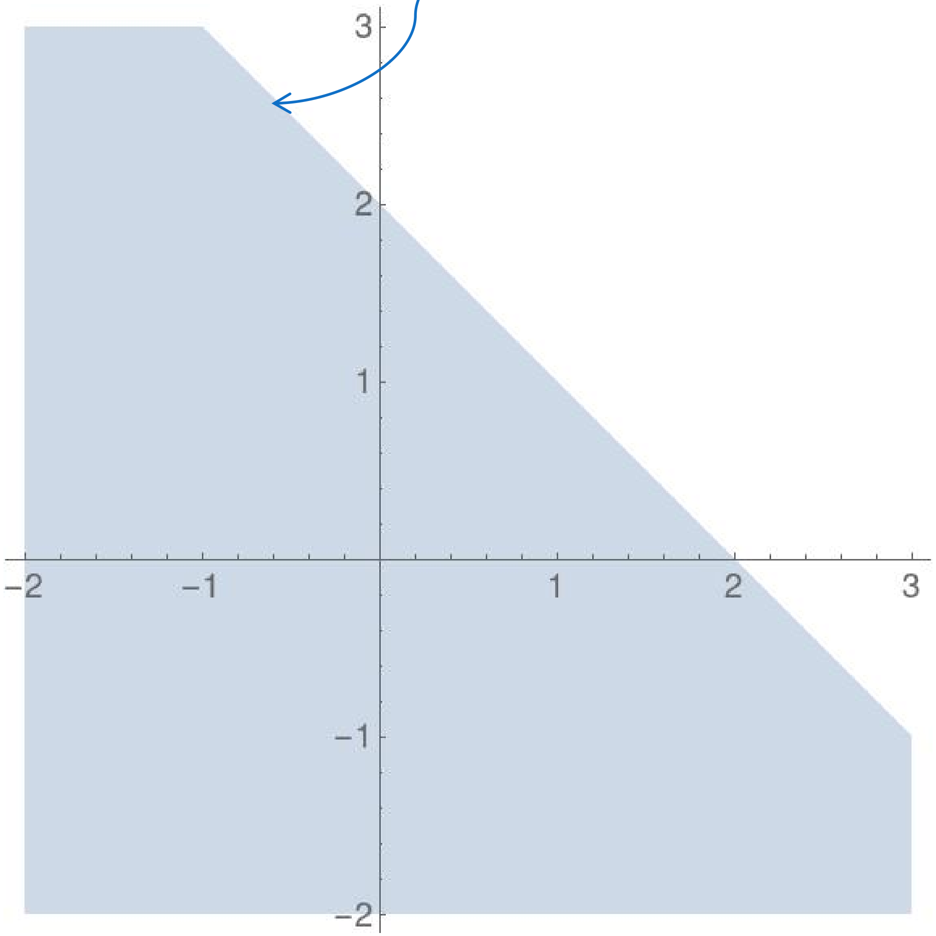
SDPs - A first example

This is an SDP

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

Let's solve it...

$x + y \leq 2$



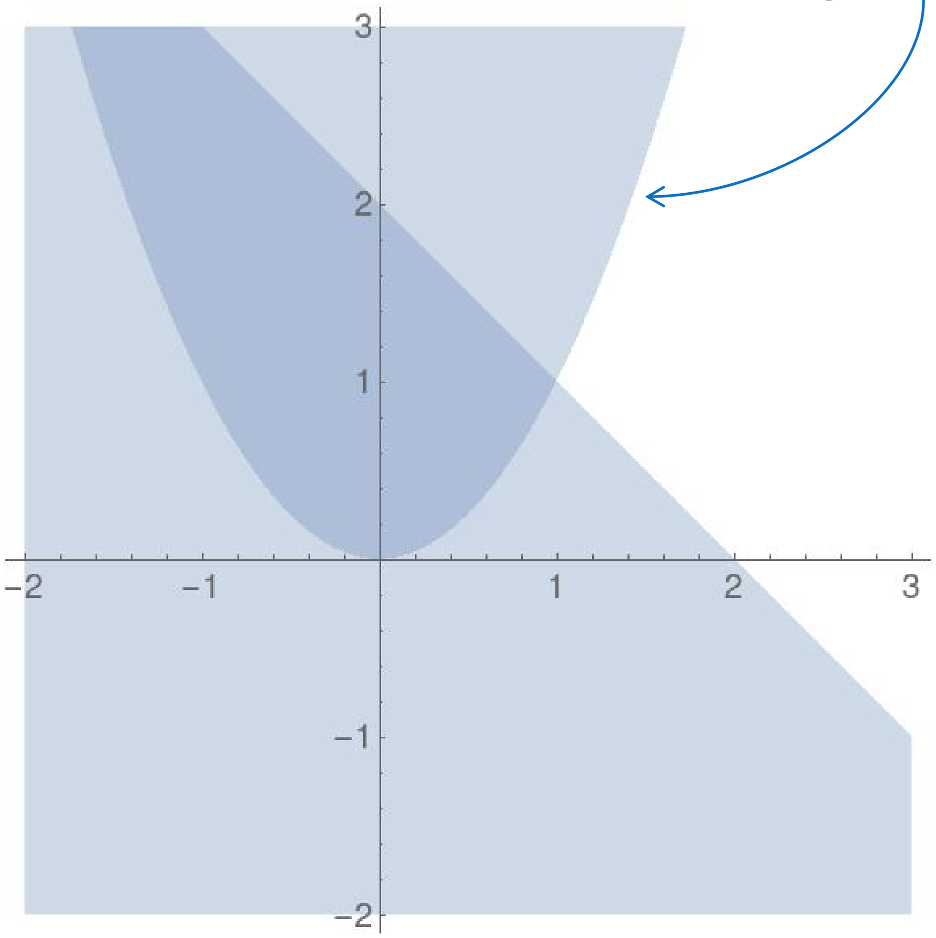
SDPs - A first example

This is an SDP

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \boxed{\begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0} \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

Let's solve it...

$y \geq x^2$

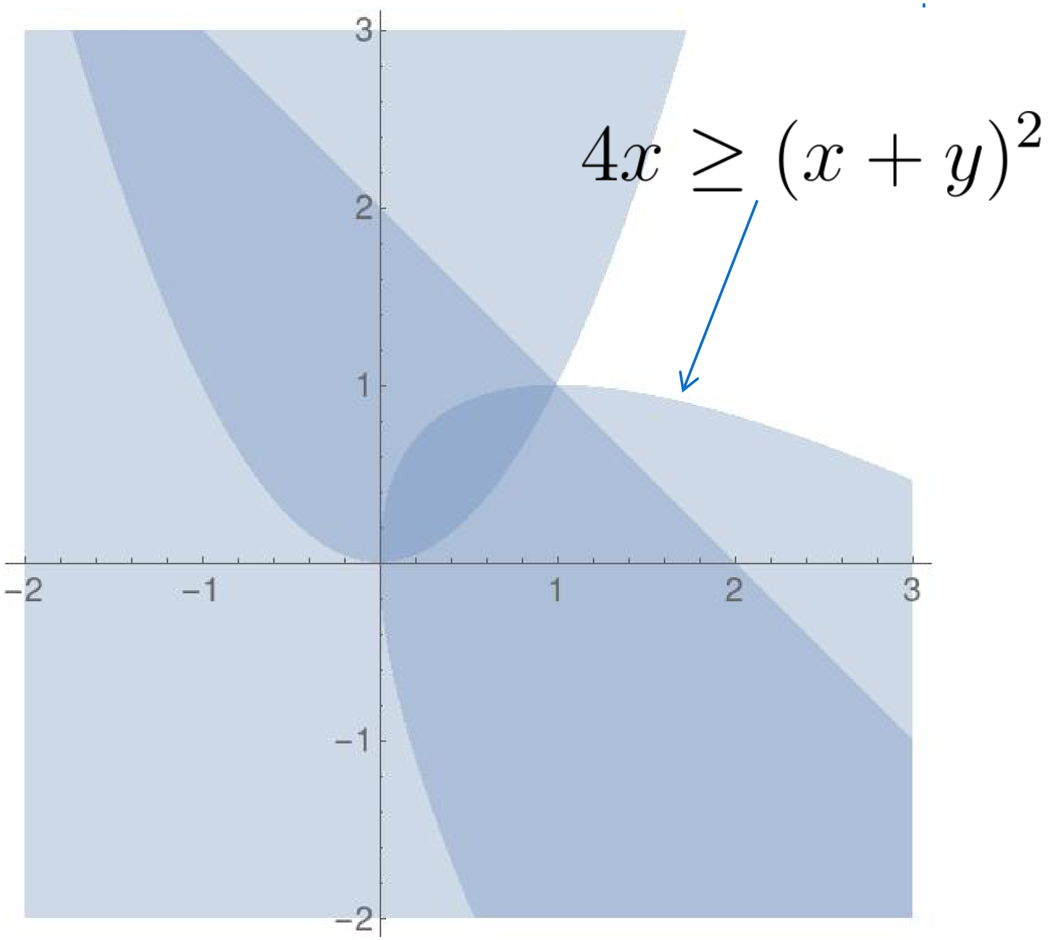


SDPs - A first example

This is an SDP

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \boxed{\begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0} \end{aligned}$$

Let’s solve it…

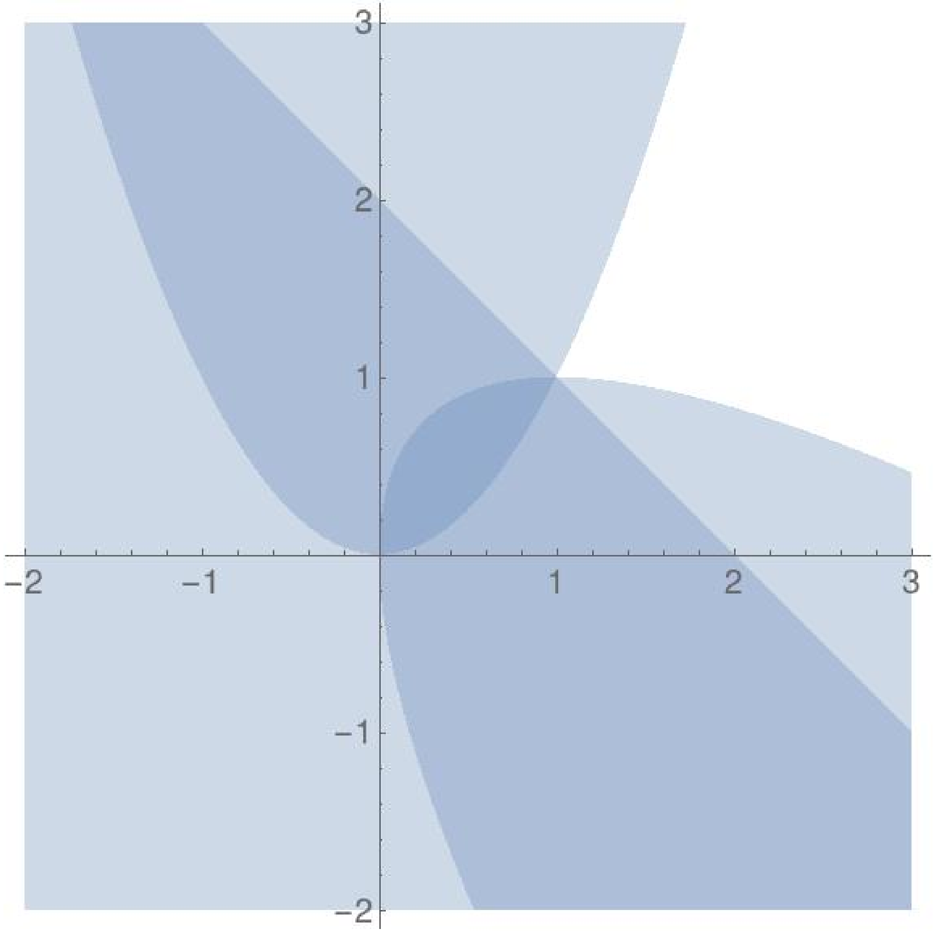


SDPs - A first example

This is an SDP

$$
\begin{aligned}
\max \quad & x \\
\text{s.t.} \quad & x + y \leq 2 \\
& \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\
& \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0
\end{aligned}
$$

Biggest x satisfying all constraints?

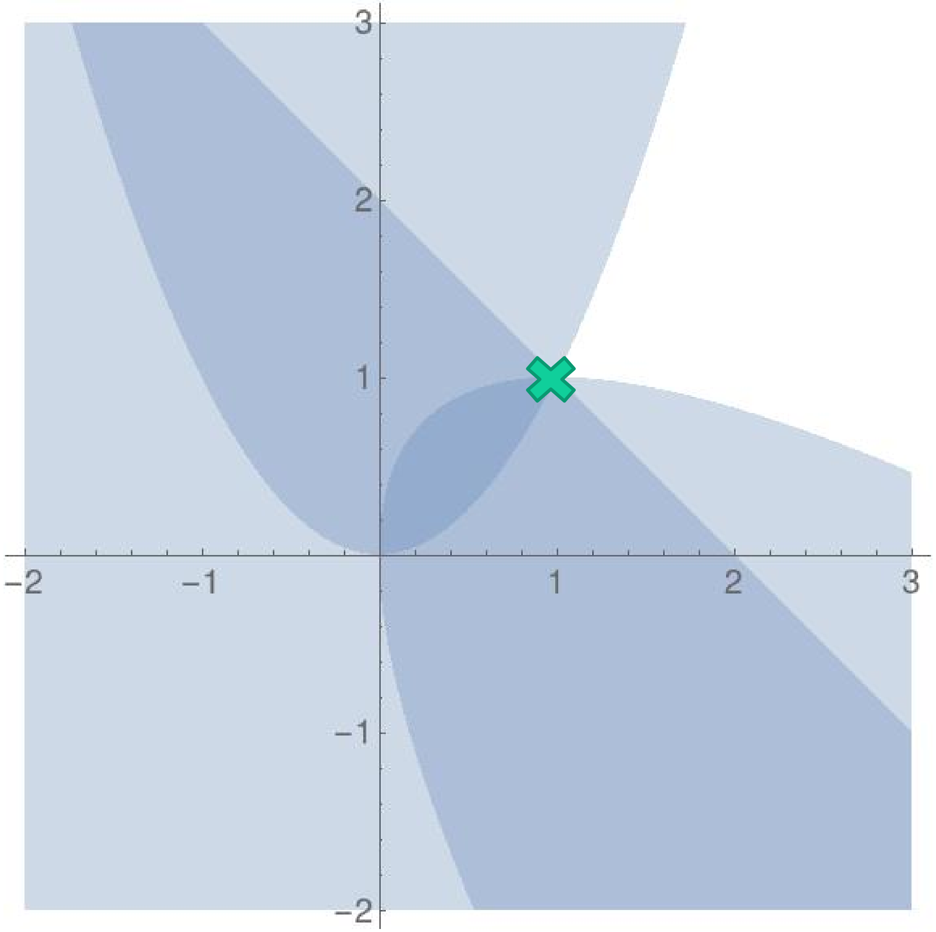


SDPs - A first example

This is an SDP

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \le 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

$$(x^*, y^*) = (1, 1)$$



So what's an SDP?

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x + y \\ x + y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

SDPs are optimization problems satisfying

- Linear objective function
- Linear (in)equalities
- Semidefinite constraints (variables appear linearly)

So what’s an SDP?

$$
\begin{aligned}
\max \quad & x \\
\text{s.t.} \quad & x + y \leq 2 \\
& \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\
& \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0
\end{aligned}
$$

SDPs are optimization problems satisfying

- Linear objective function
- Linear (in)equalities
- Semidefinite constraints (variables appear linearly)

Importantly

- Efficient to solve!
- Guaranteed global optima!

SDP standard forms

Definition (SDP)
An SDP is an optimization of the following form

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

where the Hermitian matrices $C,\ F_i$ and $\omega_i \in \mathbb{R}$ are fixed.

Note:

$$\mathrm{Tr}[AX] = \sum_{ij} a_{ij} x_{ji}$$

Just linear combination of elements of X

SDP standard forms

Definition (SDP)
An SDP is an optimization of the following form

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

where the Hermitian matrices $C,\ F_i$ and $\omega_i \in \mathbb{R}$ are fixed.

Note:

$$\mathrm{Tr}[AX] = \sum_{ij} a_{ij} x_{ji}$$

Just linear combination of elements of X

Remarks:

- Any SDP can be rewritten into the “standard form”.

  In practice we rarely convert to them though.

SDP standard forms

Definition (SDP)

An SDP is an optimization of the following form

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall i \\ & X \succeq 0 \end{aligned}$$

where the Hermitian matrices $C,\ F_i$ and $\omega_i \in \mathbb{R}$ are fixed.

Note:

$$\mathrm{Tr}[AX] = \sum_{ij} a_{ij} x_{ji}$$

Just linear combination of elements of X

Remarks:

- Any SDP can be rewritten into the "standard form".

  In practice we rarely convert to them though.

- Standard forms not unique across literature, e.g.

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \Phi(X) = B \\ & X \succeq 0 \end{aligned}$$

where Φ is a Hermitian preserving linear map.

[Watrous, The Theory of Quantum Information (2018)]

SDP standard forms

Definition (SDP)

An SDP is an optimization of the following form

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \le \omega_i \quad \forall i \\ & X \succeq 0 \end{aligned}$$

where the Hermitian matrices $C,\ F_i$ and $\omega_i \in \mathbb{R}$ are fixed.

Note:

$$\mathrm{Tr}[AX] = \sum_{ij} a_{ij} x_{ji}$$

Just linear combination of elements of X

Remarks:

- Any SDP can be rewritten into the "standard form".

  In practice we rarely convert to them though.

- Standard forms not unique across literature, e.g.

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \Phi(X) = B \\ & X \succeq 0 \end{aligned}$$

where Φ is a Hermitian preserving linear map.

[Watrous, The Theory of Quantum Information (2018)]

Exercise: Convert between the two standard forms.

SDP values

SDPs fall into one of the following categories:

Optimal value exists and attained

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \leq 2 \end{aligned}$$

SDP values

SDPs fall into one of the following categories:

Optimal value exists and attained

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \leq 2 \end{aligned}$$

Optimal value exists but *not* attained

$$\begin{aligned} \min \quad & x \\ \text{s.t.} \quad & \begin{pmatrix} x & 1 \\ 1 & y \end{pmatrix} \succeq 0 \end{aligned}$$

SDP values

SDPs fall into one of the following categories:

Optimal value exists and attained

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \leq 2 \end{aligned}$$

Optimal value exists but *not* attained

$$\begin{aligned} \min \quad & x \\ \text{s.t.} \quad & \begin{pmatrix} x & 1 \\ 1 & y \end{pmatrix} \succeq 0 \end{aligned}$$

Infeasible

SDP values

SDPs fall into one of the following categories:

Optimal value exists and attained

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \leq 2 \end{aligned}$$

Optimal value exists but *not* attained

$$\begin{aligned} \min \quad & x \\ \text{s.t.} \quad & \begin{pmatrix} x & 1 \\ 1 & y \end{pmatrix} \succeq 0 \end{aligned}$$

Infeasible

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \leq -1 \\ & x \geq 0 \end{aligned}$$

SDP values

SDPs fall into one of the following categories:

Optimal value exists and attained

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \leq 2 \end{aligned}$$

Optimal value exists but *not* attained

$$\begin{aligned} \min \quad & x \\ \text{s.t.} \quad & \begin{pmatrix} x & 1 \\ 1 & y \end{pmatrix} \succeq 0 \end{aligned}$$

Infeasible

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \leq -1 \\ & x \geq 0 \end{aligned}$$

Unbounded

SDP values

SDPs fall into one of the following categories:

Optimal value exists and attained

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \leq 2 \end{aligned}$$

Optimal value exists but *not* attained

$$\begin{aligned} \min \quad & x \\ \text{s.t.} \quad & \begin{pmatrix} x & 1 \\ 1 & y \end{pmatrix} \succeq 0 \end{aligned}$$

Infeasible

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \leq -1 \\ & x \geq 0 \end{aligned}$$

Unbounded

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x \geq 0 \end{aligned}$$

SDP duality I - Upper bounds

Lower bounds on

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

are “easy” to find. **(why?)**

SDP duality I - Upper bounds

Lower bounds on

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

are “easy” to find. **(why?)**

What about upper bounds?

SDP duality I - Upper bounds

Lower bounds on

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

are “easy” to find. **(why?)**

What about upper bounds?

Suppose you can find $\lambda_i \geq 0$ such that

$$\sum_i \lambda_i F_i - C \succeq 0$$

SDP duality I - Upper bounds

Lower bounds on

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

are "easy" to find. **(why?)**

What about upper bounds?

Suppose you can find $\lambda_i \geq 0$ such that

$$\sum_i \lambda_i F_i - C \succeq 0$$

Then for any feasible X we have

$$0 \leq \mathrm{Tr}[(\sum_i \lambda_i F_i - C)X] \qquad \textbf{(PSD)}$$

SDP duality I - Upper bounds

Lower bounds on

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

are “easy” to find. **(why?)**

What about upper bounds?

Suppose you can find $\lambda_i \geq 0$ such that

$$\sum_i \lambda_i F_i - C \succeq 0$$

Then for any feasible X we have

$$0 \leq \mathrm{Tr}[(\sum_i \lambda_i F_i - C)X] \qquad \textbf{(PSD)}$$

$$= \sum_i \lambda_i \mathrm{Tr}[F_i X] - \mathrm{Tr}[CX] \qquad \textbf{(Linearity)}$$

SDP duality I - Upper bounds

Lower bounds on

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \le \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

are “easy” to find. **(why?)**

What about upper bounds?

Suppose you can find $\lambda_i \ge 0$ such that

$$\sum_i \lambda_i F_i - C \succeq 0$$

Then for any feasible X we have

$$0 \le \mathrm{Tr}[(\sum_i \lambda_i F_i - C)X] \qquad \textbf{(PSD)}$$

$$= \sum_i \lambda_i \mathrm{Tr}[F_i X] - \mathrm{Tr}[CX] \qquad \textbf{(Linearity)}$$

$$\le \sum_i \lambda_i \omega_i - \mathrm{Tr}[CX] \qquad \textbf{(Feasibility)}$$

SDP duality I - Upper bounds

Lower bounds on

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \le \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

are “easy” to find. **(why?)**

What about upper bounds?

$$\mathrm{Tr}[CX] \le \sum_i \lambda_i \omega_i$$

Suppose you can find $\lambda_i \geq 0$ such that

$$\sum_i \lambda_i F_i - C \succeq 0$$

Then for any feasible X we have

$$0 \le \mathrm{Tr}[(\sum_i \lambda_i F_i - C)X] \qquad \textbf{(PSD)}$$

$$= \sum_i \lambda_i \mathrm{Tr}[F_i X] - \mathrm{Tr}[CX] \qquad \textbf{(Linearity)}$$

$$\le \sum_i \lambda_i \omega_i - \mathrm{Tr}[CX] \qquad \textbf{(Feasibility)}$$

SDP duality II - The dual

If we define the *primal* SDP as

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

Then the *dual* SDP is

$$\begin{aligned} \min \quad & \sum_i \lambda_i \omega_i \\ \text{s.t.} \quad & \sum_i \lambda_i F_i - C \succeq 0 \\ & \lambda_i \geq 0 \quad \forall\, i \end{aligned}$$

(Best upper bound on primal SDP)

SDP duality II - The dual

If we define the *primal* SDP as

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

Then the *dual* SDP is

$$\begin{aligned} \min \quad & \sum_i \lambda_i \omega_i \\ \text{s.t.} \quad & \sum_i \lambda_i F_i - C \succeq 0 \\ & \lambda_i \geq 0 \quad \forall\, i \end{aligned}$$

(Best upper bound on primal SDP)

- Dual of the dual is the primal.

SDP duality II - The dual

If we define the *primal* SDP as

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

Then the *dual* SDP is

$$\begin{aligned} \min \quad & \sum_i \lambda_i \omega_i \\ \text{s.t.} \quad & \sum_i \lambda_i F_i - C \succeq 0 \\ & \lambda_i \geq 0 \quad \forall\, i \end{aligned}$$

(Best upper bound on primal SDP)

- Dual of the dual is the primal.
- By construction $\mathrm{Tr}[CX^*] \leq \sum_i \lambda_i^* \omega_i$ **(Weak duality)**

SDP duality II - The dual

If we define the *primal* SDP as

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall i \\ & X \succeq 0 \end{aligned}$$

Then the *dual* SDP is

$$\begin{aligned} \min \quad & \sum_i \lambda_i \omega_i \\ \text{s.t.} \quad & \sum_i \lambda_i F_i - C \succeq 0 \\ & \lambda_i \geq 0 \quad \forall i \end{aligned}$$

(Best upper bound on primal SDP)

- Dual of the dual is the primal.
- By construction $\mathrm{Tr}[CX^*] \leq \sum_i \lambda_i^* \omega_i$ **(Weak duality)**
- Equality (*strong duality*) almost always... but not always.

SDP duality II - The dual

If we define the *primal* SDP as

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

Then the *dual* SDP is

$$\begin{aligned} \min \quad & \sum_i \lambda_i \omega_i \\ \text{s.t.} \quad & \sum_i \lambda_i F_i - C \succeq 0 \\ & \lambda_i \geq 0 \quad \forall\, i \end{aligned}$$

(Best upper bound on primal SDP)

- Dual of the dual is the primal.
- By construction $\mathrm{Tr}[CX^*] \leq \sum_i \lambda_i^* \omega_i$ **(Weak duality)**
- Equality (*strong duality*) almost always... but not always.
- Duals can give new insights / simpler forms / new interpretations!

SDP duality III - Strong duality

When does the primal value equal the dual value?

Theorem (Slater):

1. If the primal is *strictly feasible* then the primal value equals dual value and dual is attained.
2. If the dual is *strictly feasible* then the primal value equals dual value and primal is attained.

SDP duality III - Strong duality

Theorem (Slater):

When does the primal value equal the dual value?

1. If the primal is *strictly feasible* then the primal value equals dual value and dual is attained.
2. If the dual is *strictly feasible* then the primal value equals dual value and primal is attained.

Example

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

SDP duality III - Strong duality

When does the primal value equal the dual value?

Theorem (Slater):

1. If the primal is *strictly feasible* then the primal value equals dual value and dual is attained.
2. If the dual is *strictly feasible* then the primal value equals dual value and primal is attained.

Example

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

A bit of magic...

$$\longrightarrow$$

SDP duality III - Strong duality

When does the primal value equal the dual value?

Theorem (Slater):

1. If the primal is *strictly feasible* then the primal value equals dual value and dual is attained.
2. If the dual is *strictly feasible* then the primal value equals dual value and primal is attained.

Example

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

A bit of magic... $\longrightarrow$

$$\begin{aligned} \min \quad & a + 2c + 2d + 4e \\ \text{s.t.} \quad & 1 + 2b - c + 2f = 0 \\ & \begin{pmatrix} a & b \\ b & c \end{pmatrix} \succeq 0, \begin{pmatrix} d & e \\ e & f \end{pmatrix} \succeq 0 \end{aligned}$$

SDP duality III - Strong duality

Theorem (Slater):

When does the primal value equal the dual value?

1. If the primal is *strictly feasible* then the primal value equals dual value and dual is attained.
2. If the dual is *strictly feasible* then the primal value equals dual value and primal is attained.

Example

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \le 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

A bit of magic…

$$\begin{aligned} \min \quad & a + 2c + 2d + 4e \\ \text{s.t.} \quad & 1 + 2b - c + 2f = 0 \\ & \begin{pmatrix} a & b \\ b & c \end{pmatrix} \succeq 0, \ \begin{pmatrix} d & e \\ e & f \end{pmatrix} \succeq 0 \end{aligned}$$

Strictly feasible $(x, y) = (1/2, 1/2)$

Optimal value: 1

SDP duality III - Strong duality

Theorem (Slater):

When does the primal value equal the dual value?

1. If the primal is *strictly feasible* then the primal value equals dual value and dual is attained.
2. If the dual is *strictly feasible* then the primal value equals dual value and primal is attained.

Example

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

$\xrightarrow{\text{A bit of magic...}}$

$$\begin{aligned} \min \quad & a + 2c + 2d + 4e \\ \text{s.t.} \quad & 1 + 2b - c + 2f = 0 \\ & \begin{pmatrix} a & b \\ b & c \end{pmatrix} \succeq 0, \ \begin{pmatrix} d & e \\ e & f \end{pmatrix} \succeq 0 \end{aligned}$$

Strictly feasible $(x, y) = (1/2, 1/2)$

Optimal value: 1

Solution: $a = -b = c = 1/3$ Optimal value: 1

$d = e = f = 0$

SDP duality III - Strong duality

Theorem (Slater):

When does the primal value equal the dual value?

1. If the primal is *strictly feasible* then the primal value equals dual value and dual is attained.
2. If the dual is *strictly feasible* then the primal value equals dual value and primal is attained.

Example

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

A bit of magic... $\longrightarrow$

$$\begin{aligned} \min \quad & a + 2c + 2d + 4e \\ \text{s.t.} \quad & 1 + 2b - c + 2f = 0 \\ & \begin{pmatrix} a & b \\ b & c \end{pmatrix} \succeq 0, \ \begin{pmatrix} d & e \\ e & f \end{pmatrix} \succeq 0 \end{aligned}$$

Strictly feasible $(x, y) = (1/2, 1/2)$

Optimal value: 1

Solution: $a = -b = c = 1/3$ $\quad d = e = f = 0$

Optimal value: 1

Guarantees optimal solution found!

SDPs - A summary of the what

- Optimization problems involving **linear (in)equalities** and **positive semidefinite inequalities.**

SDPs - A summary of the what

- Optimization problems involving **linear (in)equalities** and **positive semidefinite inequalities.**

- Computationally efficient 

SDPs - A summary of the what

- Optimization problems involving **linear (in)equalities** and **positive semidefinite inequalities.**

- Computationally efficient 

- Guaranteed to find optimal value (approximately)

SDPs - A summary of the what

- Optimization problems involving **linear (in)equalities** and **positive semidefinite inequalities.**

- Computationally efficient

- Guaranteed to find optimal value (approximately)

- Two perspectives on the problem (primal & dual)

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & \sum_i \lambda_i \omega_i \\ \text{s.t.} \quad & \sum_i \lambda_i F_i - C \succeq 0 \\ & \lambda_i \geq 0 \quad \forall\, i \end{aligned}$$

SDPs - A summary of the what

- Optimization problems involving **linear (in)equalities** and **positive semidefinite inequalities.**

- Computationally efficient

- Guaranteed to find optimal value (approximately)

- Two perspectives on the problem (primal & dual)

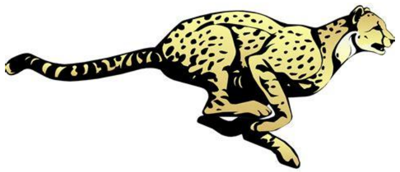
- Weak duality **always** holds.

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned} \qquad \leq \qquad \begin{aligned} \min \quad & \sum_i \lambda_i \omega_i \\ \text{s.t.} \quad & \sum_i \lambda_i F_i - C \succeq 0 \\ & \lambda_i \geq 0 \quad \forall\, i \end{aligned}$$

SDPs - A summary of the what

- Optimization problems involving **linear (in)equalities** and **positive semidefinite inequalities.**

- Computationally efficient



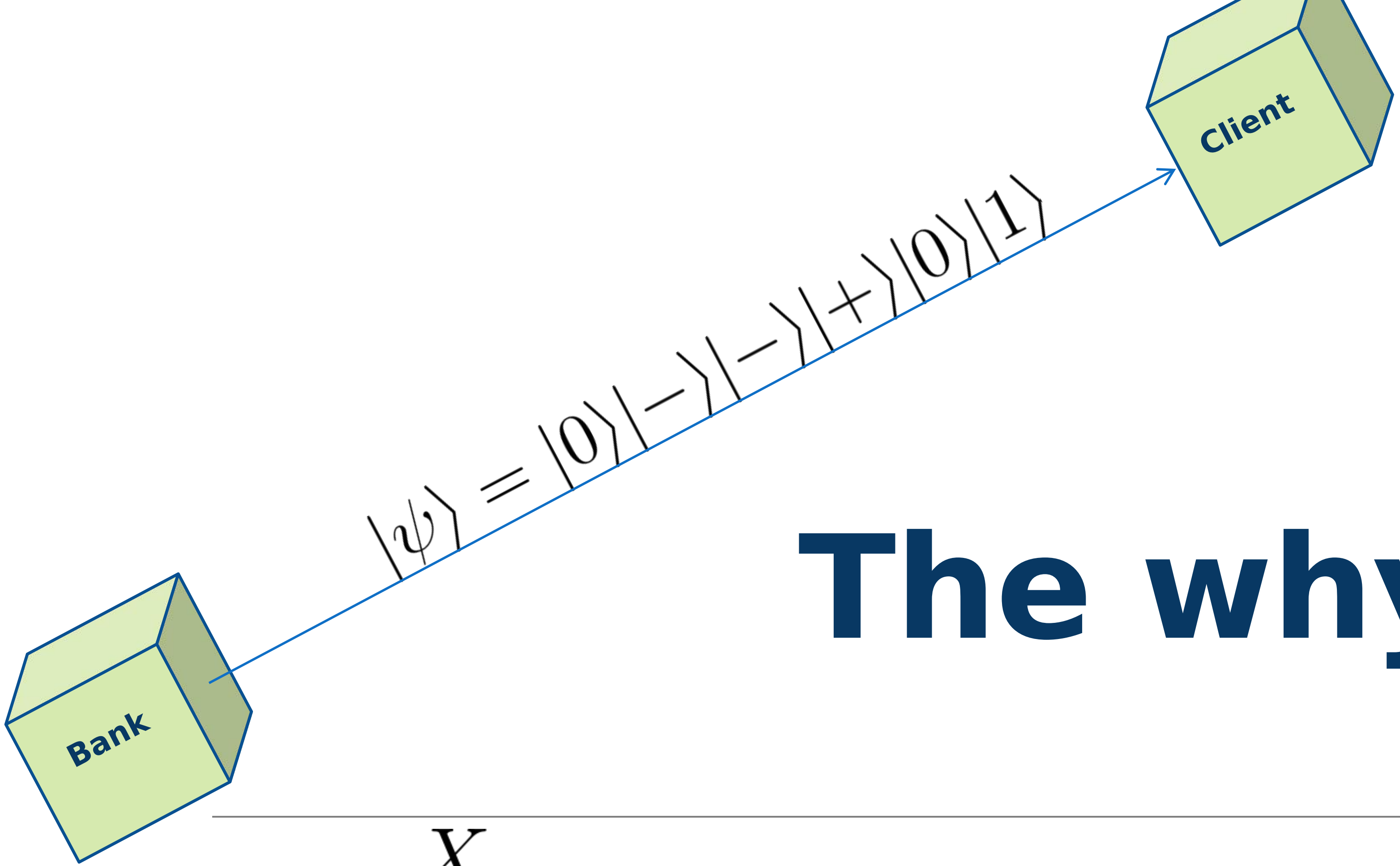
- Guaranteed to find optimal value (approximately)

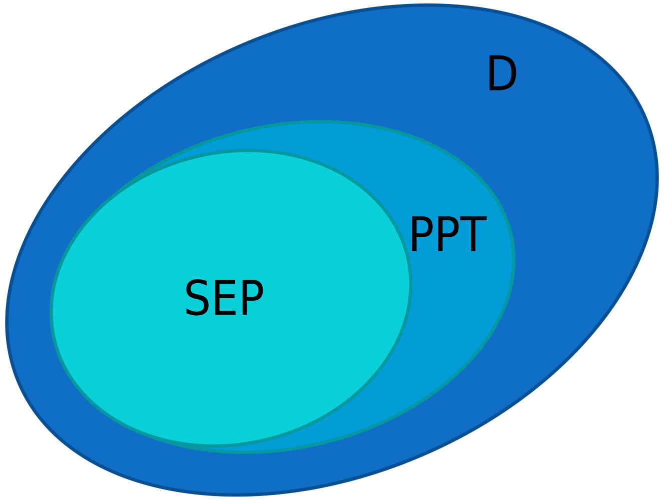
- Two perspectives on the problem (primal & dual)

$$\begin{aligned} \max \quad & \mathrm{Tr}[CX] \\ \text{s.t.} \quad & \mathrm{Tr}[F_i X] \leq \omega_i \quad \forall\, i \\ & X \succeq 0 \end{aligned} \qquad \leq \qquad \begin{aligned} \min \quad & \sum_i \lambda_i \omega_i \\ \text{s.t.} \quad & \sum_i \lambda_i F_i - C \succeq 0 \\ & \lambda_i \geq 0 \quad \forall\, i \end{aligned}$$

- Weak duality **always** holds.

- Strong duality **almost always** holds
  (easy sufficient condition to check).


Bank
$|\psi\rangle = |0\rangle|-\rangle|-\rangle|+\rangle|0\rangle|1\rangle$
Client


D
PPT
SEP

The why


X
$p(x)$
Source
ρ_x
POVM
$\{M_y\}_y$
Y

SDPs and quantum?

Quantum systems are a mishmash of PSD things with linear constraints

SDPs and quantum?

Quantum systems are a mishmash of PSD things with linear constraints

Quantum states $$\{X \mid X \succeq 0, \operatorname{Tr}[X] = 1\}$$

SDPs and quantum?

Quantum systems are a mishmash of PSD things with linear constraints

Quantum states $\{X \mid X \succeq 0, \mathrm{Tr}[X] = 1\}$

Quantum measurements $\{(X_1, \ldots, X_r) \mid X_i \succeq 0, \sum_i X_i = \mathbb{I}\}$

SDPs and quantum?

Quantum systems are a mishmash of PSD things with linear constraints

Quantum states $$\{X \mid X \succeq 0,\ \mathrm{Tr}[X] = 1\}$$

Quantum measurements $$\{(X_1, \ldots, X_r) \mid X_i \succeq 0,\ \sum_i X_i = \mathbb{I}\}$$

Quantum channels $$\{C_{AB} \mid C_{AB} \succeq 0,\ \mathrm{Tr}_B[C_{AB}] = \mathbb{I}_A\}$$

$$\Phi : L(A) \to L(B) \qquad C_{AB} := \sum_{ij} |i\rangle\langle j| \otimes \Phi(|i\rangle\langle j|)$$

SDPs and quantum?

Quantum systems are a mishmash of PSD things with linear constraints

Quantum states

$$\{X \mid X \succeq 0,\ \mathrm{Tr}[X] = 1\}$$

Quantum measurements

$$\{(X_1, \ldots, X_r) \mid X_i \succeq 0,\ \sum_i X_i = \mathbb{I}\}$$

Quantum channels

$$\{C_{AB} \mid C_{AB} \succeq 0,\ \mathrm{Tr}_B[C_{AB}] = \mathbb{I}_A\}$$

SDPs are lurking everywhere...

$$\Phi : L(A) \to L(B) \qquad C_{AB} := \sum_{ij} |i\rangle\langle j| \otimes \Phi(|i\rangle\langle j|)$$

State discrimination

State discrimination

$$P_{\text{guess}} = \sum_x p(x)\,\mathrm{Tr}[\rho_x M_x]$$

State discrimination

$$P_{\text{guess}} = \sum_x p(x)\,\mathrm{Tr}[\rho_x M_x]$$

What’s the maximal guessing probability? (Best measurement)

State discrimination

$$P_{\text{guess}} = \sum_x p(x)\,\mathrm{Tr}[\rho_x M_x]$$

What's the maximal guessing probability? (Best measurement)

Example

I send $|0\rangle$ or $|1\rangle$ with uniform probability.

State discrimination

$$P_{\text{guess}} = \sum_x p(x)\,\mathrm{Tr}[\rho_x M_x]$$

What's the maximal guessing probability? (Best measurement)

Example

I send $|0\rangle$ or $|1\rangle$ with uniform probability.

$$M_0 = |0\rangle\langle 0|, \quad M_1 = |1\rangle\langle 1|$$

Just computational basis, perfect discrimination!

State discrimination

$$P_{\text{guess}} = \sum_x p(x)\,\mathrm{Tr}[\rho_x M_x]$$

What's the maximal guessing probability? (Best measurement)

Example

I send $|0\rangle$ or $|1\rangle$ with uniform probability.

$$M_0 = |0\rangle\langle 0|, \quad M_1 = |1\rangle\langle 1|$$

Just computational basis, perfect discrimination!

General case (It's just an SDP :))

$$\max \quad \sum_x p(x)\mathrm{Tr}[\rho_x M_x]$$

$$\sum_x M_x = \mathbb{I}$$

$$M_x \succeq 0 \quad \forall x$$

State discrimination

More complex example

I send $|0\rangle$ with prob 1/2, $|+\rangle$ with prob 1/4 and $|-\rangle$ with prob 1/4.

Still simple but already tough to solve by hand.

State discrimination

More complex example

I send $|0\rangle$ with prob 1/2, $|+\rangle$ with prob 1/4 and $|-\rangle$ with prob 1/4.

Still simple but already tough to solve by hand.

Immediate solution from SDP: optimal probability 2/3

State discrimination

More complex example

I send $|0\rangle$ with prob 1/2, $|+\rangle$ with prob 1/4 and $|-\rangle$ with prob 1/4.

Still simple but already tough to solve by hand.

Immediate solution from SDP: optimal probability 2/3

Not just optimal probability

State discrimination

More complex example

I send $|0\rangle$ with prob 1/2, $|+\rangle$ with prob 1/4 and $|-\rangle$ with prob 1/4.

Still simple but already tough to solve by hand.

Immediate solution from SDP: optimal probability 2/3

Not just optimal probability

$$M_0 = \begin{pmatrix} 8/9 & 0 \\ 0 & 0 \end{pmatrix} \quad M_1 = \begin{pmatrix} 1/18 & 1/6 \\ 1/6 & 1/2 \end{pmatrix} \quad M_2 = \begin{pmatrix} 1/18 & -1/6 \\ -1/6 & 1/2 \end{pmatrix}$$

A recipe for how to perform the experiment!

Discrimination - many variants

- Fundamental information processing task -- channel coding

Discrimination - many variants

- Fundamental information processing task -- channel coding
- Crypto applications -- how well can Eve guess X (secret key)?

$$H_{\min}(A|B) \quad H^{\epsilon}_{\min}(A|B)$$

Both computable via SDP

Discrimination - many variants

- Fundamental information processing task -- channel coding
- Crypto applications -- how well can Eve guess X (secret key)?
- Interesting variants
  1. Never wrong but allowed to say “I don’t know” (Unambiguous)
  2. Discriminate between measurements
  3. Antidiscrimination -- “How badly can I discriminate?”

$H_{\min}(A|B) \quad H^{\epsilon}_{\min}(A|B)$

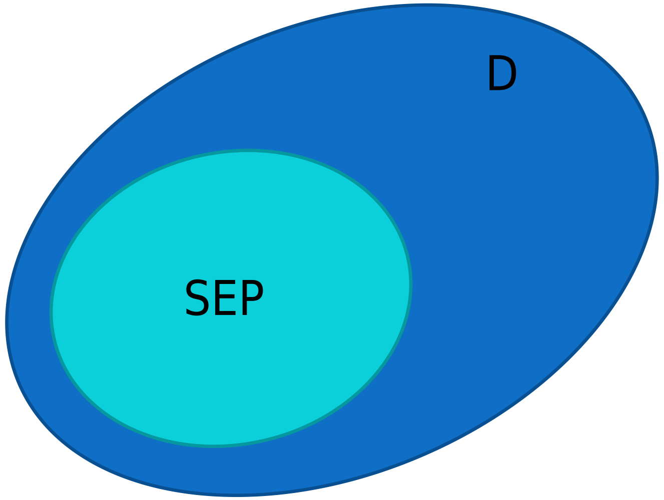
Both computable via SDP

Entanglement witnessing

A bipartite state ρ_{AB} is *separable* if there exist states τ_i on A, σ_i on B, and a distribution p_i such that

$$\rho_{AB} = \sum_i p_i \tau_i \otimes \sigma_i$$

otherwise we say ρ_{AB} is *entangled*.



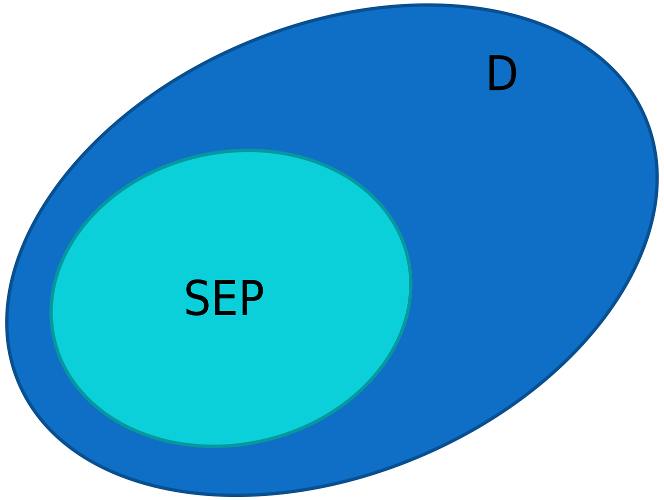
Entanglement witnessing

A bipartite state ρ_{AB} is *separable* if there exist states τ_i on A, σ_i on B, and a distribution p_i such that

$$\rho_{AB} = \sum_i p_i \tau_i \otimes \sigma_i$$

Hard to check!

otherwise we say ρ_{AB} is *entangled*.



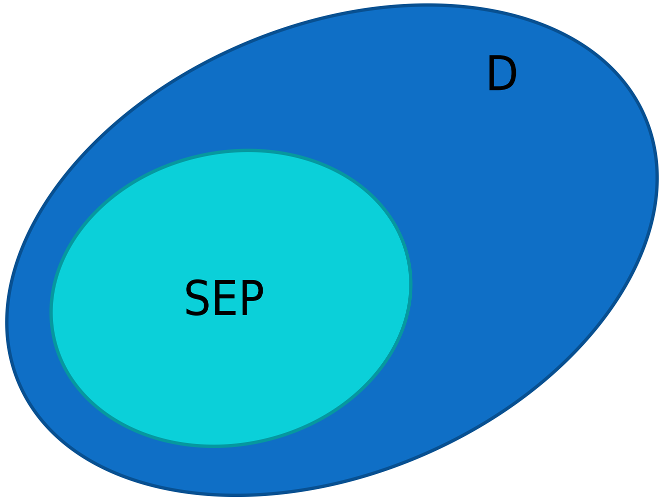
Entanglement witnessing

A bipartite state ρ_{AB} is *separable* if there exist states τ_i on A, σ_i on B, and a distribution p_i such that

$$\rho_{AB} = \sum_i p_i \tau_i \otimes \sigma_i$$

Hard to check!

otherwise we say ρ_{AB} is *entangled*.



Define *partial transpose on B* as the linear map with the action

$$(|i\rangle\langle j|_A \otimes |k\rangle\langle l|_B)^{T_B} = |i\rangle\langle j|_A \otimes |l\rangle\langle k|_B$$

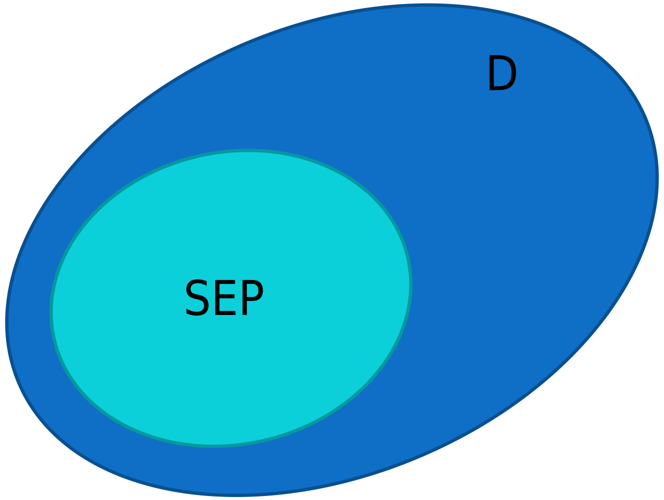
Entanglement witnessing

A bipartite state ρ_{AB} is *separable* if there exist states τ_i on A, σ_i on B, and a distribution p_i such that

$$\rho_{AB} = \sum_i p_i \tau_i \otimes \sigma_i$$

Hard to check!

otherwise we say ρ_{AB} is *entangled*.



Define *partial transpose on B* as the linear map with the action

$$\left(|i\rangle\langle j|_A \otimes |k\rangle\langle l|_B\right)^{T_B} = |i\rangle\langle j|_A \otimes |l\rangle\langle k|_B$$

Separable states remain PSD under partial transpose $\qquad \rho_{AB}^{T_B} = \sum_i p_i \tau_i \otimes \sigma_i^T \succeq 0$

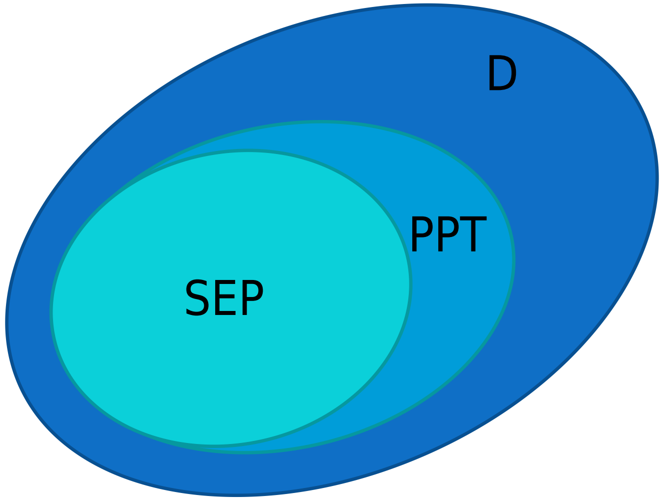
Entanglement witnessing

A bipartite state ρ_{AB} is *separable* if there exist states τ_i on A, σ_i on B, and a distribution p_i such that

$$\rho_{AB} = \sum_i p_i \tau_i \otimes \sigma_i$$

otherwise we say ρ_{AB} is *entangled*.

Hard to check!



Define *partial transpose on B* as the linear map with the action

$$\left(|i\rangle\langle j|_A \otimes |k\rangle\langle l|_B\right)^{T_B} = |i\rangle\langle j|_A \otimes |l\rangle\langle k|_B$$

Separable states remain PSD under partial transpose $\quad \rho_{AB}^{T_B} = \sum_i p_i \tau_i \otimes \sigma_i^T \succeq 0$

Consider the set of all states that remain positive after partial transpose

$$\mathrm{PPT} := \{\rho_{AB} \mid \mathrm{Tr}[\rho] = 1,\, \rho \succeq 0,\, \rho^{T_B} \succeq 0\}$$

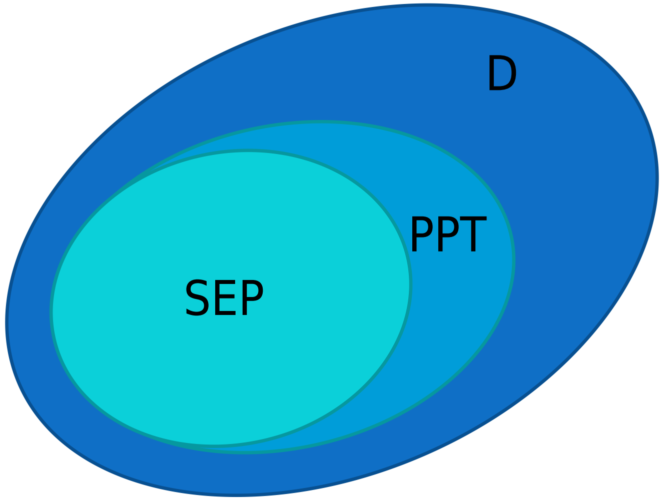
Entanglement witnessing

A bipartite state ρ_{AB} is *separable* if there exist states τ_i on A, σ_i on B, and a distribution p_i such that

$$\rho_{AB} = \sum_i p_i \tau_i \otimes \sigma_i$$

Hard to check!

otherwise we say ρ_{AB} is *entangled*.



Define *partial transpose on B* as the linear map with the action

$$\left(|i\rangle\langle j|_A \otimes |k\rangle\langle l|_B\right)^{T_B} = |i\rangle\langle j|_A \otimes |l\rangle\langle k|_B$$

Separable states remain PSD under partial transpose $\rho_{AB}^{T_B} = \sum_i p_i \tau_i \otimes \sigma_i^T \succeq 0$

Consider the set of all states that remain positive after partial transpose

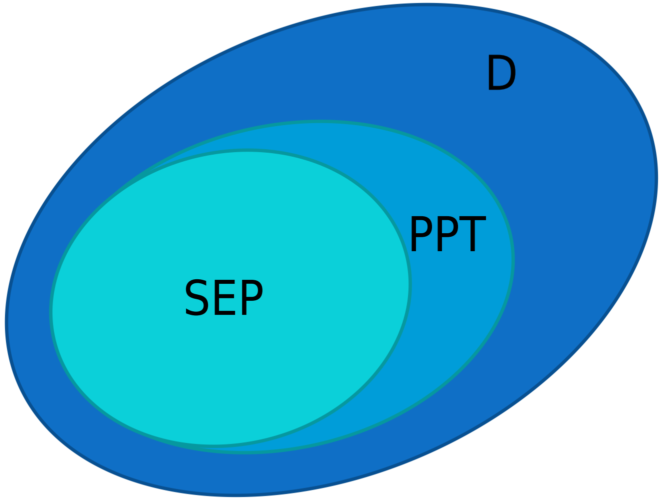
$$\mathrm{PPT} := \{\rho_{AB} \mid \mathrm{Tr}[\rho] = 1,\ \rho \succeq 0,\ \rho^{T_B} \succeq 0\}$$

Can check with SDP! Easy :)

Entanglement witnessing II

For fixed ρ_{AB} consider the SDP

$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \rho_{AB}^{T_B} - \lambda \mathbb{I} \succeq 0 \end{aligned}$$

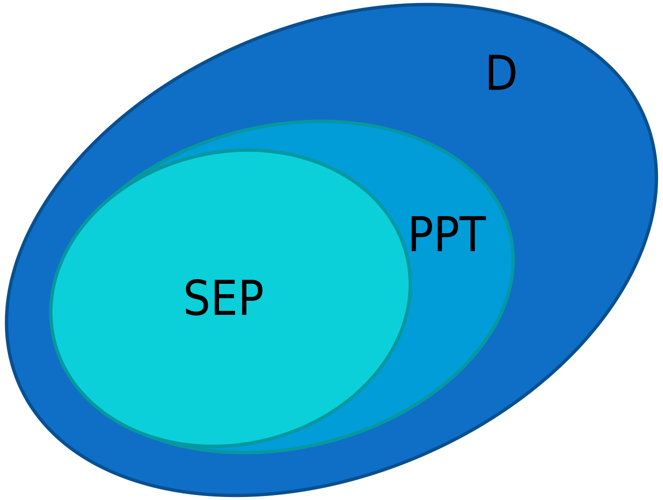


Entanglement witnessing II

For fixed ρ_{AB} consider the SDP

$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \rho_{AB}^{T_B} - \lambda \mathbb{I} \succeq 0 \end{aligned}$$

Computes smallest eigenvalue of $\rho_{AB}^{T_B}$
Exercise: prove this

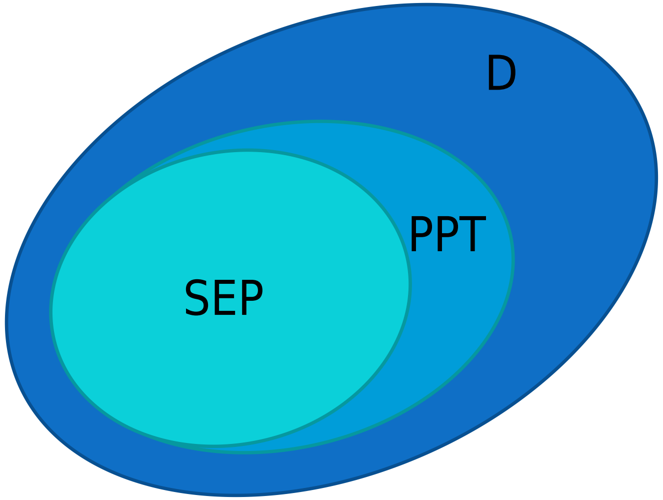


Entanglement witnessing II

For fixed ρ_{AB} consider the SDP

$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \rho_{AB}^{T_B} - \lambda \mathbb{I} \succeq 0 \end{aligned}$$

Computes smallest eigenvalue of $\rho_{AB}^{T_B}$
Exercise: prove this

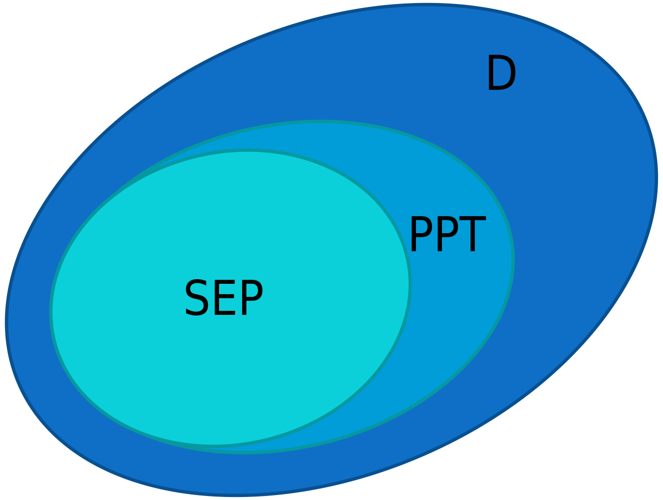


We can use this SDP to quantify entanglement

$$\lambda^* < 0 \quad \Longleftrightarrow \quad \rho_{AB}^{T_B} \not\succeq 0 \quad \Longrightarrow \quad \rho_{AB} \text{ is entangled.}$$

Entanglement witnessing II

For fixed ρ_{AB} consider the SDP



$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \rho_{AB}^{T_B} - \lambda \mathbb{I} \succeq 0 \end{aligned}$$

Computes smallest eigenvalue of $\rho_{AB}^{T_B}$
Exercise: prove this

We can use this SDP to quantify entanglement

$$\lambda^* < 0 \quad \Longleftrightarrow \quad \rho_{AB}^{T_B} \not\succeq 0 \quad \Longrightarrow \quad \rho_{AB} \text{ is entangled.}$$

Dual SDP is

$$\begin{aligned} \min \quad & \mathrm{Tr}[W \rho_{AB}] \\ \text{s.t.} \quad & \mathrm{Tr}[W] = 1 \\ & W^{T_B} \succeq 0 \end{aligned}$$

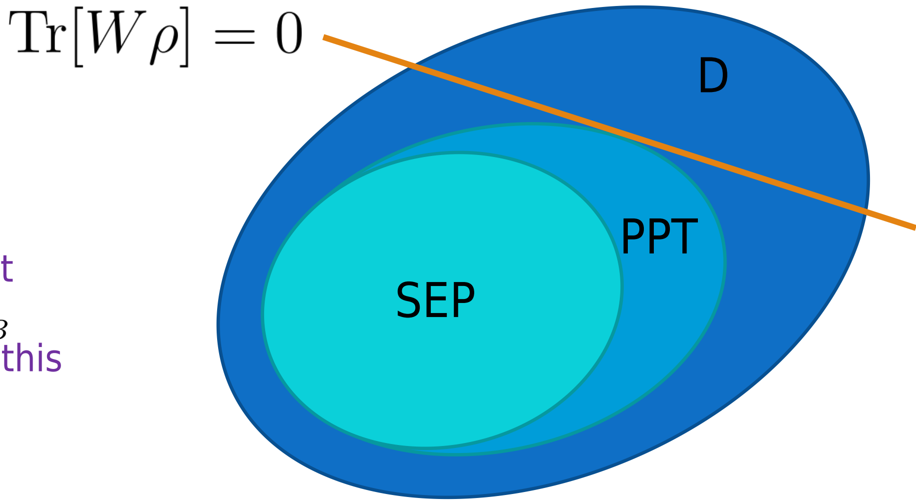
Strong duality holds (why?)

Entanglement witnessing II

For fixed ρ_{AB} consider the SDP

$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \rho_{AB}^{T_B} - \lambda \mathbb{I} \succeq 0 \end{aligned}$$

Computes smallest eigenvalue of $\rho_{AB}^{T_B}$
Exercise: prove this



We can use this SDP to quantify entanglement

$$\lambda^* < 0 \quad \Longleftrightarrow \quad \rho_{AB}^{T_B} \not\succeq 0 \quad \Longrightarrow \quad \rho_{AB} \text{ is entangled.}$$

Dual SDP is

$$\begin{aligned} \min \quad & \mathrm{Tr}[W\rho_{AB}] \\ \text{s.t.} \quad & \mathrm{Tr}[W] = 1 \\ & W^{T_B} \succeq 0 \end{aligned}$$

Strong duality holds (why?)

Dual gives experimental procedure!

- For any feasible W, by weak duality of SDPs

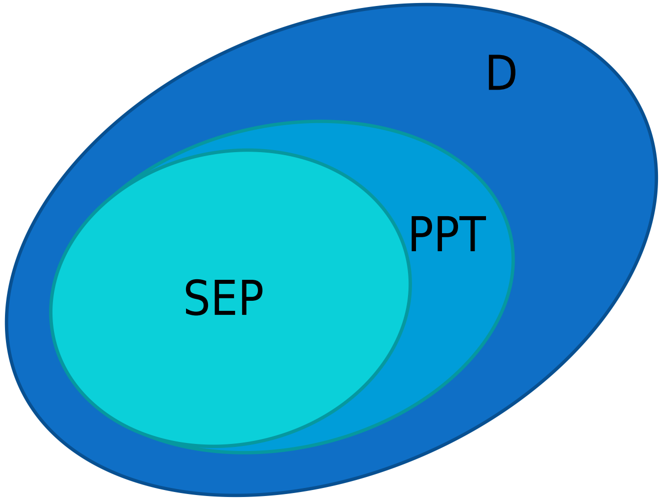
$$\mathrm{Tr}[W\sigma_{AB}] < 0 \implies \sigma_{AB} \text{ is entangled}$$

- And W is an observable, can measure in the lab!

Entanglement witnessing III

Example

Consider $|\psi_\theta\rangle = \cos(\theta)|00\rangle + \sin(\theta)|11\rangle$ for $\theta \in (0, \pi/4]$



$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \rho_{AB}^{T_B} - \lambda \mathbb{I} \succeq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & \mathrm{Tr}[W \rho_{AB}] \\ \text{s.t.} \quad & \mathrm{Tr}[W] = 1 \\ & W^{T_B} \succeq 0 \end{aligned}$$

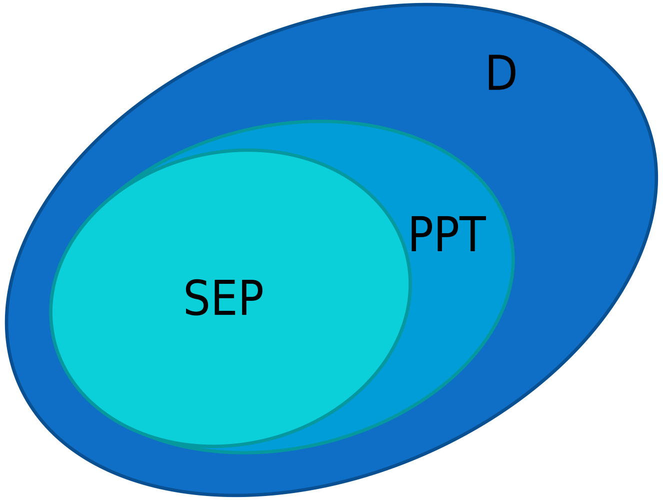
Entanglement witnessing III

Example

Consider $|\psi_\theta\rangle = \cos(\theta)|00\rangle + \sin(\theta)|11\rangle$ for $\theta \in (0, \pi/4]$

then

$$\rho_\theta^{T_B} - \lambda \mathbb{I} = \begin{pmatrix} \cos(\theta)^2 - \lambda & 0 & 0 & 0 \\ 0 & -\lambda & \cos(\theta)\sin(\theta) & 0 \\ 0 & \cos(\theta)\sin(\theta) & -\lambda & 0 \\ 0 & 0 & 0 & \sin(\theta)^2 - \lambda \end{pmatrix}$$



$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \rho_{AB}^{T_B} - \lambda \mathbb{I} \succeq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & \mathrm{Tr}[W \rho_{AB}] \\ \text{s.t.} \quad & \mathrm{Tr}[W] = 1 \\ & W^{T_B} \succeq 0 \end{aligned}$$

Entanglement witnessing III

Example

Consider $|\psi_\theta\rangle = \cos(\theta)|00\rangle + \sin(\theta)|11\rangle$ for $\theta \in (0, \pi/4]$

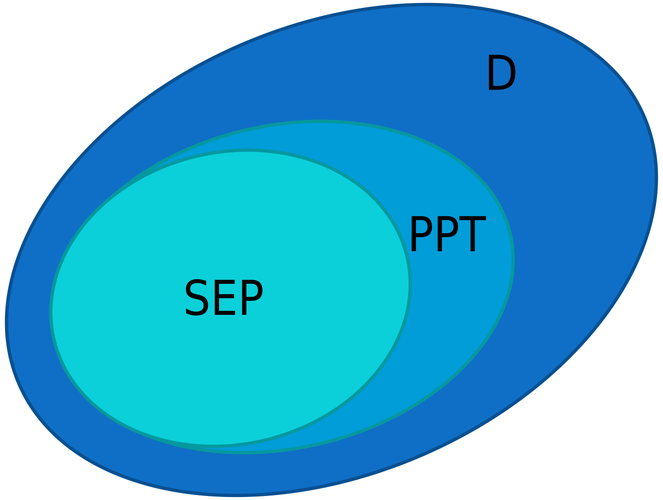
then

$$\rho_\theta^{T_B} - \lambda\mathbb{I} = \begin{pmatrix} \cos(\theta)^2 - \lambda & 0 & 0 & 0 \\ 0 & -\lambda & \cos(\theta)\sin(\theta) & 0 \\ 0 & \cos(\theta)\sin(\theta) & -\lambda & 0 \\ 0 & 0 & 0 & \sin(\theta)^2 - \lambda \end{pmatrix}$$

is PSD iff

$$\lambda \leq -\frac{1}{2}\sin(2\theta) < 0$$

θ **quantifies entanglement (as expected)**



$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \rho_{AB}^{T_B} - \lambda\mathbb{I} \succeq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & \mathrm{Tr}[W\rho_{AB}] \\ \text{s.t.} \quad & \mathrm{Tr}[W] = 1 \\ & W^{T_B} \succeq 0 \end{aligned}$$

Entanglement witnessing III

Example

Consider $|\psi_\theta\rangle = \cos(\theta)|00\rangle + \sin(\theta)|11\rangle$ for $\theta \in (0, \pi/4]$

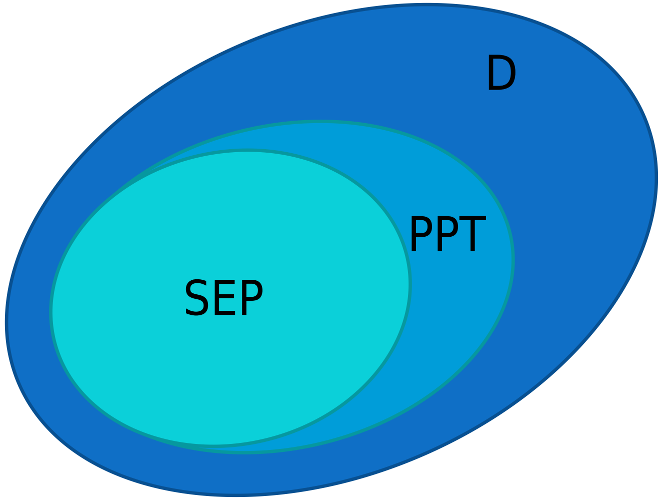
then

$$\rho_\theta^{T_B} - \lambda \mathbb{I} = \begin{pmatrix} \cos(\theta)^2 - \lambda & 0 & 0 & 0 \\ 0 & -\lambda & \cos(\theta)\sin(\theta) & 0 \\ 0 & \cos(\theta)\sin(\theta) & -\lambda & 0 \\ 0 & 0 & 0 & \sin(\theta)^2 - \lambda \end{pmatrix}$$

is PSD iff

$$\lambda \leq -\frac{1}{2}\sin(2\theta) < 0$$

θ **quantifies entanglement (as expected)**



$$\begin{aligned} \max \quad & \lambda \\ \text{s.t.} \quad & \rho_{AB}^{T_B} - \lambda \mathbb{I} \succeq 0 \end{aligned}$$

From the dual we get

$$W^* = \begin{pmatrix} 0 & 0 & 0 & -1/2 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ -1/2 & 0 & 0 & 0 \end{pmatrix}$$

Can verify entanglement in the lab with this!

$$\begin{aligned} \min \quad & \mathrm{Tr}[W \rho_{AB}] \\ \text{s.t.} \quad & \mathrm{Tr}[W] = 1 \\ & W^{T_B} \succeq 0 \end{aligned}$$

Quantum money I

The first quantum cryptography protocol was *quantum money* [Wiesner 1983].

Quantum money I

The first quantum cryptography protocol was *quantum money* [Wiesner 1983].

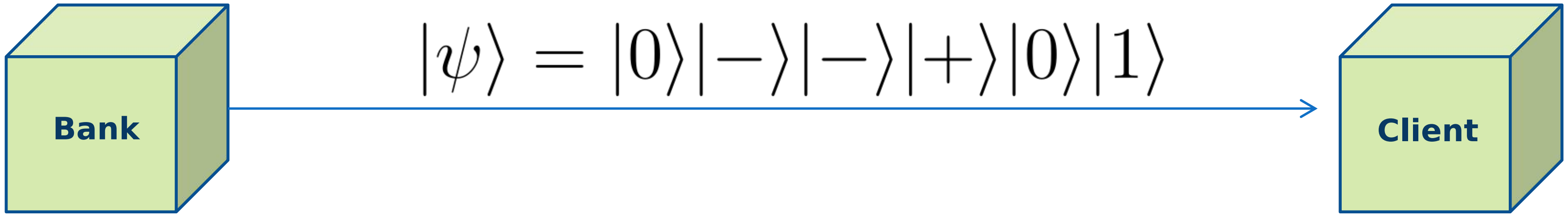
Bank prepares a quantum bank note preparing randomly n qubits chosen from states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$

Quantum money I

The first quantum cryptography protocol was *quantum money* [Wiesner 1983].

Bank prepares a quantum bank note preparing randomly n qubits chosen from states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$

$$|\psi\rangle = |0\rangle|-\rangle|-\rangle|+\rangle|0\rangle|1\rangle$$

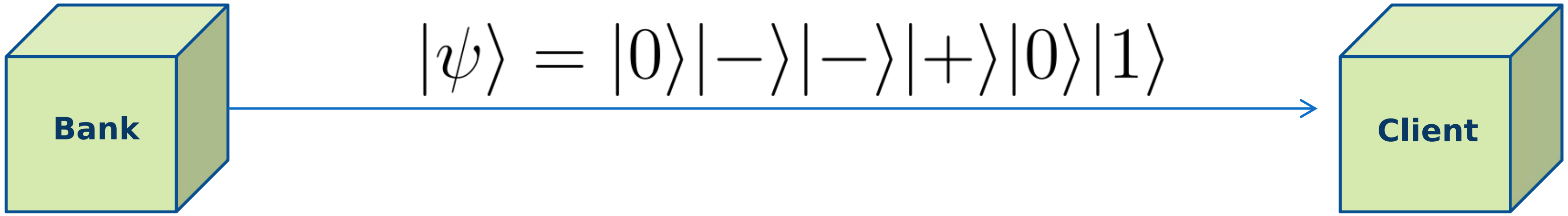


Client doesn't know the state

Quantum money I

The first quantum cryptography protocol was *quantum money* [Wiesner 1983].

Bank prepares a quantum bank note preparing randomly n qubits chosen from states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$



$$|\psi\rangle = |0\rangle|-\rangle|-\rangle|+\rangle|0\rangle|1\rangle$$

Client doesn’t know the state

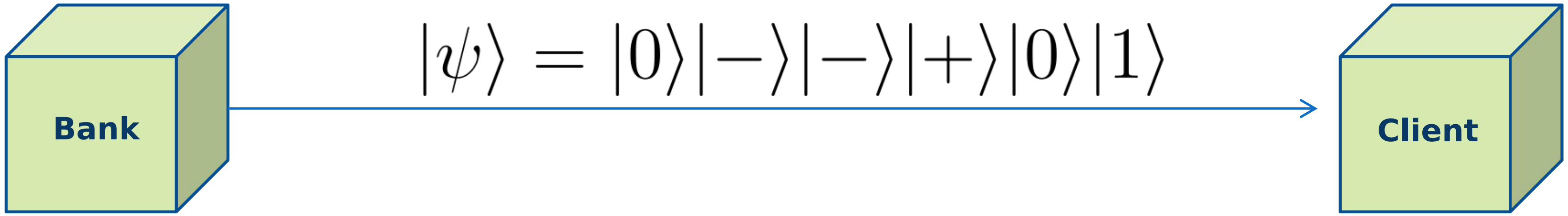
If client wants to “cash in”, they send state back to the bank.

Quantum money I

The first quantum cryptography protocol was *quantum money* [Wiesner 1983].

Bank prepares a quantum bank note preparing randomly n qubits chosen from states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$

$$|\psi\rangle = |0\rangle|-\rangle|-\rangle|+\rangle|0\rangle|1\rangle$$



Client doesn’t know the state

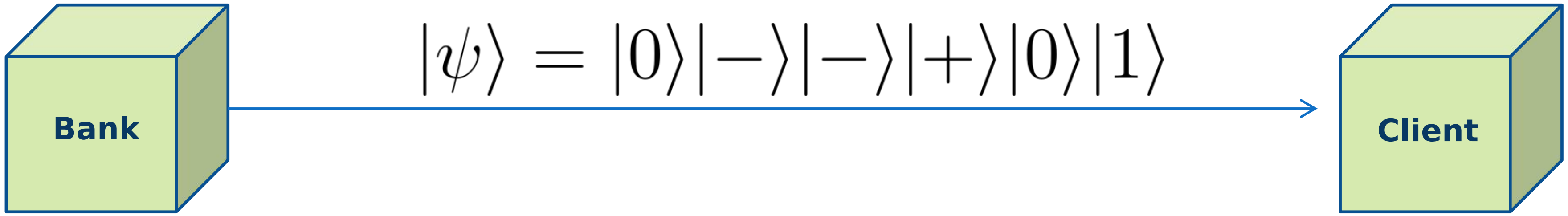
If client wants to “cash in”, they send state back to the bank.

Bank measures in correct bases to check state is valid **(Bank knows which bases were prepared)**

Quantum money I

The first quantum cryptography protocol was *quantum money* [Wiesner 1983].

Bank prepares a quantum bank note preparing randomly n qubits chosen from states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$



Client doesn't know the state

If client wants to "cash in", they send state back to the bank.

Bank measures in correct bases to check state is valid **(Bank knows which bases were prepared)**

Security intuition: If user tries to learn $|\psi\rangle$ they disturb it which is detected by bank's check.

(No-cloning / measurement disturbance -- same as BB84)

Quantum money II - no-cloning

Intuition: quantum money is secure because of no-cloning

Theorem (No cloning)

$\nexists$ a quantum channel $\mathcal{E}$ such that for all states $|\psi\rangle$

$$\mathcal{E}(|\psi\rangle\langle\psi|) = |\psi\rangle\langle\psi| \otimes |\psi\rangle\langle\psi|$$

Quantum money II - no-cloning

Intuition: quantum money is secure because of no-cloning

Theorem (No cloning)

$\nexists$ a quantum channel $\mathcal{E}$ such that for all states $|\psi\rangle$

$$\mathcal{E}(|\psi\rangle\langle\psi|) = |\psi\rangle\langle\psi| \otimes |\psi\rangle\langle\psi|$$

Wait... this is not quantitative. Maybe

Quantum money II - no-cloning

Intuition: quantum money is secure because of no-cloning

Theorem (No cloning)

$\nexists$ a quantum channel $\mathcal{E}$ such that for all states $|\psi\rangle$

$$\mathcal{E}(|\psi\rangle\langle\psi|) = |\psi\rangle\langle\psi| \otimes |\psi\rangle\langle\psi|$$

Wait... this is not quantitative. Maybe

Conjecture (Some cloning?)

$\exists$ a quantum channel $\mathcal{E}$ such that for all states $|\psi\rangle$

$$\mathcal{E}(|\psi\rangle\langle\psi|) \approx |\psi\rangle\langle\psi| \otimes |\psi\rangle\langle\psi|$$

The some cloning conjecture could make the security of quantum money very impractical.

Quantum money III - quick aside on channels

A *quantum channel* $\mathcal{E} : L(A) \to L(B)$ is a completely positive, trace preserving (CPTP) linear map.

(Sends density matrices to density matrices)

Quantum money III - quick aside on channels

A *quantum channel* $\mathcal{E} : L(A) \to L(B)$ is a completely positive, trace preserving (CPTP) linear map.

(Sends density matrices to density matrices)

<u>Lemma (Choi matrices)</u>

A linear map $\mathcal{E} : L(A) \to L(B)$ is a quantum channel iff the *Choi matrix*

$$C_{AB}^{\mathcal{E}} := \sum_{ij} |i\rangle\langle j| \otimes \mathcal{E}(|i\rangle\langle j|)$$

satisfies $C_{AB}^{\mathcal{E}} \succeq 0$ and $\mathrm{Tr}_B[C_{AB}^{\mathcal{E}}] = \mathbb{I}_A$

TL;DR - Channels are PSD matrices in disguise!

Quantum money III - quick aside on channels

A *quantum channel* $\mathcal{E} : L(A) \to L(B)$ is a completely positive, trace preserving (CPTP) linear map.

(Sends density matrices to density matrices)

Lemma (Choi matrices)
A linear map $\mathcal{E} : L(A) \to L(B)$ is a quantum channel iff the *Choi matrix*

$$C^{\mathcal{E}}_{AB} := \sum_{ij} |i\rangle\langle j| \otimes \mathcal{E}(|i\rangle\langle j|)$$

satisfies $C^{\mathcal{E}}_{AB} \succeq 0$ and $\mathrm{Tr}_B[C^{\mathcal{E}}_{AB}] = \mathbb{I}_A$

TL;DR - Channels are PSD matrices in disguise!

We can also recover channel action from Choi matrix

$$\mathcal{E}(\rho) = \mathrm{Tr}_A\left[(\rho^T \otimes \mathbb{I}_B) C_{AB}\right]$$

Importantly, linear in Choi matrix!

Quantum money IV - cloning task

Cloning game

I send you one of the states $\{|\psi_x\rangle\}_x$ with probability $p(x)$. You must design a channel that clones the states with largest average fidelity

$$\sum_x p(x)\langle\psi_x|^{\otimes 2}\mathcal{E}(|\psi_x\rangle\langle\psi_x|)|\psi_x\rangle^{\otimes 2}$$

Equals 1 if perfect cloning

Quantum money IV - cloning task

Cloning game

I send you one of the states $\{|\psi_x\rangle\}_x$ with probability $p(x)$. You must design a channel that clones the states with largest average fidelity

$$\sum_x p(x) \langle\psi_x|^{\otimes 2} \mathcal{E}(|\psi_x\rangle\langle\psi_x|) |\psi_x\rangle^{\otimes 2}$$

Equals 1 if perfect cloning

Example

I send $|0\rangle$ or $|1\rangle$ with uniform probability.

Quantum money IV - cloning task

Cloning game

I send you one of the states $\{|\psi_x\rangle\}_x$ with probability $p(x)$. You must design a channel that clones the states with largest average fidelity

$$\sum_x p(x) \langle \psi_x|^{\otimes 2} \mathcal{E}(|\psi_x\rangle\langle\psi_x|) |\psi_x\rangle^{\otimes 2}$$

Equals 1 if perfect cloning

Example

I send $|0\rangle$ or $|1\rangle$ with uniform probability. Define the 1 -> 2 qubit channel

$$\mathcal{E}(\rho) = V\rho V^\dagger \quad \text{where } V = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

Exercise: Use Choi lemma to show it is quantum channel

Then $\mathcal{E}(|0\rangle\langle 0|) = |0\rangle\langle 0| \otimes |0\rangle\langle 0|$ and $\mathcal{E}(|1\rangle\langle 1|) = |1\rangle\langle 1| \otimes |1\rangle\langle 1|$

Perfect cloning! (but not surprising - perfectly distinguishable)

Quantum money V - optimal cloner

$\mathcal{E} : L(A_1) \to L(A_2 A_3)$

The optimal cloner can be found using the SDP

Complex conjugate

$$
\begin{aligned}
\max \quad & \sum_x p(x) \left(\langle \overline{\psi_x} | \otimes \langle \psi_x | \otimes \langle \psi_x | \right) C_{A_1 A_2 A_3} \left(| \overline{\psi_x} \rangle \otimes | \psi_x \rangle \otimes | \psi_x \rangle \right) \\
\text{s.t.} \quad & \mathrm{Tr}_{A_2 A_3} [C_{A_1 A_2 A_3}] = \mathbb{I}_{A_1} \\
& C_{A_1 A_2 A_3} \succeq 0
\end{aligned}
$$

SDP returns optimal fidelity AND the best channel!

Quantum money V - optimal cloner

$\mathcal{E} : L(A_1) \to L(A_2 A_3)$

The optimal cloner can be found using the SDP

Complex conjugate

$$\max \quad \sum_x p(x) \left(\langle \overline{\psi_x} | \otimes \langle \psi_x | \otimes \langle \psi_x | \right) C_{A_1 A_2 A_3} \left(| \overline{\psi_x} \rangle \otimes | \psi_x \rangle \otimes | \psi_x \rangle \right)$$

$$\text{s.t.} \quad \mathrm{Tr}_{A_2 A_3} [C_{A_1 A_2 A_3}] = \mathbb{I}_{A_1}$$

$$C_{A_1 A_2 A_3} \succeq 0$$

SDP returns optimal fidelity AND the best channel!

<u>Example (Quantum money)</u>

I send $|0\rangle, |1\rangle, |+\rangle, |-\rangle$ with uniform probability.

[Molina, Vidick, Watrous, *Optimal Counterfeiting Attacks and Generalizations for Wiesner's Quantum Money* (2012)]

Quantum money V - optimal cloner

$\mathcal{E} : L(A_1) \to L(A_2 A_3)$

The optimal cloner can be found using the SDP

Complex conjugate

$$\max \quad \sum_x p(x) \left(\langle\overline{\psi_x}| \otimes \langle\psi_x| \otimes \langle\psi_x|\right) C_{A_1 A_2 A_3} \left(|\overline{\psi_x}\rangle \otimes |\psi_x\rangle \otimes |\psi_x\rangle\right)$$

$$\text{s.t.} \quad \mathrm{Tr}_{A_2 A_3}[C_{A_1 A_2 A_3}] = \mathbb{I}_{A_1}$$

$$C_{A_1 A_2 A_3} \succeq 0$$

SDP returns optimal fidelity AND the best channel!

Example (Quantum money)

I send $|0\rangle, |1\rangle, |+\rangle, |-\rangle$ with uniform probability. The optimal quantum channel is

$$\mathcal{E}(\rho) = K_0 \rho K_0^\dagger + K_1 \rho K_1^\dagger \qquad K_0 = \frac{1}{\sqrt{12}} \begin{pmatrix} 3 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \quad K_1 = \frac{1}{\sqrt{12}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 3 \end{pmatrix}$$

achieves average fidelity 3/4.

Quantum money V - optimal cloner

$\mathcal{E} : L(A_1) \to L(A_2 A_3)$

The optimal cloner can be found using the SDP

Complex conjugate

$$\max \quad \sum_x p(x) \left(\langle \overline{\psi_x} | \otimes \langle \psi_x | \otimes \langle \psi_x | \right) C_{A_1 A_2 A_3} \left(| \overline{\psi_x} \rangle \otimes | \psi_x \rangle \otimes | \psi_x \rangle \right)$$

$$\text{s.t.} \quad \mathrm{Tr}_{A_2 A_3}[C_{A_1 A_2 A_3}] = \mathbb{I}_{A_1}$$

$$C_{A_1 A_2 A_3} \succeq 0$$

SDP returns optimal fidelity AND the best channel!

Example (Quantum money)

I send $|0\rangle, |1\rangle, |+\rangle, |-\rangle$ with uniform probability. The optimal quantum channel is

$$\mathcal{E}(\rho) = K_0 \rho K_0^\dagger + K_1 \rho K_1^\dagger$$

$$K_0 = \frac{1}{\sqrt{12}} \begin{pmatrix} 3 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \quad K_1 = \frac{1}{\sqrt{12}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 3 \end{pmatrix}$$

achieves average fidelity 3/4.

Can use SDP to prove n-copy cloning scales as $(3/4)^n$

Exercise: Find 4 qubit states that are harder to clone than the Wiesner conjugate coding states.

PICOS

Convex.jl

YALMIP

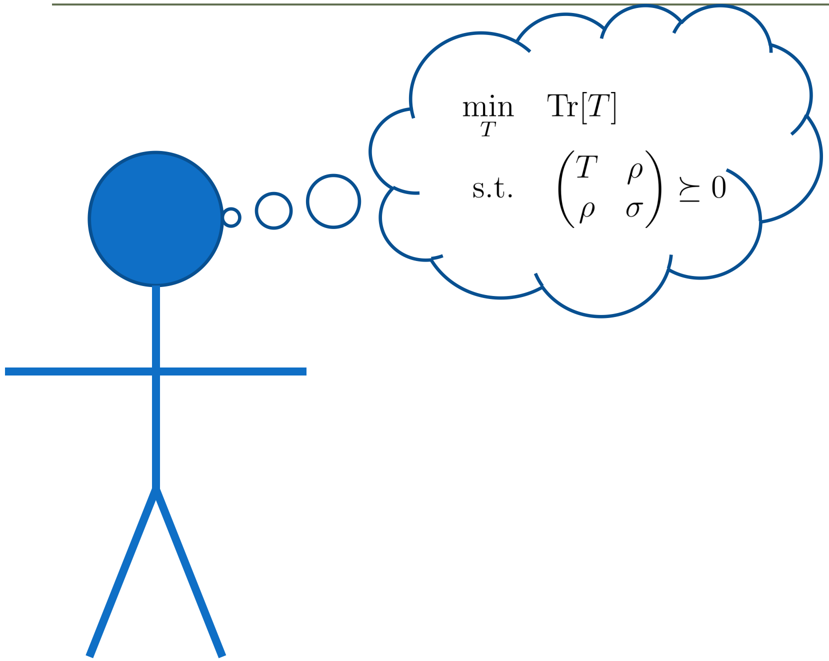
The how

SCS

SPLITTING CONIC SOLVER

mosek

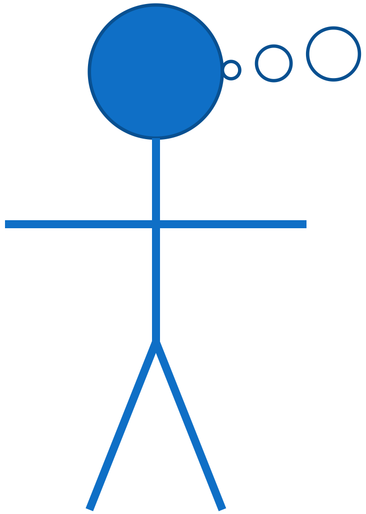
Solving SDPs in practice



$$\min_{T} \quad \mathrm{Tr}[T] \qquad \text{s.t.} \quad \begin{pmatrix} T & \rho \\ \rho & \sigma \end{pmatrix} \succeq 0$$

You

Solving SDPs in practice



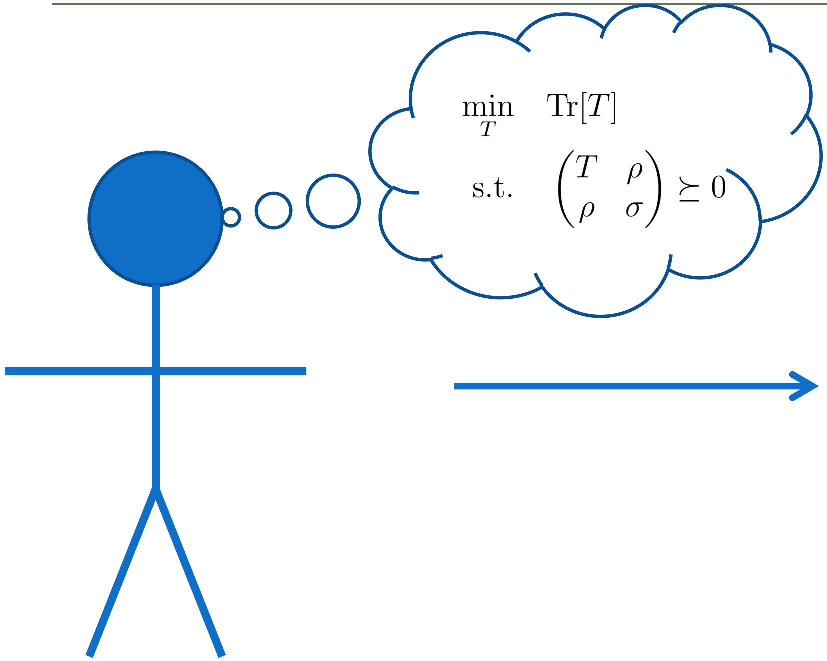
$$\min_{T} \quad \mathrm{Tr}[T] \qquad \text{s.t.} \quad \begin{pmatrix} T & \rho \\ \rho & \sigma \end{pmatrix} \succeq 0$$

```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho = pc.Constant([[1,0],[0,0]])
sigma = pc.Constant([[1/2,1/4],[1/4,1/2]])
T = pc.HermitianVariable("T", (2, 2))

sdp.add_constraint( ((T & rho) // (rho & sigma)) >> 0 )
sdp.set_objective("min", pc.trace(T))

sdp.solve()
print(sdp.value)
```

You

High-level modeller

Solving SDPs in practice

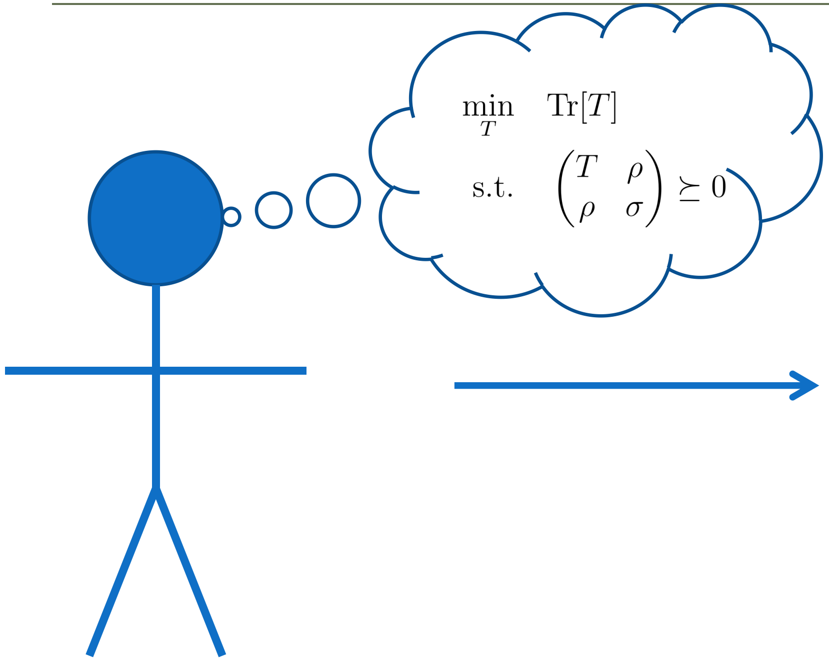


$$\min_{T} \quad \mathrm{Tr}[T] \qquad \text{s.t.} \quad \begin{pmatrix} T & \rho \\ \rho & \sigma \end{pmatrix} \succeq 0$$

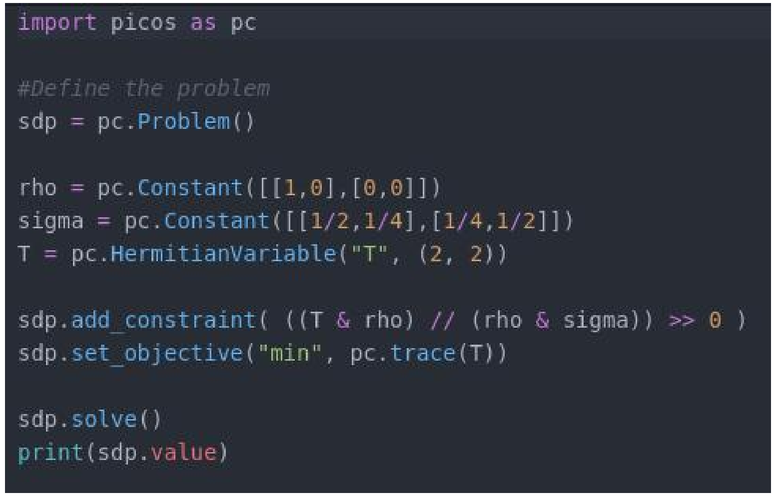
```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho = pc.Constant([[1,0],[0,0]])
sigma = pc.Constant([[1/2,1/4],[1/4,1/2]])
T = pc.HermitianVariable("T", (2, 2))

sdp.add_constraint( ((T & rho) // (rho & sigma)) >> 0 )
sdp.set_objective("min", pc.trace(T))

sdp.solve()
print(sdp.value)
```

You | **High-level modeller** | **Solver**

Solving SDPs in practice



You



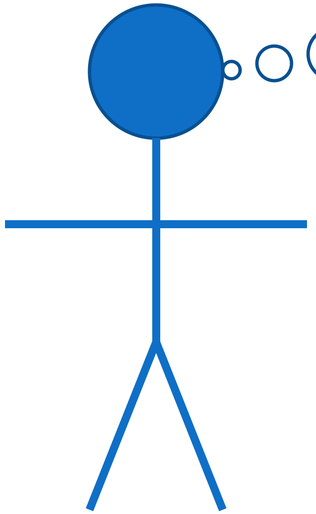
```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho = pc.Constant([[1,0],[0,0]])
sigma = pc.Constant([[1/2,1/4],[1/4,1/2]])
T = pc.HermitianVariable("T", (2, 2))

sdp.add_constraint( ((T & rho) // (rho & sigma)) >> 0 )
sdp.set_objective("min", pc.trace(T))

sdp.solve()
print(sdp.value)
```

High-level modeller

Solver

Solving SDPs in practice

$$\min_T \quad \mathrm{Tr}[T] \qquad \text{s.t.} \quad \begin{pmatrix} T & \rho \\ \rho & \sigma \end{pmatrix} \succeq 0$$



You

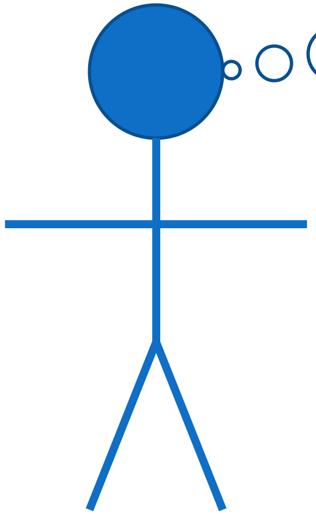
```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho = pc.Constant([[1,0],[0,0]])
sigma = pc.Constant([[1/2,1/4],[1/4,1/2]])
T = pc.HermitianVariable("T", (2, 2))

sdp.add_constraint( ((T & rho) // (rho & sigma)) >> 0 )
sdp.set_objective("min", pc.trace(T))

sdp.solve()
print(sdp.value)
```

High-level modeller

Solver

Solving SDPs in practice

$$\min_{T} \quad \mathrm{Tr}[T]$$
$$\text{s.t.} \quad \begin{pmatrix} T & \rho \\ \rho & \sigma \end{pmatrix} \succeq 0$$



```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho = pc.Constant([[1,0],[0,0]])
sigma = pc.Constant([[1/2,1/4],[1/4,1/2]])
T = pc.HermitianVariable("T", (2, 2))

sdp.add_constraint( ((T & rho) // (rho & sigma)) >> 0 )
sdp.set_objective("min", pc.trace(T))

sdp.solve()
print(sdp.value)
```

You **High-level modeller** **Solver**

It’s easy to SDP in practice :)

High-level modelling

High level modelling allows us to code SDPs like we’re writing the mathematics.

High-level modelling

High level modelling allows us to code SDPs like we're writing the mathematics.

A very non-exhaustive list...

Programming language	Available modellers
python	PICOS / CVXPY / PyLMI-SDP
matlab	YALMIP / CVX
julia	Convex.jl
mathematica	built-in

High-level modelling

Example in PICOS (python)

```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho0 = pc.Constant([[1,0],[0,0]])
rho1 = pc.Constant([[1/2,1/2],[1/2,1/2]])
rho2 = pc.Constant([[1/2,-1/2],[-1/2,1/2]])

M0 = pc.HermitianVariable("M0", (2, 2))
M1 = pc.HermitianVariable("M1", (2, 2))
M2 = pc.HermitianVariable("M2", (2, 2))

sdp.add_constraint( M0 + M1 + M2 == [[1,0],[0,1]] )
sdp.add_constraint( M0 >> 0 )
sdp.add_constraint( M1 >> 0 )
sdp.add_constraint( M2 >> 0 )

obj = pc.trace(0.5 * M0 * rho0 + 0.25 * M1 * rho1 + 0.25 * M2 * rho2)
sdp.set_objective("max", obj)

sdp.solve()
```

High-level modelling

Example in PICOS (python)

```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho0 = pc.Constant([[1,0],[0,0]])
rho1 = pc.Constant([[1/2,1/2],[1/2,1/2]])
rho2 = pc.Constant([[1/2,-1/2],[-1/2,1/2]])

M0 = pc.HermitianVariable("M0", (2, 2))
M1 = pc.HermitianVariable("M1", (2, 2))
M2 = pc.HermitianVariable("M2", (2, 2))

sdp.add_constraint( M0 + M1 + M2 == [[1,0],[0,1]] )
sdp.add_constraint( M0 >> 0 )
sdp.add_constraint( M1 >> 0 )
sdp.add_constraint( M2 >> 0 )

obj = pc.trace(0.5 * M0 * rho0 + 0.25 * M1 * rho1 + 0.25 * M2 * rho2)
sdp.set_objective("max", obj)

sdp.solve()
```

Defining constants

High-level modelling

Example in PICOS (python)

```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho0 = pc.Constant([[1,0],[0,0]])
rho1 = pc.Constant([[1/2,1/2],[1/2,1/2]])
rho2 = pc.Constant([[1/2,-1/2],[-1/2,1/2]])

M0 = pc.HermitianVariable("M0", (2, 2))
M1 = pc.HermitianVariable("M1", (2, 2))
M2 = pc.HermitianVariable("M2", (2, 2))

sdp.add_constraint( M0 + M1 + M2 == [[1,0],[0,1]] )
sdp.add_constraint( M0 >> 0 )
sdp.add_constraint( M1 >> 0 )
sdp.add_constraint( M2 >> 0 )

obj = pc.trace(0.5 * M0 * rho0 + 0.25 * M1 * rho1 + 0.25 * M2 * rho2)
sdp.set_objective("max", obj)

sdp.solve()
```

Defining constants

Defining variables

High-level modelling

Example in PICOS (python)

```
import picos as pc

#Define the problem
sdp = pc.Problem()

rho0 = pc.Constant([[1,0],[0,0]])
rho1 = pc.Constant([[1/2,1/2],[1/2,1/2]])
rho2 = pc.Constant([[1/2,-1/2],[-1/2,1/2]])

M0 = pc.HermitianVariable("M0", (2, 2))
M1 = pc.HermitianVariable("M1", (2, 2))
M2 = pc.HermitianVariable("M2", (2, 2))

sdp.add_constraint( M0 + M1 + M2 == [[1,0],[0,1]] )
sdp.add_constraint( M0 >> 0 )
sdp.add_constraint( M1 >> 0 )
sdp.add_constraint( M2 >> 0 )

obj = pc.trace(0.5 * M0 * rho0 + 0.25 * M1 * rho1 + 0.25 * M2 * rho2)
sdp.set_objective("max", obj)

sdp.solve()
```

Defining constants

Defining variables

POVM constraints

High-level modelling

Example in PICOS (python)

```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho0 = pc.Constant([[1,0],[0,0]])
rho1 = pc.Constant([[1/2,1/2],[1/2,1/2]])
rho2 = pc.Constant([[1/2,-1/2],[-1/2,1/2]])

M0 = pc.HermitianVariable("M0", (2, 2))
M1 = pc.HermitianVariable("M1", (2, 2))
M2 = pc.HermitianVariable("M2", (2, 2))

sdp.add_constraint( M0 + M1 + M2 == [[1,0],[0,1]] )
sdp.add_constraint( M0 >> 0 )
sdp.add_constraint( M1 >> 0 )
sdp.add_constraint( M2 >> 0 )

obj = pc.trace(0.5 * M0 * rho0 + 0.25 * M1 * rho1 + 0.25 * M2 * rho2)
sdp.set_objective("max", obj)

sdp.solve()
```

Defining constants

Defining variables

POVM constraints

Objective function

High-level modelling

Example in PICOS (python)

```python
import picos as pc

#Define the problem
sdp = pc.Problem()

rho0 = pc.Constant([[1,0],[0,0]])
rho1 = pc.Constant([[1/2,1/2],[1/2,1/2]])
rho2 = pc.Constant([[1/2,-1/2],[-1/2,1/2]])

M0 = pc.HermitianVariable("M0", (2, 2))
M1 = pc.HermitianVariable("M1", (2, 2))
M2 = pc.HermitianVariable("M2", (2, 2))

sdp.add_constraint( M0 + M1 + M2 == [[1,0],[0,1]] )
sdp.add_constraint( M0 >> 0 )
sdp.add_constraint( M1 >> 0 )
sdp.add_constraint( M2 >> 0 )

obj = pc.trace(0.5 * M0 * rho0 + 0.25 * M1 * rho1 + 0.25 * M2 * rho2)
sdp.set_objective("max", obj)

sdp.solve()
```

Defining constants

Defining variables

POVM constraints

Objective function

Solve it please!

High-level modelling

Example in PICOS (python)

```
import picos as pc

#Define the problem
sdp = pc.Problem()

rho0 = pc.Constant([[1,0],[0,0]])
rho1 = pc.Constant([[1/2,1/2],[1/2,1/2]])
rho2 = pc.Constant([[1/2,-1/2],[-1/2,1/2]])

M0 = pc.HermitianVariable("M0", (2, 2))
M1 = pc.HermitianVariable("M1", (2, 2))
M2 = pc.HermitianVariable("M2", (2, 2))

sdp.add_constraint( M0 + M1 + M2 == [[1,0],[0,1]] )
sdp.add_constraint( M0 >> 0 )
sdp.add_constraint( M1 >> 0 )
sdp.add_constraint( M2 >> 0 )

obj = pc.trace(0.5 * M0 * rho0 + 0.25 * M1 * rho1 + 0.25 * M2 * rho2)
sdp.set_objective("max", obj)

sdp.solve()
```

To solve other discrimination problems need to modify only a few lines!

Defining constants

Defining variables

POVM constraints

Objective function

Solve it please!

Solvers

Modelling languages interact with various solvers

Available solvers	Completely unbiased opinion
Mosek	Best overall solver
SCS	Fast and inaccurate “My problem is too big for the other solvers”
SDPA (family)	“I really need a lot of precision”
CVXOPT / SeDuMi / SDPT3 / Hypatia / …	…

Solvers

Modelling languages interact with various solvers

Available solvers	Completely unbiased opinion
Mosek	Best overall solver
SCS	Fast and inaccurate “My problem is too big for the other solvers”
SDPA (family)	“I really need a lot of precision”
CVXOPT / SeDuMi / SDPT3 / Hypatia / ...	...

And they’re easy to use...

```
import mosek
sdp.solve(solver='mosek')
```

Further topics / reading

- Surprisingly powerful for applications in mathematical proofs!

Further topics / reading

- Surprisingly powerful for applications in mathematical proofs!

- SDP relaxations

  My problem is not an SDP but can be bounded by one

  NPA hierarchy

  See arXiv:2307.02551 for a review focused on quantum applications

  Lasserre hierarchy

  DPS hierarchy

Further topics / reading

- Surprisingly powerful for applications in mathematical proofs!

- SDP relaxations

  NPA hierarchy

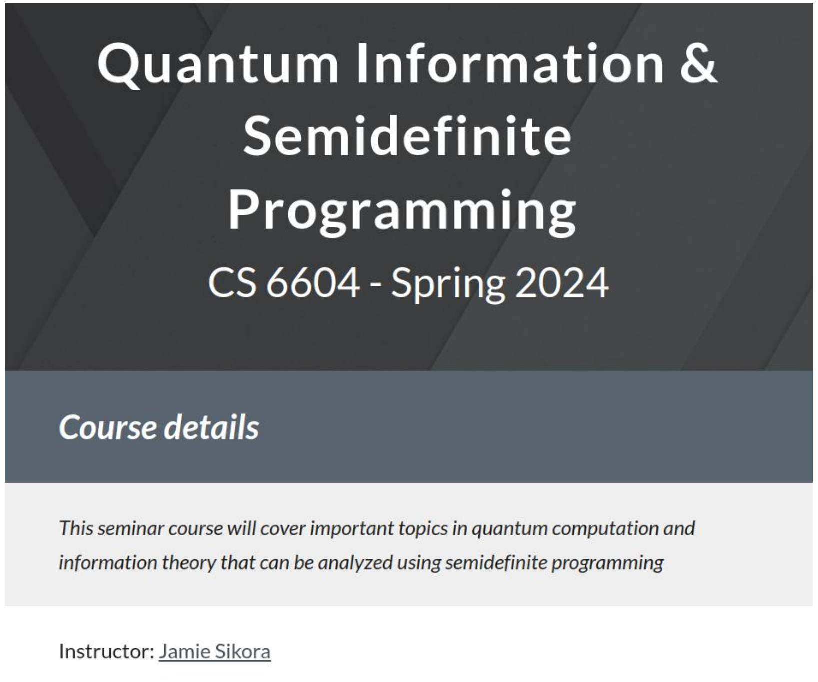
  Lasserre hierarchy

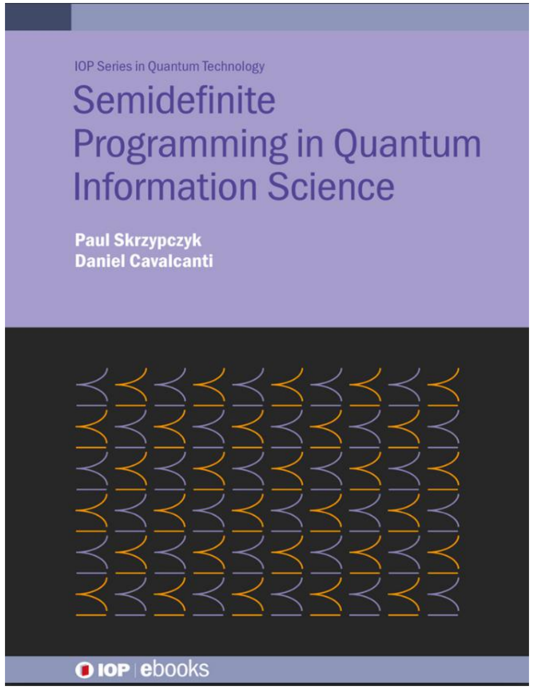
  DPS hierarchy

- Resources

My problem is not an SDP but can be bounded by one

See arXiv:2307.02551 for a review focused on quantum applications





Further topics / reading

- Surprisingly powerful for applications in mathematical proofs!

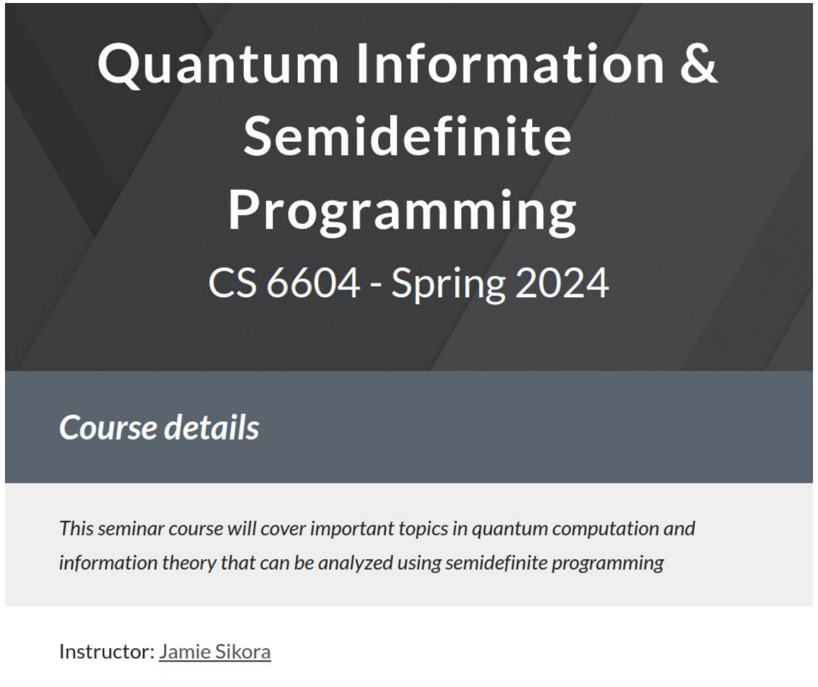
- SDP relaxations — **My problem is not an SDP but can be bounded by one**

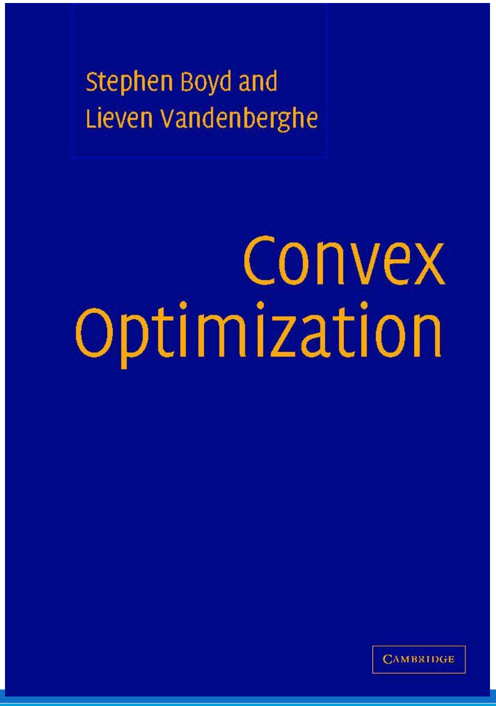
  NPA hierarchy — **See arXiv:2307.02551 for a review focused on quantum applications**

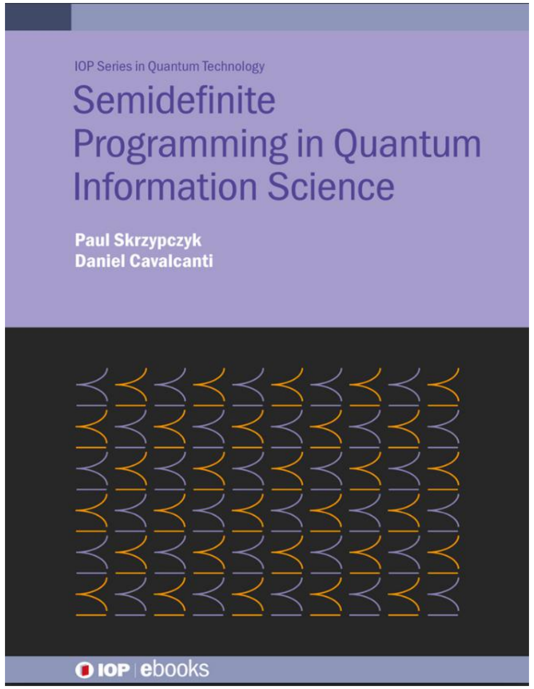
  Lasserre hierarchy

  DPS hierarchy

- Resources







I'll upload example scripts / slides at **github.com/peterjbrown519/QSI-school**

Bonus slides

Duality gap example

$$\begin{aligned} \max \quad & -a-d \\ \text{s.t.} \quad & a=0 \\ & d+2c=1 \\ & \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} \succeq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & \lambda_2 \\ \text{s.t.} \quad & \begin{pmatrix} \lambda_1+1 & 0 & \lambda_2 \\ 0 & \lambda_2+1 & 0 \\ \lambda_2 & 0 & 0 \end{pmatrix} \succeq 0 \end{aligned}$$

Optimal value: -1

Optimal value: 0

SDP standard forms - Rewriting tricks

Some tricks

- min/max $$\min f(x) = -\max -f(x)$$
- Equalities -> Inequalities $$x = y \iff x \leq y \text{ and } -x \leq -y$$
- Inequalities -> Equalities + slack $$x \leq y \iff x = y + s \text{ and } s \geq 0$$
- Hermitian matrices $$X \text{ is Hermitian} \iff X = X_1 - X_2 \text{ and } X_1, X_2 \succeq 0$$
- Multiple PSD constraints $$X \succeq 0 \text{ and } Y \succeq 0 \iff \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} \succeq 0$$

SDP duality IV - A hack to find the dual…

Step 1: Form the *Lagrangian* **(Big function of all the variables + some new ones…)**

$$
\begin{aligned}
\max \quad & f(x) \\
\text{s.t.} \quad & \cdots \\
& \cdots
\end{aligned}
$$

SDP duality IV - A hack to find the dual...

Step 1: Form the *Lagrangian*

Step 1.1: Add the objective

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & \cdots \\ & \cdots \end{aligned} \quad \longrightarrow \quad \mathcal{L} = f(x)$$

Just start Lagrangian with objective

SDP duality IV - A hack to find the dual…

Step 1: Form the Lagrangian

Step 1.2: Add real inequalities

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & \cdots \\ & g_i(x) \leq w_i \\ & \cdots \end{aligned} \longrightarrow \begin{aligned} \mathcal{L} = f(x) + \ldots \\ + \lambda_i(w_i - g_i(x)) \end{aligned}$$

For each real inequality:

1. **Rewrite as positive inequality** $w_i - g_i(x) \geq 0$
2. **Introduce dual variable** $\lambda_i \geq 0$
3. **Add product to Lagrangian** $\lambda_i(w_i - g_i(x))$

SDP duality IV - A hack to find the dual…

Step 1: Form the Lagrangian

Step 1.3: Add real equalities

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & \cdots \\ & h_i(x) = v_i \\ & \cdots \end{aligned} \longrightarrow \begin{aligned} \mathcal{L} = f(x) + \ldots \\ + \mu_i(v_i - h_i(x)) \end{aligned}$$

For each real inequality:

1. **Rewrite as 0 equality** $v_i - h_i(x) = 0$
2. **Introduce dual variable** $\mu_i \in \mathbb{R}$
3. **Add product to Lagrangian** $\mu_i(v_i - h_i(x))$

SDP duality IV - A hack to find the dual…

Step 1: Form the Lagrangian

Step 1.4: Add PSD inequalities

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & \cdots \\ & G_i(x) \preceq W_i \\ & \cdots \end{aligned} \longrightarrow \begin{aligned} \mathcal{L} = & f(x) + \ldots \\ & + \mathrm{Tr}[A_i(W_i - G_i(x))] \end{aligned}$$

For each PSD inequality:

1. **Rewrite as positive inequality** $W_i - G_i(x) \succeq 0$
2. **Introduce dual variable** $A_i \succeq 0$
3. **Add "product" to Lagrangian** $\mathrm{Tr}[A_i(W_i - G_i(x))]$

SDP duality IV - A hack to find the dual…

Step 1: Form the Lagrangian

Step 1.4: Add matrix equalities

$$\begin{aligned} \max \quad & f(x) \\ \text{s.t.} \quad & \cdots \\ & H_i(x) = V_i \\ & \cdots \end{aligned} \longrightarrow \begin{aligned} \mathcal{L} = f(x) + \ldots \\ + \operatorname{Tr}[B_i(V_i - H_i(x))] \end{aligned}$$

For each PSD inequality:

1. **Rewrite as 0 equality** $V_i - H_i(x) = 0$
2. **Introduce dual variable** B_i **(Hermitian)**
3. **Add "product" to Lagrangian** $\operatorname{Tr}[B_i(V_i - H_i(x))]$

SDP duality IV - A hack to find the dual…

Step 1: Form the Lagrangian

Dual variables: $\lambda_i \geq 0$ $\mu_i \in \mathbb{R}$ $A_i \succeq 0$ B_i **(Hermitian)**

$$\begin{aligned}\mathcal{L} = f(x) & \\ &+ \sum_i \lambda_i (w_i - g_i(x)) \\ &+ \sum_i \mu_i (v_i - h_i(x)) \\ &+ \sum_i \mathrm{Tr}[A_i (W_i - G_i(x))] \\ &+ \sum_i \mathrm{Tr}[B_i (V_i - H_i(x))]\end{aligned}$$

SDP duality IV - A hack to find the dual…

Step 1: Form the Lagrangian

$$\mathcal{L} = f(x) + \sum_i \lambda_i (w_i - g_i(x)) + \sum_i \mu_i (v_i - h_i(x)) + \sum_i \mathrm{Tr}[A_i(W_i - G_i(x))] + \sum_i \mathrm{Tr}[B_i(V_i - H_i(x))]$$

Dual variables: $\lambda_i \geq 0$ $\mu_i \in \mathbb{R}$ $A_i \succeq 0$ B_i **(Hermitian)**

We can recover primal constraints by asking:

When is Lagrangian **bounded** if we **minimize** over **dual** variables?

SDP duality IV - A hack to find the dual...

Step 1: Form the Lagrangian

Dual variables: $\lambda_i \geq 0$ $\mu_i \in \mathbb{R}$ $A_i \succeq 0$ B_i **(Hermitian)**

We can recover primal constraints by asking:

When is Lagrangian **bounded** if we **minimize** over **dual** variables?

$$\mathcal{L} = f(x)$$

$$+ \sum_i \lambda_i (w_i - g_i(x)) \xrightarrow[\lambda_i \geq 0]{\min}$$ Not $-\infty$ iff $w_i - g_i(x) \geq 0$

$$+ \sum_i \mu_i (v_i - h_i(x))$$

$$+ \sum_i \mathrm{Tr}[A_i (W_i - G_i(x))]$$

$$+ \sum_i \mathrm{Tr}[B_i (V_i - H_i(x))]$$

SDP duality IV - A hack to find the dual...

Step 1: Form the Lagrangian

Dual variables: $\lambda_i \geq 0$ $\mu_i \in \mathbb{R}$ $A_i \succeq 0$ B_i **(Hermitian)**

We can recover primal constraints by asking:

When is Lagrangian **bounded** if we **minimize** over **dual** variables?

$$\mathcal{L} = f(x)$$

$$+ \sum_i \lambda_i (w_i - g_i(x)) \xrightarrow[\lambda_i \geq 0]{\min} \text{Not } -\infty \text{ iff } w_i - g_i(x) \geq 0$$

$$+ \sum_i \mu_i (v_i - h_i(x)) \xrightarrow[\mu_i \in \mathbb{R}]{\min} \text{Not } -\infty \text{ iff } v_i - h_i(x) = 0$$

$$+ \sum_i \mathrm{Tr}[A_i (W_i - G_i(x))]$$

$$+ \sum_i \mathrm{Tr}[B_i (V_i - H_i(x))]$$

SDP duality IV - A hack to find the dual...

Step 1: Form the Lagrangian

Dual variables: $\lambda_i \geq 0$ $\mu_i \in \mathbb{R}$ $A_i \succeq 0$ B_i **(Hermitian)**

We can recover primal constraints by asking:

When is Lagrangian **bounded** if we **minimize** over **dual** variables?

$$\mathcal{L} = f(x)$$

$$+ \sum_i \lambda_i (w_i - g_i(x)) \xrightarrow[\lambda_i \geq 0]{\min} \text{Not } -\infty \text{ iff } w_i - g_i(x) \geq 0$$

$$+ \sum_i \mu_i (v_i - h_i(x)) \xrightarrow[\mu_i \in \mathbb{R}]{\min} \text{Not } -\infty \text{ iff } v_i - h_i(x) = 0$$

$$+ \sum_i \mathrm{Tr}[A_i(W_i - G_i(x))] \xrightarrow[A_i \succeq 0]{\min} \text{Not } -\infty \text{ iff } W_i - G_i(x) \succeq 0$$

$$+ \sum_i \mathrm{Tr}[B_i(V_i - H_i(x))]$$

SDP duality IV - A hack to find the dual…

Step 1: Form the Lagrangian

Dual variables: $\lambda_i \geq 0$ $\mu_i \in \mathbb{R}$ $A_i \succeq 0$ B_i **(Hermitian)**

We can recover primal constraints by asking:

When is Lagrangian **bounded** if we **minimize** over **dual** variables?

$$\mathcal{L} = f(x)$$

$$+ \sum_i \lambda_i (w_i - g_i(x)) \xrightarrow[\lambda_i \geq 0]{\min}$$ Not $-\infty$ iff $w_i - g_i(x) \geq 0$

$$+ \sum_i \mu_i (v_i - h_i(x)) \xrightarrow[\mu_i \in \mathbb{R}]{\min}$$ Not $-\infty$ iff $v_i - h_i(x) = 0$

$$+ \sum_i \mathrm{Tr}[A_i (W_i - G_i(x))] \xrightarrow[A_i \succeq 0]{\min}$$ Not $-\infty$ iff $W_i - G_i(x) \succeq 0$

$$+ \sum_i \mathrm{Tr}[B_i (V_i - H_i(x))] \xrightarrow[B_i]{\min}$$ Not $-\infty$ iff $V_i - H_i(x) = 0$

SDP duality IV - A hack to find the dual...

Step 1: Form the Lagrangian

Dual variables: $\lambda_i \geq 0$ $\mu_i \in \mathbb{R}$ $A_i \succeq 0$ B_i **(Hermitian)**

We can recover primal constraints by asking:

When is Lagrangian **bounded** if we **minimize** over **dual** variables?

$$\mathcal{L} = f(x)$$

$$+ \sum_i \lambda_i (w_i - g_i(x)) \xrightarrow[\lambda_i \geq 0]{\min} \text{Not } -\infty \text{ iff } w_i - g_i(x) \geq 0$$

$$+ \sum_i \mu_i (v_i - h_i(x)) \xrightarrow[\mu_i \in \mathbb{R}]{\min} \text{Not } -\infty \text{ iff } v_i - h_i(x) = 0$$

$$+ \sum_i \mathrm{Tr}[A_i (W_i - G_i(x))] \xrightarrow[A_i \succeq 0]{\min} \text{Not } -\infty \text{ iff } W_i - G_i(x) \succeq 0$$

$$+ \sum_i \mathrm{Tr}[B_i (V_i - H_i(x))] \xrightarrow[B_i]{\min} \text{Not } -\infty \text{ iff } V_i - H_i(x) = 0$$

Lagrangian contains all information about primal SDP

SDP duality IV - A hack to find the dual...

Step 2: Rearrange the Lagrangian

$$\mathcal{L} = \sum_i \lambda_i w_i + \sum_j \mu_j v_j + \sum_k \mathrm{Tr}[A_k W_k] + \sum_l \mathrm{Tr}[B_l V_l]$$

(Terms without primal variables)

$$- \left(\sum_i \lambda_i g_i(x) + \sum_j \mu_j h_j(x) + \sum_k \mathrm{Tr}[A_k G_k(x)] + \sum_l \mathrm{Tr}[B_l H_l(x)] \right)$$

(Terms with primal variables)

SDP duality IV - A hack to find the dual…

Step 3: Form the dual

$$\mathcal{L} = \sum_i \lambda_i w_i + \sum_j \mu_j v_j + \sum_k \mathrm{Tr}[A_k W_k] + \sum_l \mathrm{Tr}[B_l V_l]$$

$$- \left(\sum_i \lambda_i g_i(x) + \sum_j \mu_j h_j(x) + \sum_k \mathrm{Tr}[A_k G_k(x)] + \sum_l \mathrm{Tr}[B_l H_l(x)] \right)$$

SDP duality IV - A hack to find the dual...

Step 3: Form the dual

$$\mathcal{L} = \sum_i \lambda_i w_i + \sum_j \mu_j v_j + \sum_k \mathrm{Tr}[A_k W_k] + \sum_l \mathrm{Tr}[B_l V_l]$$
$$- \left(\sum_i \lambda_i g_i(x) + \sum_j \mu_j h_j(x) + \sum_k \mathrm{Tr}[A_k G_k(x)] + \sum_l \mathrm{Tr}[B_l H_l(x)] \right)$$

When is Lagrangian **bounded** if we **maximize** over **primal** variables?

SDP duality IV - A hack to find the dual...

Step 3: Form the dual

$$\mathcal{L} = \sum_i \lambda_i w_i + \sum_j \mu_j v_j + \sum_k \mathrm{Tr}[A_k W_k] + \sum_l \mathrm{Tr}[B_l V_l]$$
$$- \left(\sum_i \lambda_i g_i(x) + \sum_j \mu_j h_j(x) + \sum_k \mathrm{Tr}[A_k G_k(x)] + \sum_l \mathrm{Tr}[B_l H_l(x)] \right)$$

When is Lagrangian **bounded** if we **maximize** over **primal** variables?

$$\begin{aligned} \min \quad & \sum_i \lambda_i w_i + \sum_j \mu_j v_j + \sum_k \mathrm{Tr}[A_k W_k] + \sum_l \mathrm{Tr}[B_l V_l] \\ \text{s.t.} \quad & \text{Constraints on dual variables implied by boundedness} \\ & \lambda_i \geq 0, \;\; \mu_j \in \mathbb{R}, \;\; A_k \succeq 0, \;\; B_l \text{ Hermitian.} \end{aligned}$$

Constraints introduced when forming Lagrangian

SDP duality IV - A hack to find the dual…

Example:

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \le 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

SDP duality IV - A hack to find the dual...

Example:

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

$\longrightarrow$

$$\mathcal{L} = x + \lambda(2 - x - y) + \mathrm{Tr}\left[\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} 1 & x \\ x & y \end{pmatrix}\right] + \mathrm{Tr}\left[\begin{pmatrix} d & e \\ e & f \end{pmatrix}\begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix}\right]$$

SDP duality IV - A hack to find the dual…

Example:

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

$$\mathcal{L} = x + \lambda(2 - x - y) + \mathrm{Tr}\left[\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} 1 & x \\ x & y \end{pmatrix}\right] + \mathrm{Tr}\left[\begin{pmatrix} d & e \\ e & f \end{pmatrix}\begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix}\right]$$

$$\begin{aligned} \mathcal{L} = {} & 2\lambda + a + 2d \\ & + x(1 - \lambda + 2b + 2e + 2f) \\ & + y(-\lambda + c + 2e) \end{aligned}$$

SDP duality IV - A hack to find the dual...

Example:

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

$$\mathcal{L} = x + \lambda(2 - x - y) + \mathrm{Tr}\left[\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} 1 & x \\ x & y \end{pmatrix}\right] + \mathrm{Tr}\left[\begin{pmatrix} d & e \\ e & f \end{pmatrix}\begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix}\right]$$

$$\begin{aligned} \mathcal{L} = {} & 2\lambda + a + 2d \\ & + x(1 - \lambda + 2b + 2e + 2f) \\ & + y(-\lambda + c + 2e) \end{aligned}$$

$$\begin{aligned} \min \quad & 2\lambda + a + 2d \\ \text{s.t.} \quad & 1 - \lambda + 2b + 2e + 2f = 0 \\ & -\lambda + c + 2e = 0 \\ & \begin{pmatrix} a & b \\ b & c \end{pmatrix} \succeq 0, \ \begin{pmatrix} d & e \\ e & f \end{pmatrix} \succeq 0 \\ & \lambda \geq 0\,. \end{aligned}$$

SDP duality IV - A hack to find the dual...

Example:

$$\begin{aligned} \max \quad & x \\ \text{s.t.} \quad & x + y \leq 2 \\ & \begin{pmatrix} 1 & x \\ x & y \end{pmatrix} \succeq 0 \\ & \begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix} \succeq 0 \end{aligned}$$

$$\mathcal{L} = x + \lambda(2 - x - y) + \mathrm{Tr}\left[\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} 1 & x \\ x & y \end{pmatrix}\right] + \mathrm{Tr}\left[\begin{pmatrix} d & e \\ e & f \end{pmatrix}\begin{pmatrix} 2 & x+y \\ x+y & 2x \end{pmatrix}\right]$$

$$\mathcal{L} = 2\lambda + a + 2d + x(1 - \lambda + 2b + 2e + 2f) + y(-\lambda + c + 2e)$$

$$\begin{aligned} \min \quad & 2\lambda + a + 2d \\ \text{s.t.} \quad & 1 - \lambda + 2b + 2e + 2f = 0 \\ & -\lambda + c + 2e = 0 \\ & \begin{pmatrix} a & b \\ b & c \end{pmatrix} \succeq 0, \ \begin{pmatrix} d & e \\ e & f \end{pmatrix} \succeq 0 \\ & \lambda \geq 0\,. \end{aligned}$$

$$\begin{aligned} \min \quad & a + 2c + 2d + 4e \\ \text{s.t.} \quad & 1 + 2b - c + 2f = 0 \\ & \begin{pmatrix} a & b \\ b & c \end{pmatrix} \succeq 0, \ \begin{pmatrix} d & e \\ e & f \end{pmatrix} \succeq 0 \end{aligned}$$