SDP as a tool for quantum information

Lecture 2: Dual certificates, strong duality, and numerical SDPs

Marco Túlio Quintino

QALYPSO 2026

Lecture 2: Dual certificates and computation

6. Duality theory 2.0

Lecture 2: Dual certificates and computation

6. Duality theory 2.0
7. Analytical applications

Lecture 2: Dual certificates and computation

6. Duality theory 2.0
7. Analytical applications
8. Numerical solvers

Lecture 2: Dual certificates and computation

6. Duality theory 2.0
7. Analytical applications
8. Numerical solvers
9. Computer-assisted proofs

Lecture 2: Dual certificates and computation

6. Duality theory 2.0
7. Analytical applications
8. Numerical solvers
9. Computer-assisted proofs
10. Outlook

Recap

Primal points achieve; dual points certify

PRIMAL – achievable

$$\begin{aligned} \alpha := \sup_{\rho} \quad & \mathrm{tr}(A\rho) \\ \text{subject to} \quad & \Lambda(\rho) = B \\ & \rho \succeq 0 \end{aligned}$$

DUAL – upper bound

Primal points achieve; dual points certify

PRIMAL – achievable

$$\begin{aligned} \alpha := \sup_{\rho} \quad & \operatorname{tr}(A\rho) \\ \text{subject to} \quad & \Lambda(\rho) = B \\ & \rho \succeq 0 \end{aligned}$$

Feasible points are attainable strategies

DUAL – upper bound

$$\begin{aligned} \beta := \inf_{Y} \quad & \operatorname{tr}(BY) \\ \text{subject to} \quad & \Lambda^\dagger(Y) \succeq A \\ & Y = Y^\dagger \end{aligned}$$

Feasible points are rigorous upper bounds

Primal points achieve; dual points certify

PRIMAL – achievable

$$\begin{aligned} \alpha := \sup_{\rho} \quad & \mathrm{tr}(A\rho) \\ \text{subject to} \quad & \Lambda(\rho) = B \\ & \rho \succeq 0 \end{aligned}$$

Feasible points are attainable strategies

DUAL – upper bound

$$\begin{aligned} \beta := \inf_{Y} \quad & \mathrm{tr}(BY) \\ \text{subject to} \quad & \Lambda^\dagger(Y) \succeq A \\ & Y = Y^\dagger \end{aligned}$$

Feasible points are rigorous upper bounds

$\alpha \leq \beta$ (weak duality)

State discrimination is our running SDP

An ensemble $\{p_i, \rho_i\}_{i=1}^N$ is given; the measurement is the variable

State discrimination is our running SDP

An ensemble $\{p_i, \rho_i\}_{i=1}^N$ is given; the measurement is the variable

PRIMAL – achievable

$$
\begin{aligned}
p^\star_{\mathrm{succ}} = \sup_{\{M_i\}} \quad & \sum_{i=1}^N p_i \operatorname{tr}(\rho_i M_i) \\
\text{subject to} \quad & \sum_{i=1}^N M_i = \mathbb{1} \\
& M_i \succeq 0 \qquad \text{for every } i
\end{aligned}
$$

State discrimination is our running SDP

An ensemble $\{p_i, \rho_i\}_{i=1}^N$ is given; the measurement is the variable

PRIMAL – achievable

$$
\begin{aligned}
p^\star_{\mathrm{succ}} = \sup_{\{M_i\}} \quad & \sum_{i=1}^N p_i \, \mathrm{tr}(\rho_i M_i) \\
\text{subject to} \quad & \sum_{i=1}^N M_i = \mathbb{1} \\
& M_i \succeq 0 \qquad \text{for every } i
\end{aligned}
$$

Linear objective + affine normalisation + PSD constraints $\implies$ SDP

6. Duality theory 2.0

From upper bounds to strong duality

The dual problem for state discrimination

PRIMAL – achievable

$$
\begin{aligned}
p^\star_{\text{succ}} := \max_{\{M_i\}} \quad & \sum_{i=1}^{N} p_i \operatorname{tr}(M_i \rho_i) \\
\text{subject to} \quad & M_i \succeq 0 \qquad \forall i \\
& \sum_{i=1}^{N} M_i = \mathbb{1}
\end{aligned}
$$

The dual problem for state discrimination

PRIMAL – achievable

$$p^{\star}_{\text{succ}} := \max_{\{M_i\}} \quad \sum_{i=1}^{N} p_i \operatorname{tr}(M_i \rho_i)$$

$$\text{subject to} \qquad M_i \succeq 0 \qquad \forall i \qquad \longleftrightarrow \qquad \sigma_i \succeq 0$$

$$\sum_{i=1}^{N} M_i = \mathbb{1}$$

The dual problem for state discrimination

PRIMAL – achievable

$$
\begin{aligned}
p^\star_{\text{succ}} := \max_{\{M_i\}} \quad & \sum_{i=1}^{N} p_i \operatorname{tr}(M_i \rho_i) & & \\
\text{subject to} \quad & M_i \succeq 0 \qquad \forall i & \longleftrightarrow \quad & \sigma_i \succeq 0 \\
& \sum_{i=1}^{N} M_i = \mathbb{1} & \longleftrightarrow \quad & Y = Y^\dagger
\end{aligned}
$$

Write the Lagrangian and factorise M_i

PRIMAL – achievable

$$p^\star_{\text{succ}} := \max_{\{M_i\}} \quad \sum_i p_i \operatorname{tr}(M_i \rho_i)$$

$$\text{subject to} \quad M_i \succeq 0, \quad \sum_i M_i = \mathbb{1}$$

$$L(\{M_i\}, Y, \{\sigma_i\}_i) := \sum_i p_i \operatorname{tr}(M_i \rho_i) + \operatorname{tr}\left[Y\left(\mathbb{1} - \sum_i M_i\right)\right] + \sum_i \operatorname{tr}(\sigma_i M_i)$$

Write the Lagrangian and factorise M_i

PRIMAL – achievable

$$p^\star_{\text{succ}} := \max_{\{M_i\}} \quad \sum_i p_i \operatorname{tr}(M_i \rho_i)$$

$$\text{subject to} \quad M_i \succeq 0, \quad \sum_i M_i = \mathbb{1}$$

$$L(\{M_i\}, Y, \{\sigma_i\}_i) := \sum_i p_i \operatorname{tr}(M_i \rho_i) + \operatorname{tr}\left[Y\left(\mathbb{1} - \sum_i M_i\right)\right] + \sum_i \operatorname{tr}(\sigma_i M_i)$$

$$= \operatorname{tr}(Y) + \sum_i \operatorname{tr}[M_i\,(p_i\rho_i - Y + \sigma_i)]$$

Write the Lagrangian and factorise M_i

PRIMAL – achievable

$$p^\star_{\text{succ}} := \max_{\{M_i\}} \sum_i p_i \operatorname{tr}(M_i \rho_i)$$

subject to $\quad M_i \succeq 0, \quad \sum_i M_i = \mathbb{1}$

DUAL – upper bound

minimise $\quad \operatorname{tr}(Y)$

subject to $\quad p_i\rho_i - Y + \sigma_i = 0 \quad \forall i$

$\quad \sigma_i \succeq 0 \quad \forall i$

$$L(\{M_i\}, Y, \{\sigma_i\}_i) := \sum_i p_i \operatorname{tr}(M_i\rho_i) + \operatorname{tr}\left[Y\left(\mathbb{1} - \sum_i M_i\right)\right] + \sum_i \operatorname{tr}(\sigma_i M_i)$$

$$= \operatorname{tr}(Y) + \sum_i \operatorname{tr}[M_i\,(p_i\rho_i - Y + \sigma_i)]$$

Eliminate slack variables

$$p_i \rho_i - Y + \sigma_i = 0 \quad \iff \quad \sigma_i = Y - p_i \rho_i \qquad \forall i$$

Eliminate slack variables

$$p_i\rho_i - Y + \sigma_i = 0 \quad \Longleftrightarrow \quad \sigma_i = Y - p_i\rho_i \qquad \forall i$$

Since $\sigma_i \succeq 0$,

$$\sigma_i = Y - p_i\rho_i \succeq 0 \quad \Longleftrightarrow \quad p_i\rho_i \preceq Y \qquad \forall i$$

σ_i is a redundant slack variable

Eliminate slack variables

$$p_i\rho_i - Y + \sigma_i = 0 \quad \Longleftrightarrow \quad \sigma_i = Y - p_i\rho_i \qquad \forall i$$

Since $\sigma_i \succeq 0$,

$$\sigma_i = Y - p_i\rho_i \succeq 0 \quad \Longleftrightarrow \quad p_i\rho_i \preceq Y \qquad \forall i$$

σ_i is a redundant slack variable

PRIMAL – achievable

$$p^\star_{\text{succ}} := \max_{\{M_i\}} \quad \sum_i p_i \operatorname{tr}(M_i\rho_i)$$

subject to $\quad M_i \succeq 0, \quad \sum_i M_i = \mathbb{1}$

DUAL – upper bound

$$\beta := \min_{Y} \quad \operatorname{tr}(Y)$$

subject to $\quad p_i\rho_i \preceq Y \quad \forall i$

State discrimination problem from Lecture 1

$$p_1 = p_2 = \frac{1}{2}, \qquad \rho_1 = |0\rangle\langle 0|\,, \qquad \rho_2 = |+\rangle\langle +|$$

State discrimination problem from Lecture 1

$$p_1 = p_2 = \frac{1}{2}, \qquad \rho_1 = |0\rangle\langle 0|, \qquad \rho_2 = |+\rangle\langle +|$$

$$|m_1\rangle = \cos\left(\frac{\pi}{8}\right)|0\rangle - \sin\left(\frac{\pi}{8}\right)|1\rangle, \qquad |m_2\rangle = \sin\left(\frac{\pi}{8}\right)|0\rangle + \cos\left(\frac{\pi}{8}\right)|1\rangle, \qquad M_i = |m_i\rangle\langle m_i|$$

State discrimination problem from Lecture 1

$$p_1 = p_2 = \frac{1}{2}, \qquad \rho_1 = |0\rangle\langle 0|\,, \qquad \rho_2 = |+\rangle\langle +|$$

$$|m_1\rangle = \cos\left(\frac{\pi}{8}\right)|0\rangle - \sin\left(\frac{\pi}{8}\right)|1\rangle\,, \qquad |m_2\rangle = \sin\left(\frac{\pi}{8}\right)|0\rangle + \cos\left(\frac{\pi}{8}\right)|1\rangle\,, \qquad M_i = |m_i\rangle\langle m_i|$$

$$\begin{aligned} p_{\text{succ}}(\{M_1, M_2\}) &= \frac{1}{2}\operatorname{tr}(\rho_1 M_1) + \frac{1}{2}\operatorname{tr}(\rho_2 M_2) \\ &= \cos^2\left(\frac{\pi}{8}\right) \\ &= \frac{2+\sqrt{2}}{4} \end{aligned}$$

This proves achievability; optimality still needs an upper bound

A dual operator reaches the same value

DUAL – upper bound

$$\begin{aligned} \min_{Y} \quad & \operatorname{tr}(Y) \\ \text{subject to} \quad & \frac{1}{2}|0\rangle\langle 0| \preceq Y \\ & \frac{1}{2}|+\rangle\langle +| \preceq Y \end{aligned}$$

A dual operator reaches the same value

DUAL – upper bound

$$\min_{Y} \quad \mathrm{tr}(Y)$$

subject to

$$\frac{1}{2}\,|0\rangle\langle 0| \preceq Y$$

$$\frac{1}{2}\,|+\rangle\langle +| \preceq Y$$

$$Y^\star := \frac{1}{4}\left(\rho_1 + \rho_2 + \frac{\mathbb{1}}{\sqrt{2}}\right)$$

A dual operator reaches the same value

DUAL – upper bound

$$\min_{Y} \quad \mathrm{tr}(Y)$$

subject to

$$\frac{1}{2}|0\rangle\langle 0| \preceq Y$$

$$\frac{1}{2}|+\rangle\langle +| \preceq Y$$

$$Y^\star := \frac{1}{4}\left(\rho_1 + \rho_2 + \frac{\mathbb{1}}{\sqrt{2}}\right)$$

$$\mathrm{tr}(Y^\star) = \frac{1}{4}\left(1 + 1 + \frac{2}{\sqrt{2}}\right) = \frac{2+\sqrt{2}}{4}$$

A dual operator reaches the same value

DUAL – upper bound

$$\min_{Y} \quad \mathrm{tr}(Y)$$

$$\text{subject to} \quad \frac{1}{2}|0\rangle\langle 0| \preceq Y, \qquad \frac{1}{2}|+\rangle\langle +| \preceq Y$$

$$Y^\star := \frac{1}{4}\left(\rho_1 + \rho_2 + \frac{\mathbb{1}}{\sqrt{2}}\right)$$

$$\mathrm{tr}(Y^\star) = \frac{1}{4}\left(1 + 1 + \frac{2}{\sqrt{2}}\right) = \frac{2+\sqrt{2}}{4}$$

$$Y^\star - \frac{1}{2}\rho_1 = \frac{1}{2\sqrt{2}} M_2 \succeq 0,$$

A dual operator reaches the same value

DUAL – upper bound

$$\min_{Y} \quad \mathrm{tr}(Y)$$

$$\text{subject to} \quad \frac{1}{2}|0\rangle\langle 0| \preceq Y, \quad \frac{1}{2}|+\rangle\langle +| \preceq Y$$

$$Y^\star := \frac{1}{4}\left(\rho_1 + \rho_2 + \frac{\mathbb{1}}{\sqrt{2}}\right)$$

$$\mathrm{tr}(Y^\star) = \frac{1}{4}\left(1 + 1 + \frac{2}{\sqrt{2}}\right) = \frac{2+\sqrt{2}}{4}$$

$$Y^\star - \frac{1}{2}\rho_1 = \frac{1}{2\sqrt{2}} M_2 \succeq 0, \qquad Y^\star - \frac{1}{2}\rho_2 = \frac{1}{2\sqrt{2}} M_1 \succeq 0$$

A dual operator reaches the same value

DUAL – upper bound

$$\min_{Y} \quad \mathrm{tr}(Y)$$

subject to
$$\frac{1}{2}|0\rangle\langle 0| \preceq Y$$
$$\frac{1}{2}|+\rangle\langle +| \preceq Y$$

$$Y^\star := \frac{1}{4}\left(\rho_1 + \rho_2 + \frac{\mathbb{1}}{\sqrt{2}}\right)$$

$$\mathrm{tr}(Y^\star) = \frac{1}{4}\left(1 + 1 + \frac{2}{\sqrt{2}}\right) = \frac{2+\sqrt{2}}{4}$$

$$Y^\star - \frac{1}{2}\rho_1 = \frac{1}{2\sqrt{2}}M_2 \succeq 0, \qquad Y^\star - \frac{1}{2}\rho_2 = \frac{1}{2\sqrt{2}}M_1 \succeq 0$$

$Y^\star$ is dual feasible, hence $\mathrm{tr}(Y^\star)$ is an upper bound.

Matching bounds prove optimality

$$\frac{2+\sqrt{2}}{4} = p_{\mathrm{succ}}(\{M_1, M_2\}) \leq p^{\star}_{\mathrm{succ}} \leq \mathrm{tr}(Y^{\star}) = \frac{2+\sqrt{2}}{4}$$

Matching bounds prove optimality

$$\frac{2+\sqrt{2}}{4} = p_{\text{succ}}(\{M_1, M_2\}) \leq p^{\star}_{\text{succ}} \leq \operatorname{tr}(Y^{\star}) = \frac{2+\sqrt{2}}{4}$$

$$\boxed{p^{\star}_{\text{succ}} = \frac{2+\sqrt{2}}{4}}$$

Strong duality

For an SDP, the primal and dual sometimes have the same optimal value. That is, $\alpha = \beta$.

Strong duality

For an SDP, the primal and dual sometimes have the same optimal value. That is, $\alpha = \beta$.

When this happens, we say that **strong duality** holds.

Strong duality

For an SDP, the primal and dual sometimes have the same optimal value. That is, $\alpha = \beta$.

When this happens, we say that **strong duality** holds.

Strong duality holds for many (perhaps most) of the problems we care about.

Strong duality

For an SDP, the primal and dual sometimes have the same optimal value. That is, $\alpha = \beta$.

When this happens, we say that **strong duality** holds.

Strong duality holds for many (perhaps most) of the problems we care about.

But when does strong duality hold?

Slater's theorem

Slater conditions

If the primal is bounded above and strictly feasible, that is,

$$\alpha < +\infty \qquad \text{and} \qquad \exists\, \rho \succ 0 \text{ such that } \Lambda(\rho) = B,$$

Slater's theorem

Slater conditions

If the primal is bounded above and strictly feasible, that is,

$$\alpha < +\infty \qquad \text{and} \qquad \exists\, \rho \succ 0 \text{ such that } \Lambda(\rho) = B,$$

then $\boxed{\alpha = \beta}$. That is, we have strong duality.

Slater's theorem

Slater conditions

If the primal is bounded above and strictly feasible, that is,

$$\alpha < +\infty \qquad \text{and} \qquad \exists\, \rho \succ 0 \text{ such that } \Lambda(\rho) = B,$$

then $\boxed{\alpha = \beta}$. That is, we have strong duality.

Two-sided Slater conditions

If the primal and dual are strictly feasible, then $\boxed{\alpha = \beta}$.

State discrimination satisfies Slater conditions

PRIMAL – achievable

$$M_i = \frac{\mathbb{1}}{N} \succ 0, \qquad \sum_{i=1}^{N} M_i = \mathbb{1}$$

DUAL – upper bound

State discrimination satisfies Slater conditions

PRIMAL – achievable

$$M_i = \frac{\mathbb{1}}{N} \succ 0, \qquad \sum_{i=1}^{N} M_i = \mathbb{1}$$

DUAL – upper bound

$$Y = \lambda \mathbb{1}, \qquad \lambda > \max_i p_i$$

$$Y - p_i \rho_i \succ 0 \qquad \forall i$$

State discrimination satisfies Slater conditions

PRIMAL – achievable

$$M_i = \frac{\mathbb{1}}{N} \succ 0, \qquad \sum_{i=1}^{N} M_i = \mathbb{1}$$

DUAL – upper bound

$$Y = \lambda \mathbb{1}, \qquad \lambda > \max_i p_i$$

$$Y - p_i \rho_i \succ 0 \qquad \forall i$$

Strong duality holds, and both optima are attained

$$p^\star_{\text{succ}} := \max_{\substack{M_i \succeq 0 \ \forall i \\ \sum_i M_i = \mathbb{1}}} \sum_i p_i \operatorname{tr}(\rho_i M_i) = \min_{Y \succeq p_i \rho_i \ \forall i} \operatorname{tr}(Y)$$

7. Analytical applications

A useful certificate from the identity

A universal dimension bound

Theorem

For any ensemble $\{p_i, \rho_i\}_{i=1}^{N}$ on a d-dimensional Hilbert space, $p^{\star}_{\mathrm{succ}} \leq d \max_i p_i$.

A universal dimension bound

Theorem

For any ensemble $\{p_i, \rho_i\}_{i=1}^N$ on a d-dimensional Hilbert space, $p^\star_{\mathrm{succ}} \leq d \max_i p_i$.

Proof. Every density operator satisfies $0 \preceq \rho_i \preceq \mathbb{1}$.

A universal dimension bound

Theorem

For any ensemble $\{p_i, \rho_i\}_{i=1}^N$ on a d-dimensional Hilbert space, $p^\star_{\text{succ}} \leq d \max_i p_i$.

Proof. Every density operator satisfies $0 \preceq \rho_i \preceq \mathbb{1}$.
Hence,

$$p_i \rho_i \preceq p_i \mathbb{1} \preceq \left(\max_i p_i\right) \mathbb{1} = Y$$

A universal dimension bound

Theorem

For any ensemble $\{p_i, \rho_i\}_{i=1}^N$ on a d-dimensional Hilbert space, $p_{\mathrm{succ}}^\star \leq d \max_i p_i$.

Proof. Every density operator satisfies $0 \preceq \rho_i \preceq \mathbb{1}$.
Hence,

$$p_i \rho_i \preceq p_i \mathbb{1} \preceq \left(\max_i p_i\right) \mathbb{1} = Y$$

Thus $Y = \left(\max_i p_i\right) \mathbb{1}$ is dual feasible and $\mathrm{tr}(Y) = d \max_i p_i$ is an upper bound. □

Uniform priors reduce the bound to d/N

Suppose

$$p_i = \frac{1}{N} \qquad i = 1, \ldots, N$$

Uniform priors reduce the bound to d/N

Suppose

$$p_i = \frac{1}{N} \qquad i = 1, \ldots, N$$

The universal certificate becomes

$$Y = \frac{\mathbb{1}}{N} \succeq \frac{\rho_i}{N} \qquad \text{for every } i$$

Uniform priors reduce the bound to d/N

Suppose

$$p_i = \frac{1}{N} \qquad i = 1, \ldots, N$$

The universal certificate becomes

$$Y = \frac{\mathbb{1}}{N} \succeq \frac{\rho_i}{N} \qquad \text{for every } i$$

For uniform priors, $p^\star_{\text{succ}} \leq \operatorname{tr}(Y) = \dfrac{d}{N}$.

The most discriminable states

Theorem

Let $\{|\psi_i\rangle\}_{i=1}^N$ be pure states in $\mathbb{C}^d$. If

$$\frac{1}{N}\sum_{i=1}^{N} |\psi_i\rangle\langle\psi_i| = \frac{\mathbb{1}}{d},$$

then, for uniform priors, $p^\star_{\mathrm{succ}} = \dfrac{d}{N}$.

The most discriminable states

Proof. Set $M_i := \dfrac{d}{N} |\psi_i\rangle\langle\psi_i|$. Then $M_i \succeq 0$ and $\sum_i M_i = \frac{d}{N} \sum_i |\psi_i\rangle\langle\psi_i| = \mathbb{1}$, so $\{M_i\}$ is a POVM.

$$
\begin{aligned}
p_{\text{succ}}(\{M_i\}) &= \frac{d}{N^2} \sum_{i=1}^{N} \operatorname{tr}(|\psi_i\rangle\langle\psi_i|\,|\psi_i\rangle\langle\psi_i|) \\
&= \frac{d}{N}.
\end{aligned}
$$

The most discriminable states

Proof. Set $M_i := \dfrac{d}{N} |\psi_i\rangle\langle\psi_i|$. Then $M_i \succeq 0$ and $\sum_i M_i = \frac{d}{N} \sum_i |\psi_i\rangle\langle\psi_i| = \mathbb{1}$, so $\{M_i\}$ is a POVM.

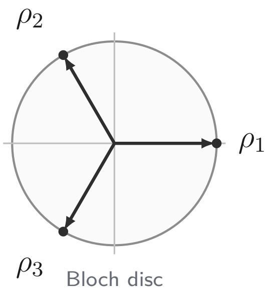
$$p_{\text{succ}}(\{M_i\}) = \frac{d}{N^2} \sum_{i=1}^{N} \operatorname{tr}(|\psi_i\rangle\langle\psi_i|\,|\psi_i\rangle\langle\psi_i|) = \frac{d}{N}.$$

The dual feasible point $Y = \frac{\mathbb{1}}{N}$ gives $\operatorname{tr}(Y) = \frac{d}{N}$. $\square$

The trine states

Let $\omega = e^{2\pi \mathrm{i}/3}$ and, for $i = 1, 2, 3$,

$$|\psi_i\rangle = \frac{|0\rangle + \omega^{i-1}|1\rangle}{\sqrt{2}}, \qquad \rho_i = |\psi_i\rangle\langle\psi_i|$$

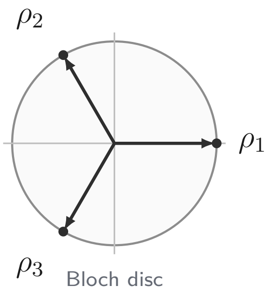


The trine states

Let $\omega = e^{2\pi \mathrm{i}/3}$ and, for $i = 1, 2, 3$,

$$|\psi_i\rangle = \frac{|0\rangle + \omega^{i-1}|1\rangle}{\sqrt{2}}, \qquad \rho_i = |\psi_i\rangle\langle\psi_i|$$

$$\frac{1}{3}\sum_{i=1}^{3}\rho_i = \frac{\mathbb{1}}{2}$$



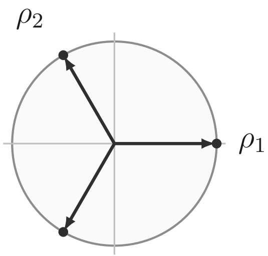
The trine states

Let $\omega = e^{2\pi \mathrm{i}/3}$ and, for $i = 1, 2, 3$,

$$|\psi_i\rangle = \frac{|0\rangle + \omega^{i-1}|1\rangle}{\sqrt{2}}, \qquad \rho_i = |\psi_i\rangle\langle\psi_i|$$

$$\frac{1}{3}\sum_{i=1}^{3} \rho_i = \frac{\mathbb{1}}{2}$$

$$M_i = \frac{2}{3}\rho_i, \qquad Y = \frac{\mathbb{1}}{3}, \qquad p^{\star}_{\text{succ}} = \frac{2}{3}$$

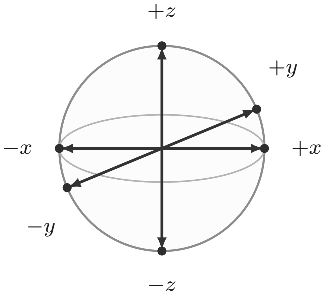


Bloch disc

Pauli eigenstates form an optimal octahedron

For $a \in \{x, y, z\}$ and $s \in \{+1, -1\}$, let

$$\rho_{a,s} = \frac{1}{2}(\mathbb{1} + s\sigma_a)$$



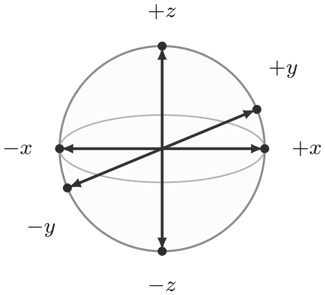
Bloch sphere

Pauli eigenstates form an optimal octahedron

For $a \in \{x, y, z\}$ and $s \in \{+1, -1\}$, let

$$\rho_{a,s} = \frac{1}{2}(\mathbb{1} + s\sigma_a)$$

$$\frac{1}{6} \sum_{a\in\{x,y,z\}} \sum_{s\in\{+1,-1\}} \rho_{a,s} = \frac{\mathbb{1}}{2}$$



Bloch sphere

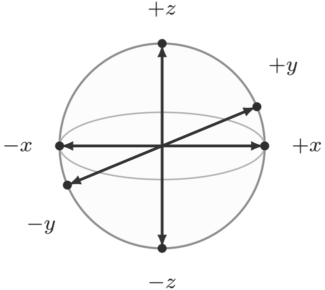
Pauli eigenstates form an optimal octahedron

For $a \in \{x, y, z\}$ and $s \in \{+1, -1\}$, let

$$\rho_{a,s} = \frac{1}{2}(\mathbb{1} + s\sigma_a)$$

$$\frac{1}{6} \sum_{a\in\{x,y,z\}} \sum_{s\in\{+1,-1\}} \rho_{a,s} = \frac{\mathbb{1}}{2}$$

$$M_{a,s} = \frac{1}{3}\rho_{a,s}, \qquad Y = \frac{\mathbb{1}}{6}, \qquad p^\star_{\text{succ}} = \frac{1}{3}$$



Bloch sphere

8. Numerical solvers

From an exact model to approximate candidates

Numerical solvers

- There are *efficient* computational algorithms to solve SDPs.

Numerical solvers

- There are *efficient* computational algorithms to solve SDPs.
- For a well-posed SDP of manageable size, a solver can usually return approximate primal and dual candidates—or diagnose a problem.

Efficient means controlled scaling

An SDP algorithm receives

$$\text{matrix data} \quad + \quad \text{an accuracy request } \varepsilon$$

Efficient means controlled scaling

An SDP algorithm receives

$$\text{matrix data} \quad + \quad \text{an accuracy request } \varepsilon$$

and returns approximate primal and dual candidates, objective values, and diagnostics.

Efficient means controlled scaling

An SDP algorithm receives

matrix data $\quad + \quad$ an accuracy request ε

and returns approximate primal and dual candidates, objective values, and diagnostics.

Under standard regularity and finite-precision assumptions, theoretical algorithms have polynomial scaling in the encoded input size and $\log(1/\varepsilon)$

The time complexity for solving an SDP

Given: $A = A^\dagger \in \mathcal{L}(\mathbb{C}^d)$, $\quad B = B^\dagger \in \mathcal{L}(\mathbb{C}^{d'})$

$\Lambda : \mathcal{L}(\mathbb{C}^d) \longrightarrow \mathcal{L}(\mathbb{C}^{d'})$ Hermiticity-preserving

$$\begin{aligned} \text{maximise} \quad & \operatorname{tr}(A\rho) \\ \text{subject to} \quad & \Lambda(\rho) = B, \qquad \rho \succeq 0 \end{aligned}$$

The time complexity for solving an SDP

Given: $A = A^\dagger \in \mathcal{L}(\mathbb{C}^d)$, $\quad B = B^\dagger \in \mathcal{L}(\mathbb{C}^{d'})$

$\Lambda : \mathcal{L}(\mathbb{C}^d) \longrightarrow \mathcal{L}(\mathbb{C}^{d'})$ Hermiticity-preserving

$$\begin{aligned} \text{maximise} \quad & \operatorname{tr}(A\rho) \\ \text{subject to} \quad & \Lambda(\rho) = B, \qquad \rho \succeq 0 \end{aligned}$$

Roughly speaking,

the time needed to solve this problem to precision ε grows polynomially in d, d', and $\log(1/\varepsilon)$.

Examples?

The primal SDP is one short function

```julia
using LinearAlgebra
using JuMP
import Clarabel

function state_discrimination(rho, p)
    N, d = length(rho), size(rho[1], 1)
    model = Model(Clarabel.Optimizer)
    M = [@variable(model, [1:d, 1:d] in HermitianPSDCone()) for i in 1:N]
    @constraint(model, sum(M) == I)
    @objective(
        model,
        Max,
        sum(p[i] * real(tr(rho[i] * M[i])) for i in 1:N),
    )
    optimize!(model)
    return objective_value(model), value.(M)
end
```

The dual is just as direct

```
function state_discrimination_dual(rho, p)
    N, d = length(rho), size(rho[1], 1)
    model = Model(Clarabel.Optimizer)
    @variable(model, Y[1:d, 1:d] in HermitianPSDCone())
    @constraint(
        model,
        [i = 1:N],
        Hermitian(Y - p[i] * rho[i]) in HermitianPSDCone(),
    )
    @objective(model, Min, real(tr(Y)))
    optimize!(model)
    return objective_value(model), value.(Y)
end
```

Discriminating $|0\rangle$ from $|+\rangle$ in a few lines

```
include("state_discrimination.jl")

ket0 = [1; 0]
ket_plus = [1; 1] / sqrt(2)

rho_1 = ket0 * ket0'
rho_2 = ket_plus * ket_plus'
p_1 = 1 / 2
p_2 = 1 / 2

rho = [rho_1, rho_2]
p = [p_1, p_2]

p_primal, M = state_discrimination(rho, p)
p_dual, Y = state_discrimination_dual(rho, p)

exact = (2 + sqrt(2)) / 4

println("rho      = ", rho)
println("p        = ", p)
println("primal   = ", p_primal)
println("dual     = ", p_dual)
println("expected = ", exact)
```

Live demonstration!

The numerical outcome

NUMERICAL

primal	0.8535533694
dual	0.8535533898
expected	0.8535533906

EXACT

$$p^{\star}_{\text{succ}} = \frac{2+\sqrt{2}}{4}$$

$$Y^{\star} = \frac{1}{4}\left(\rho_1 + \rho_2 + \frac{\mathbb{1}}{\sqrt{2}}\right)$$

$$M_1 = |m_1\rangle\langle m_1| , \qquad M_2 = |m_2\rangle\langle m_2|$$

Discriminating the six Pauli eigenstates

```julia
include("state_discrimination.jl")
sigma_x = [0 1; 1 0]
sigma_y = [0 -im; im 0]
sigma_z = [1 0; 0 -1]

rho = [Matrix((I + s * sigma) / 2)
       for sigma in (sigma_x, sigma_y, sigma_z) for s in (1, -1)]
p = fill(1 / 6, 6)

p_primal, M = state_discrimination(rho, p)
p_dual, Y = state_discrimination_dual(rho, p)

exact = 1 / 3

println("rho      = ", rho)
println("p        = ", p)
println("primal   = ", p_primal)
println("dual     = ", p_dual)
println("expected = ", exact)
```

The numerical outcome

NUMERICAL

primal	0.3333333325
dual	0.3333333330
expected	0.3333333333

The optimal POVM is not unique.

EXACT

$$\color{green}{M_{a,s}} = \frac{1}{3}\rho_{a,s}, \qquad \color{magenta}{Y} = \frac{1}{6}\mathbb{1}$$

$$p^{\star}_{\mathrm{succ}} = \frac{1}{3}$$

One exact optimal pair, independent of the solver output

Solving SDPs with computers

Language

Julia

Open source; fast.

MATLAB

Proprietary; fast.

Python

Open source; broad ecosystem; model construction can add overhead.

Solving SDPs with computers

Language	Modelling layer
Julia Open source; fast.	**JuMP** Open source.
MATLAB Proprietary; fast.	**CVX/YALMIP** Open source.
Python Open source; broad ecosystem; model construction can add overhead.	**CVXPY** Open source.

Solving SDPs with computers

Language	Modelling layer
Julia Open source; fast.	**JuMP** Open source.
MATLAB Proprietary; fast.	**CVX/YALMIP** Open source.
Python Open source; broad ecosystem; model construction can add overhead.	**CVXPY** Open source.

Solvers

Clarabel	**SCS**	**MOSEK**
Open source; modern.	Open source; fast; modest accuracy.	Proprietary; free for academics; fast and robust.

Quantum information libraries

Quantum-information libraries can simplify common tasks:

- Partial traces, partial transposition, and other convenient functions

Quantum information libraries

Quantum-information libraries can simplify common tasks:

- Partial traces, partial transposition, and other convenient functions
- Functions for tasks such as state discrimination are already implemented efficiently

Quantum information libraries

Quantum-information libraries can simplify common tasks:

- Partial traces, partial transposition, and other convenient functions
- Functions for tasks such as state discrimination are already implemented efficiently
- Entanglement tests and other standard tools

Quantum information libraries

Quantum-information libraries can simplify common tasks:

- Partial traces, partial transposition, and other convenient functions
- Functions for tasks such as state discrimination are already implemented efficiently
- Entanglement tests and other standard tools

Julia	MATLAB	Python
Ket.jl	**QETLAB**	$\vert$**toqito**$\rangle$

This is not an exhaustive list—only a few libraries I have used and liked.

Floating-point arithmetic?

Floating-point arithmetic is powerful and reliable.

Floating-point arithmetic?

Floating-point arithmetic is powerful and reliable.

But this is not a mathematical proof!

9. Computer-assisted proofs

From numerical structure to rigorous bounds

Do you trust your computer?

Sometimes yes, sometimes no.

Do you trust your computer?

Sometimes yes, sometimes no.

Floating-point numbers are convenient.
But do you trust them?

What is a floating-point number?

As in scientific notation, a normal binary floating-point number has the form

$$\underbrace{(-1)^s}_{\text{sign}} \times \underbrace{(1.b_1 b_2 \ldots b_k)_2}_{\text{significand}} \times \underbrace{2^e}_{\text{exponent}}$$

What is a floating-point number?

As in scientific notation, a normal binary floating-point number has the form

$$\underbrace{(-1)^s}_{\text{sign}} \times \underbrace{(1.b_1 b_2 \ldots b_k)_2}_{\text{significand}} \times \underbrace{2^e}_{\text{exponent}}$$

Only finitely many values can be stored exactly.

Most inputs are rounded to a nearby representable number.

Distinct numbers can share one Float64

EXACT

$$x = 9\,007\,199\,254\,740\,992, \qquad y = 9\,007\,199\,254\,740\,993$$

$$x \neq y$$

Distinct numbers can share one Float64

EXACT

$$x = 9\,007\,199\,254\,740\,992, \qquad y = 9\,007\,199\,254\,740\,993, \qquad x \neq y$$

NUMERICAL

$$\texttt{Float64}(x) = \texttt{Float64}(y) = 9\,007\,199\,254\,740\,992$$

Distinct numbers can share one Float64

EXACT

$$x = 9\,007\,199\,254\,740\,992, \qquad y = 9\,007\,199\,254\,740\,993, \qquad x \neq y$$

NUMERICAL

$$\texttt{Float64}(x) = \texttt{Float64}(y) = 9\,007\,199\,254\,740\,992$$

A finite floating-point format cannot represent every real number distinctly

Distinct numbers can share one Float64

EXACT

$$x = 9\,007\,199\,254\,740\,992, \qquad y = 9\,007\,199\,254\,740\,993, \qquad x \neq y$$

NUMERICAL

$$\texttt{Float64}(x) = \texttt{Float64}(y) = 9\,007\,199\,254\,740\,992$$

A finite floating-point format cannot represent every real number distinctly

Live demonstration time!

Floating-point addition is not associative

In exact real arithmetic, addition is associative:

$$(a + b) + c = a + (b + c)$$

Floating-point addition is not associative

In exact real arithmetic, addition is associative:

$$(a + b) + c = a + (b + c)$$

For instance, if $a = 10^{16}$, $b = -10^{16}$, and $c = 1$, then

$$(a + b) + c = a + (b + c) = 1$$

Floating-point addition is not associative

In exact real arithmetic, addition is associative:

$$(a+b)+c = a+(b+c)$$

For instance, if $a = 10^{16}$, $b = -10^{16}$, and $c = 1$, then

$$(a+b)+c = a+(b+c) = 1$$

But for `Float64`:

$$(a+b)+c = 1.0$$

but

$$a+(b+c) = 0.0$$

Floating-point addition is not associative

In exact real arithmetic, addition is associative:

$$(a + b) + c = a + (b + c)$$

For instance, if $a = 10^{16}$, $b = -10^{16}$, and $c = 1$, then

$$(a + b) + c = a + (b + c) = 1$$

But for `Float64`:

$$(a + b) + c = 1.0$$

but

$$a + (b + c) = 0.0$$

Live demonstration time!

The solver returns primal and dual candidates

$$p_1 = p_2 = \frac{1}{2}, \qquad \rho_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \rho_2 = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Primal solver output

Dual solver output

$$p^{\text{num}}_{\text{primal}} = 0.8535533694$$

$$p^{\text{num}}_{\text{dual}} = 0.8535533898$$

The solver returns primal and dual candidates

$$p_1 = p_2 = \frac{1}{2}, \qquad \rho_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \rho_2 = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Primal solver output

$$p_{\text{primal}}^{\text{num}} = 0.8535533694$$

$$M_1^{\text{num}} \approx \begin{pmatrix} 0.853551 & -0.353555 \\ -0.353555 & 0.146449 \end{pmatrix}$$

$$M_2^{\text{num}} \approx \begin{pmatrix} 0.146449 & 0.353555 \\ 0.353555 & 0.853551 \end{pmatrix}$$

Dual solver output

$$p_{\text{dual}}^{\text{num}} = 0.8535533898$$

$$Y^{\text{num}} \approx \begin{pmatrix} 0.551772 & 0.124996 \\ 0.124996 & 0.301781 \end{pmatrix}$$

The variables are part of the numerical output too.

The solver returns primal and dual candidates

$$p_1 = p_2 = \frac{1}{2}, \qquad \rho_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \qquad \rho_2 = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

Primal solver output

$$p^{\text{num}}_{\text{primal}} = 0.8535533694$$

$$M_1^{\text{num}} \approx \begin{pmatrix} 0.853551 & -0.353555 \\ -0.353555 & 0.146449 \end{pmatrix}$$

$$M_2^{\text{num}} \approx \begin{pmatrix} 0.146449 & 0.353555 \\ 0.353555 & 0.853551 \end{pmatrix}$$

Dual solver output

$$p^{\text{num}}_{\text{dual}} = 0.8535533898$$

$$Y^{\text{num}} \approx \begin{pmatrix} 0.551772 & 0.124996 \\ 0.124996 & 0.301781 \end{pmatrix}$$

The variables are part of the numerical output too.

Can these decimal candidates lead to an exact proof?

A recipe for a dual certificate

For the dual:

1. Round or reconstruct an exact Hermitian candidate:

$$Y^{\mathrm{num}} \rightsquigarrow \widetilde{Y} = \widetilde{Y}^{\dagger}.$$

A recipe for a dual certificate

For the dual:

1. Round or reconstruct an exact Hermitian candidate:

$$Y^{\mathrm{num}} \rightsquigarrow \widetilde{Y} = \widetilde{Y}^{\dagger}.$$

2. Obtain a rigorous bound on every dual slack and choose an exact $\varepsilon \geq 0$ such that

$$\lambda_{\min}\big(\widetilde{Y} - p_i \rho_i\big) \geq -\varepsilon \quad \forall i.$$

Then

$$Y^{\mathrm{cert}} := \widetilde{Y} + \varepsilon \mathbb{1} \succeq p_i \rho_i \quad \forall i.$$

A recipe for a dual certificate

For the dual:

1. Round or reconstruct an exact Hermitian candidate:

$$Y^{\mathrm{num}} \rightsquigarrow \widetilde{Y} = \widetilde{Y}^{\dagger}.$$

2. Obtain a rigorous bound on every dual slack and choose an exact $\varepsilon \geq 0$ such that

$$\lambda_{\min}\big(\widetilde{Y} - p_i \rho_i\big) \geq -\varepsilon \quad \forall i.$$

   Then

$$Y^{\mathrm{cert}} := \widetilde{Y} + \varepsilon \mathbb{1} \succeq p_i \rho_i \quad \forall i.$$

3. Evaluate $\mathrm{tr}(Y^{\mathrm{cert}})$ exactly: this is a rigorous upper bound.

An exact analytical proof from numerics

- Reconstruct an exact matrix from the decimal pattern:[1]

$$Y^{\text{exact}} := \frac{1}{8}\begin{pmatrix} 3+\sqrt{2} & 1 \\ 1 & 1+\sqrt{2} \end{pmatrix}.$$

[1] Of course, this is not always possible. But sometimes, yes! It is worth doing a few checks.

An exact analytical proof from numerics

- Reconstruct an exact matrix from the decimal pattern:[1]

$$Y^{\text{exact}} := \frac{1}{8}\begin{pmatrix} 3+\sqrt{2} & 1 \\ 1 & 1+\sqrt{2} \end{pmatrix}.$$

- Verify both inequalities by exact rank-one factorisations:

$$Y^{\text{exact}} - \frac{1}{2}\rho_1 = \frac{1}{2\sqrt{2}}\,|m_2\rangle\langle m_2| \succeq 0, \qquad Y^{\text{exact}} - \frac{1}{2}\rho_2 = \frac{1}{2\sqrt{2}}\,|m_1\rangle\langle m_1| \succeq 0.$$

[1] Of course, this is not always possible. But sometimes, yes! It is worth doing a few checks.

An exact analytical proof from numerics

- Reconstruct an exact matrix from the decimal pattern:[1]

$$Y^{\text{exact}} := \frac{1}{8}\begin{pmatrix} 3+\sqrt{2} & 1 \\ 1 & 1+\sqrt{2} \end{pmatrix}.$$

- Verify both inequalities by exact rank-one factorisations:

$$Y^{\text{exact}} - \frac{1}{2}\rho_1 = \frac{1}{2\sqrt{2}}\,|m_2\rangle\langle m_2| \succeq 0, \qquad Y^{\text{exact}} - \frac{1}{2}\rho_2 = \frac{1}{2\sqrt{2}}\,|m_1\rangle\langle m_1| \succeq 0.$$

- Evaluate the dual objective exactly:

$$p^\star_{\text{succ}} \leq p^\star_{\text{upper}} := \operatorname{tr}\left(Y^{\text{exact}}\right) = \frac{(3+\sqrt{2})+(1+\sqrt{2})}{8} = \frac{2+\sqrt{2}}{4}.$$

[1] Of course, this is not always possible. But sometimes, yes! It is worth doing a few checks.

A recipe for a primal certificate

For the primal:

1. Round or reconstruct exact Hermitian matrices:

$$M_i^{\mathrm{num}} \rightsquigarrow \widetilde{M_i} = \widetilde{M_i}^\dagger.$$

A recipe for a primal certificate

For the primal:

1. Round or reconstruct exact Hermitian matrices:
$$M_i^{\mathrm{num}} \rightsquigarrow \widetilde{M}_i = \widetilde{M}_i^{\dagger}.$$
2. Repair the normalisation constraint exactly. For example, set
$$R := \mathbb{1} - \sum_j \widetilde{M}_j, \qquad \widehat{M}_i := \widetilde{M}_i + \frac{R}{N}, \qquad \sum_i \widehat{M}_i = \mathbb{1}.$$

A recipe for a primal certificate

For the primal:

1. Round or reconstruct exact Hermitian matrices:
$$M_i^{\mathrm{num}} \rightsquigarrow \widetilde{M}_i = \widetilde{M}_i{}^{\dagger}.$$
2. Repair the normalisation constraint exactly. For example, set
$$R := \mathbb{1} - \sum_j \widetilde{M}_j, \qquad \widehat{M}_i := \widetilde{M}_i + \frac{R}{N}, \qquad \sum_i \widehat{M}_i = \mathbb{1}.$$
3. Obtain an exact $\delta \geq 0$ for which you can prove $\lambda_{\min}(\widehat{M}_i) \geq -\delta$ for every i. Choose an exact $\eta \in [0,1]$ satisfying $\eta \geq N\delta/(1+N\delta)$, and set
$$M_i^{\mathrm{cert}} := (1-\eta)\widehat{M}_i + \eta\frac{\mathbb{1}}{N} \succeq 0 \quad \forall i.$$

A recipe for a primal certificate

For the primal:

1. Round or reconstruct exact Hermitian matrices:
$$M_i^{\mathrm{num}} \rightsquigarrow \widetilde{M}_i = \widetilde{M}_i^{\dagger}.$$
2. Repair the normalisation constraint exactly. For example, set
$$R := \mathbb{1} - \sum_j \widetilde{M}_j, \qquad \widehat{M}_i := \widetilde{M}_i + \frac{R}{N}, \qquad \sum_i \widehat{M}_i = \mathbb{1}.$$
3. Obtain an exact $\delta \geq 0$ for which you can prove $\lambda_{\min}(\widehat{M}_i) \geq -\delta$ for every i. Choose an exact $\eta \in [0,1]$ satisfying $\eta \geq N\delta/(1+N\delta)$, and set
$$M_i^{\mathrm{cert}} := (1-\eta)\widehat{M}_i + \eta\frac{\mathbb{1}}{N} \succeq 0 \quad \forall i.$$
4. Evaluate the primal objective exactly to obtain a rigorous lower bound:
$$\sum_i p_i \operatorname{tr}\Big(\rho_i M_i^{\mathrm{cert}}\Big) \leq p_{\mathrm{succ}}^{\star}.$$

Certify the primal candidate

- Reconstruct exact matrices from the decimal pattern:[2]

$$M_1^{\text{exact}} := \frac{1}{4}\begin{pmatrix} 2+\sqrt{2} & -\sqrt{2} \\ -\sqrt{2} & 2-\sqrt{2} \end{pmatrix}, \qquad M_2^{\text{exact}} := \frac{1}{4}\begin{pmatrix} 2-\sqrt{2} & \sqrt{2} \\ \sqrt{2} & 2+\sqrt{2} \end{pmatrix}.$$

[2] Again, not always possible. But sometimes it works!

Certify the primal candidate

- Reconstruct exact matrices from the decimal pattern:[2]

$$M_1^{\text{exact}} := \frac{1}{4}\begin{pmatrix} 2+\sqrt{2} & -\sqrt{2} \\ -\sqrt{2} & 2-\sqrt{2} \end{pmatrix}, \qquad M_2^{\text{exact}} := \frac{1}{4}\begin{pmatrix} 2-\sqrt{2} & \sqrt{2} \\ \sqrt{2} & 2+\sqrt{2} \end{pmatrix}.$$

- Verify the constraints exactly:

$$M_i^{\text{exact}} = |m_i\rangle\langle m_i| \succeq 0 \qquad (i = 1, 2),$$
$$M_1^{\text{exact}} + M_2^{\text{exact}} = \frac{1}{4}\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix} = \mathbb{1}.$$

[2] Again, not always possible. But sometimes it works!

Certify the primal candidate

- Reconstruct exact matrices from the decimal pattern:[2]

$$M_1^{\text{exact}} := \frac{1}{4}\begin{pmatrix} 2+\sqrt{2} & -\sqrt{2} \\ -\sqrt{2} & 2-\sqrt{2} \end{pmatrix}, \qquad M_2^{\text{exact}} := \frac{1}{4}\begin{pmatrix} 2-\sqrt{2} & \sqrt{2} \\ \sqrt{2} & 2+\sqrt{2} \end{pmatrix}.$$

- Verify the constraints exactly:

$$\begin{aligned} M_i^{\text{exact}} &= |m_i\rangle\langle m_i| \succeq 0 \qquad (i = 1, 2), \\ M_1^{\text{exact}} + M_2^{\text{exact}} &= \frac{1}{4}\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix} = \mathbb{1}. \end{aligned}$$

- Evaluate the primal objective:

$$\begin{aligned} p_{\text{lower}}^{\star} &:= \frac{1}{2}\operatorname{tr}\big(\rho_1 M_1^{\text{exact}}\big) + \frac{1}{2}\operatorname{tr}\big(\rho_2 M_2^{\text{exact}}\big) \\ &= \frac{2+\sqrt{2}}{4} = p_{\text{upper}}^{\star}, \end{aligned}$$

$$p_{\text{lower}}^{\star} \le p_{\text{succ}}^{\star} \le p_{\text{upper}}^{\star} \quad \Longrightarrow \quad \boxed{p_{\text{succ}}^{\star} = \frac{2+\sqrt{2}}{4}}.$$

[2] Again, not always possible. But sometimes it works!

10. Outlook

What the certificate viewpoint unlocks

Lecture 2 in brief

- The dual problem can provide good (and tight) upper bounds.

Lecture 2 in brief

- The dual problem can provide good (and tight) upper bounds.

- Slater conditions ensure strong duality.

Lecture 2 in brief

- The dual problem can provide good (and tight) upper bounds.
- Slater conditions ensure strong duality.
- There are efficient numerical methods.

Lecture 2 in brief

- The dual problem can provide good (and tight) upper bounds.
- Slater conditions ensure strong duality.
- There are efficient numerical methods.
- Numerical solutions can sometimes be turned into computer-assisted proofs.

The SDP structure may be hidden

- State discrimination is **clearly an SDP**. For some problems, this is not at all obvious.

The SDP structure may be hidden

- State discrimination is **clearly an SDP**. For some problems, this is not at all obvious.
- Sometimes we must **“massage” the formulation** to recognise its SDP structure.

The SDP structure may be hidden

- State discrimination is **clearly an SDP**. For some problems, this is not at all obvious.
- Sometimes we must **"massage" the formulation** to recognise its SDP structure.
- If all else fails, **SDP hierarchies** may still provide systematic upper and/or lower bounds.

The SDP structure may be hidden

- State discrimination is **clearly an SDP**. For some problems, this is not at all obvious.
- Sometimes we must **“massage” the formulation** to recognise its SDP structure.
- If all else fails, **SDP hierarchies** may still provide systematic upper and/or lower bounds.
- This should make more sense during Jessica’s talk.

SDP in quantum information

- entanglement and steering
- measurement compatibility
- Bell nonlocality and correlations in networks
- quantum marginal problem
- channel discrimination
- quantum channel capacities
- quantum metrology
- higher-order computing, indefinite causality
- ...

Want more?

- Paul Skrzypczyk and Daniel Cavalcanti, *Semidefinite Programming in Quantum Information Science*;

Want more?

- Paul Skrzypczyk and Daniel Cavalcanti, *Semidefinite Programming in Quantum Information Science*;
- Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*;

Want more?

- Paul Skrzypczyk and Daniel Cavalcanti, *Semidefinite Programming in Quantum Information Science*;
- Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*;
- slides and lecture notes on `mtcq.github.io`.

Want more?

- Paul Skrzypczyk and Daniel Cavalcanti, *Semidefinite Programming in Quantum Information Science*;
- Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*;
- slides and lecture notes on `mtcq.github.io`.
- Feel free to just come and talk to me.

Want more?

- Paul Skrzypczyk and Daniel Cavalcanti, *Semidefinite Programming in Quantum Information Science*;
- Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*;
- slides and lecture notes on `mtcq.github.io`.
- Feel free to just come and talk to me.

Thank you! 🌠