QMA(2) Tutorial Part II: The Best Separable State problem

Cedric Lin
QuICS, UMD

July 28, 2016

Agenda

- Introduction to BSS
- SDP hierarchies
- The Sum-of-Squares (SoS) hierarchy
- Net-based methods

The Best Separable State (BSS) problem

Notation

- Set of unit vectors on $\mathbb{C}^d$: $B(d)$

  Set of unit vectors on $\mathcal{H}$: $B(\mathcal{H})$
- Set of density matrices on $\mathbb{C}^d$: $D(d)$

  Set of density matrices on $\mathcal{H}$: $D(\mathcal{H})$
- Set of separable density matrices on $\mathbb{C}^d \otimes \mathbb{C}^d$:

  $\mathrm{Sep}(d) \coloneqq \mathrm{conv}\{|\psi_1\rangle\langle\psi_1| \otimes |\psi_2\rangle\langle\psi_2| \colon |\psi_1\rangle, |\psi_2\rangle \in B(d)\}$

Best Separable State (BSS(ϵ))

- Given as input a $d^2 \times d^2$ Hermitian matrix M s.t. $0 \preceq M \preceq I$, compute
$$h_{\mathrm{Sep}(d)}(M) \coloneqq \max\{\mathrm{Tr}[M\rho] : \rho \in \mathrm{Sep}(d)\}$$
up to ϵ additive error. We denote this problem by BSS(ϵ).
- Can think of M as a measurement, and ρ as a separable state on $\mathbb{C}^d \otimes \mathbb{C}^d$; want to compute the maximum acceptance probability.

- Also known as the *support function* of $\mathrm{Sep}(d)$

Weak membership problem

- Given as input a density matrix $\rho \in D(d^2)$, and promised either

$$\rho \in \mathrm{Sep}(d), \qquad\qquad \text{OR}$$

$$\|\rho - \sigma\|_1 \geq \varepsilon \qquad \text{for all } \sigma \in \mathrm{Sep}(d)$$

  determine which is the case.
- Want to determine if ρ is separable or far from separable.

- Roughly equivalent to BSS (under poly reductions), e.g. [Liu07, Gha10]
  We'll only focus on BSS in this talk

Relationship with QMA(2)

- Note that $\mathrm{QMA}_{\log(d)}(2)$ reduces to computing $h_{\mathrm{Sep}(d)}(M)$.

  Most recent work on QMA(2) has been investigating h_{Sep}
- BSS(ϵ) is NP-hard if $\epsilon = 1/\operatorname{poly}(d)$ [Gurvits03]

  3−SAT$_{\widetilde{\Omega}(\log^2(d))}$- hard if $\epsilon = \Theta(1)$ [ABDFS09, HM10]

  So assuming ETH, computing $h_{\mathrm{Sep}(d)}(M)$ to constant precision requires $\exp(\widetilde{\Omega}(\log^2(d))) = d^{\widetilde{\Omega}(\log d)}$ time.
- A better algorithm for BSS could imply an upper bound for QMA(2)

Relationship with other problems

- Let $\mathrm{ProdSym}(d) \coloneqq \mathrm{conv}\{|\psi\rangle\langle\psi| \otimes |\psi\rangle\langle\psi| : |\psi\rangle \in B(d)\}$.
  Equivalently, $\mathrm{ProdSym}(d) = \{\rho \,:\, \rho F = F\rho = \rho, \rho \in D(d)\}$,
  where F is the operator that swaps the two systems.
- We can define a simpler equivalent version of BSS: compute
$$h_{\mathrm{ProdSym}(d)}(M) \coloneqq \max\{\mathrm{Tr}[M\rho] \,:\, \rho \in \mathrm{ProdSym}(d)\}$$
- Let $M' = |01\rangle\langle 01| \otimes M$. Then
$$h_{\mathrm{ProdSym}(2d)}(M') = h_{\mathrm{Sep}(d)}(M)/4$$
$$h_{\mathrm{Sep}(2d)}(M') = \left[1 + h_{\mathrm{ProdSym}(d)}(M)\right]/2$$

Relationship with other problems

- Consider the problem of approximating the $2 \to 4$ norm:

$$\|A\|_{2\to 4} := \max_{x\neq 0} \frac{\|Ax\|_4}{\|x\|_2}, \qquad \|y\|_p = (\Sigma_i |y_i|^p)^{1/p}$$

- This problem is equivalent to BSS under polynomial reductions.

  e.g. if $A = \Sigma_i \sqrt{p_i}|i\rangle\langle\psi_i|$, then

$$\|A\|_{2\to 4}^4 = \max_{|\psi\rangle\in B(d)} \Sigma_i p_i |\langle\psi_i|\psi\rangle|^2$$

$$= h_{\mathrm{ProdSym}(d)}\left(\Sigma_i p_i (|\psi_i\rangle\langle\psi_i|)^{\otimes 2}\right)$$

For the other direction, see [HM10], [BBHKSZ12]

Relationship with other problems

- Approximating the $2 \to 4$ norm is an important problem in classical computational complexity.
- In particular, the Small-Set Expansion hypothesis – a variant of the Unique Games Conjecture (UGC) – is equivalent to the assertion that $||A||_{2\to 4}$ is hard to approximate for a certain class of A's [BBHKSZ12].
- Studying BSS could give insight on possible refutations of UGC.

Semidefinite Programming (SDP) Hierarchies

Semidefinite program (SDP)

- A semidefinite program is an optimization problem of the form

$$\begin{aligned} \text{Maximize} \quad & \mathrm{Tr}[AX] \\ \text{Subject to} \quad & \Phi(X) = B \\ & X \succeq 0, \end{aligned}$$

where A, B are Hermitian matrices and Φ is a Hermiticity-preserving linear map.

SDPs can typically be solved in time polynomial in the dimension of the matrices A, B, and the logarithm of the desired accuracy $\log(1/\epsilon)$.

More general SDPs

- Linear and positive semidefinite constraints can be composed by using block matrices and extra slack variables. For example,

$$\begin{array}{ll} \text{Maximize} & \mathrm{Tr}[A_1 X_1 + A_2 X_2] \\ \text{Subject to} & X_1 + X_2 \preceq B,\ X_1 \succeq 0, X_2 \succeq 0 \end{array}$$

can be rewritten as

$$\begin{array}{ll} \text{Maximize} & \mathrm{Tr}\left[\begin{pmatrix} A_1 & & \\ & A_2 & \\ & & 0 \end{pmatrix}\begin{pmatrix} X_1 & & \\ & X_2 & \\ & & S \end{pmatrix}\right] \\ \text{Subject to} & X_1 + X_2 + S = B,\ \begin{pmatrix} X_1 & & \\ & X_2 & \\ & & S \end{pmatrix} \succeq 0 \end{array}$$

Some final comments on SDPs

- There are many equivalent formulations of SDPs;
  our definition follows Watrous
- Can define a *dual* problem to our original SDP (the *primal* problem).
  In general the optimal values of the problems are equal.
- SDPs are extremely common in the study of quantum complexity,
  e.g. quantum games, interactive proofs, communication/query complexity, …
- For more information on SDPs and their use in quantum information,
  see e.g. Jamie Sikora's lecture notes

k-extendable hierarchy [DPS04]

- A bipartite state ρ_{AB} is k-extendable if $\exists$ a state $\rho_{AB_1B_2\cdots B_k}$ s.t.

$$\rho_{AB_1} = \rho_{AB_2} = \cdots = \rho_{AB_k} = \rho_{AB}.$$

- We can optimize over k-extendable states using a SDP:

$$\begin{aligned} \text{Maximize} \quad & \operatorname{Tr}\left[M\rho_{AB_1}\right] \\ \text{Subject to} \quad & \operatorname{Tr}\left[\rho_{AB_1B_2\cdots B_k}\right] = 1 \\ & \rho_{AB_1} = \rho_{AB_2} = \cdots = \rho_{AB_k} \\ & \rho_{AB_1B_2\cdots B_k} \succeq 0 \end{aligned}$$

- This optimization takes $d^{O(k)}\text{polylog}(1/\epsilon)$ operations.

k-extendable hierarchy, general case

- A bipartite state ρ_{AB} is k-extendable if ∃ a state $\rho_{AB_1B_2\cdots B_k}$ s.t.

$$\rho_{AB_1} = \rho_{AB_2} = \cdots = \rho_{AB_k} = \rho_{AB}.$$

- Fact: ρ_{AB} is k-extendable for all k iff ρ^{AB} is separable.

  But how high does k need to be to get an ϵ-approximation?

- For the general case, $k = O(d/\epsilon)$ suffices.

  This is tight; an example can be given by using antisymmetric states
- So in general need $d^{O(d/\epsilon)}$ time. Not very good!

k-extendable hierarchy, 1-LOCC case

- Optimizing over k-extendable states takes $d^{O(k)}\mathrm{polylog}(\epsilon^{-1})$ time.
- [BCY10] show that $k = O(\log d\ /\epsilon^2)$ suffices when M is 1-LOCC. Hence $h_{\mathrm{Sep}(d)}(M)$ can be approximated in $\exp(O(\log^2 d/\epsilon^2))$ time

k-extendable hierarchy, 1-LOCC case

- Optimizing over k-extendable states takes $d^{O(k)}\mathrm{polylog}(\epsilon^{-1})$ time.
- [BCY10] show that $k = O(\log d\,/\epsilon^2)$ suffices when M is 1-LOCC.
  Hence $h_{\mathrm{Sep}(d)}(M)$ can be approximated in $\exp(O(\log^2 d/\epsilon^2))$ time

- Computing $h_{\mathrm{Sep}(d)}(M)$ requires $\exp\left(\widetilde{\Omega}(\log^2 d/\epsilon^2)\right)$ time
  for *general* M, assuming ETH [ABDFS09, HM10, BBHKSZ12].
- Coincidence?

k-extendable hierarchy, 1-LOCC case

- Optimizing over k-extendable states takes $d^{O(k)}\mathrm{polylog}(\epsilon^{-1})$ time.
- [BCY10] show that $k = O(\log d\,/\epsilon^2)$ suffices when M is 1-LOCC.
  Hence $h_{\mathrm{Sep}(d)}(M)$ can be approximated in $\exp(O(\log^2 d/\epsilon^2))$ time

- Computing $h_{\mathrm{Sep}(d)}(M)$ requires $\exp\left(\widetilde{\Omega}(\log^2 d/\epsilon^2)\right)$ time
  for *general* M, assuming ETH [ABDFS09, HM10, BBHKSZ12].
- Coincidence????????????

PPT criterion [Peres96, HHH96]

- Let $\rho \in D(\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_k)$be a k-partite system, and $S \subseteq \{1, \dots, k\}$.

  Let the matrix elements of ρ be $\rho_{r_1 \cdots r_k, s_1 \cdots s_k} = \langle r_1 | \cdots \langle r_k | \rho | s_1 \rangle \cdots | s_k \rangle$.
- The *partial transpose* ρ^{T_S} is given by switching the row and column indices of subsystems labelled by S.

  e.g. If $S = \{1,3\}$, then $\rho^{T_S}_{r_1 \cdots r_k, s_1 \cdots s_k} = \rho^{T_S}_{s_1 r_2 s_3 r_4 \cdots r_k, r_1 s_2 r_3 s_4 \cdots s_k}$
- If $\rho^{T_S} \succeq 0$ for all S, ρ satisfies the *Positive Partial Transpose* (PPT) criterion.
- Note that all separable states automatically satisfy the PPT criterion:

  e.g. if $\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j| \otimes |\phi_j\rangle\langle\phi_j|$ then

  $$\rho^{T_1} = \sum_j p_j |\psi_j^*\rangle\langle\psi_j^*| \otimes |\phi_j\rangle\langle\phi_j|$$

DPS hierarchy [DPS04]

- We can add the PPT criterion to the k-extendible hierarchy:

$$\begin{aligned}
\text{Maximize} \quad & \mathrm{Tr}\left[M\rho_{AB_1}\right] \\
\text{Subject to} \quad & \mathrm{Tr}\left[\rho_{AB_1B_2\cdots B_k}\right] = 1 \\
& \rho_{AB_1} = \rho_{AB_2} = \cdots = \rho_{AB_k} \\
& \rho_{AB_1B_2\cdots B_k} \succeq 0,\ \rho_{AB_1B_2\cdots B_k}^{T_S} \succeq 0 \text{ for all } S
\end{aligned}$$

- Can also require $\rho_{AB_1B_2\cdots B_k}$ supported on symmetric subspace of B subsystems, i.e. $F\rho_{AB_1B_2\cdots B_k} = \rho_{AB_1B_2\cdots B_k}F = \rho_{AB_1B_2\cdots B_k}$ for any permutation F on the B's.
- This is known as the DPS hierarchy [DPS04]

DPS hierarchy, cont'd

- $k = O(d/\epsilon)$ suffices, but this is not known to be tight.
  PPT defeats known problematic cases for k-extendable hierarchy.
- $k = \widetilde{\Omega}(\log d\ /\epsilon^2)$ levels required [HNW16, Wu-Monday]

- The dual of the DPS hierarchy gives entanglement witnesses:
  operators W such that $\mathrm{Tr}[W\rho] \geq 0$ for all $\rho \in \mathrm{Sep}(d)$

The Sum-of-Squares (SoS) hierarchy

Some simplifying assumptions

- Assume M is real and symmetric.
  Then only need to optimize over real ρ, since $\mathrm{Tr}[M\rho] = \mathrm{Tr}[M\rho^T]$ and $\rho' = (\rho + \rho^T)/2$ is real
- We'll consider
  $$h_{\mathrm{ProdSym}(d)}(M) \coloneqq \max\{\mathrm{Tr}[M\rho] : \rho \in \mathrm{ProdSym}(d)\}$$
  where $\mathrm{ProdSym}(d) \coloneqq \mathrm{conv}\{|\psi\rangle\langle\psi| \otimes |\psi\rangle\langle\psi| : |\psi\rangle \in B(d)\}$.
  Recall that this is equivalent to BSS.

Another way of looking at BSS

- Want to estimate, for a real matrix M,

$$h_{\mathrm{ProdSym}(d)}(M) \coloneqq \max\{\mathrm{Tr}[M\rho] : \rho \in \mathrm{ProdSym}(d)\}$$

- We can consider the optimal ρ to be given by a probability distribution over vectors in $B(d)$, i.e. a distribution over vectors

$$\Sigma_i x_i |i\rangle = (x_1, \cdots, x_d) \in \mathbb{R}^d, \qquad \Sigma_j x_j^2 = 1$$

- By definition, $\rho = \Sigma_{r_1,r_2,s_1,s_2} \mathbb{E}\left[x_{r_1} x_{r_2} x_{s_1} x_{s_2}\right] |r_1\rangle |r_2\rangle \langle s_1| \langle s_2|$, so

$$\rho_{r_1 r_2, s_1 s_2} = \mathbb{E}\left[x_{r_1} x_{r_2} x_{s_1} x_{s_2}\right] \text{ for our optimal } \rho$$

Another way of looking at BSS, cont'd

- By definition, $\rho_{r_1 r_2, s_1 s_2} = \mathbb{E}\left[x_{r_1} x_{r_2} x_{s_1} x_{s_2}\right]$ for our optimal ρ, and therefore

$$\begin{aligned} h_{\mathrm{ProdSym}(d)}(M) &= \mathrm{Tr}[M\rho] \\ &= \Sigma_{r_1, r_2, s_1, s_2} M_{r_1 r_2, s_1 s_2} \mathbb{E}\left[x_{r_1} x_{r_2} x_{s_1} x_{s_2}\right] \end{aligned}$$

- So we only need to optimize over a linear function of expectations of 4th order moments, satisfying $\Sigma_j x_j^2 = 1$.
- Problem: This optimization is hard to do.
- Instead of trying to optimize over valid expectations, we'll try to optimize over *pseudo-expectations*: functions that look like expectations because of the limited computational power of our algorithm.

Pseudo-expectations

- Let $\widetilde{\mathbb{E}}$ be a functional that maps a polynomial P over $\mathbb{R}^d$ of degree at most r to a real number $\widetilde{\mathbb{E}}[P]$. We say that $\widetilde{\mathbb{E}}$ is a *degree-k pseudo-expectation* if:
  1. Linearity: $\widetilde{\mathbb{E}}[\alpha P + \beta Q] = \alpha\widetilde{\mathbb{E}}[P]$+$\beta\widetilde{\mathbb{E}}[Q]$
  2. Positivity: $\widetilde{\mathbb{E}}[P^2] \geq 0$ if P has degree $\leq k/2$
  3. Normalization: $\widetilde{\mathbb{E}}[1] = 1$
- Can also demand our pseudo-expectations satisfy additional constraints.

  In this case want $\widetilde{\mathbb{E}}\left[(\Sigma_j x_j^2 - 1)^2\right] = 0$
- Since $\widetilde{\mathbb{E}}[Q^2] = 0$ implies $\widetilde{\mathbb{E}}[PQ] = 0$ (exercise!),

  this implies $\widetilde{\mathbb{E}}\left[\left(\Sigma_j x_j^2\right)P\right] = \widetilde{\mathbb{E}}[P]$ for any P with degree $\leq k/2$.

Example: Degree-4 Pseudoexpectations

- Consider the case $d = 2, k = 4$.
- Suffices to specify values of pseudoexpectation on monomials.

$$\begin{array}{c} \\ x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{array}\begin{array}{c} \begin{array}{cccc} \quad x_1^2 \quad & \quad x_1x_2 \quad & \quad x_2x_1 \quad & \quad x_2^2 \quad \end{array} \\ \begin{pmatrix} \widetilde{\mathbb{E}}[x_1^4] & \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] \\ \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] \\ \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] \\ \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] & \widetilde{\mathbb{E}}[x_1x_2^3] & \widetilde{\mathbb{E}}[x_2^4] \end{pmatrix} \end{array}$$

- Claim: This matrix is positive semidefinite.

  This follows from the positivity requirement.

$$\begin{array}{c} \\ x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{array} \begin{array}{c} \begin{array}{cccc} x_1^2 & x_1x_2 & x_2x_1 & x_2^2 \end{array} \\ \begin{pmatrix} \tilde{\mathbb{E}}[x_1^4] & \tilde{\mathbb{E}}[x_1^3x_2] & \tilde{\mathbb{E}}[x_1^3x_2] & \tilde{\mathbb{E}}[x_1^2x_2^2] \\ \tilde{\mathbb{E}}[x_1^3x_2] & \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1x_2^3] \\ \tilde{\mathbb{E}}[x_1^3x_2] & \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1x_2^3] \\ \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1x_2^3] & \tilde{\mathbb{E}}[x_1x_2^3] & \tilde{\mathbb{E}}[x_2^4] \end{pmatrix} \end{array}$$

- Claim: This matrix is positive semidefinite.
- Proof: for any $a, b, c, d \in \mathbb{R}$,

$$\begin{pmatrix} a & b & c & d \end{pmatrix} \begin{pmatrix} \tilde{\mathbb{E}}[x_1^4] & \tilde{\mathbb{E}}[x_1^3x_2] & \tilde{\mathbb{E}}[x_1^3x_2] & \tilde{\mathbb{E}}[x_1^2x_2^2] \\ \tilde{\mathbb{E}}[x_1^3x_2] & \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1x_2^3] \\ \tilde{\mathbb{E}}[x_1^3x_2] & \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1x_2^3] \\ \tilde{\mathbb{E}}[x_1^2x_2^2] & \tilde{\mathbb{E}}[x_1x_2^3] & \tilde{\mathbb{E}}[x_1x_2^3] & \tilde{\mathbb{E}}[x_2^4] \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

$$
\begin{array}{c} \\ x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{array}
\begin{array}{c} \begin{array}{cccc} x_1^2 & x_1x_2 & x_2x_1 & x_2^2 \end{array} \\
\begin{pmatrix}
\widetilde{\mathbb{E}}[x_1^4] & \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] \\
\widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] \\
\widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] \\
\widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] & \widetilde{\mathbb{E}}[x_1x_2^3] & \widetilde{\mathbb{E}}[x_2^4]
\end{pmatrix} \end{array}
$$

- Claim: This matrix is positive semidefinite.
- Proof: for any $a, b, c, d \in \mathbb{R}$,

$$
\begin{pmatrix} a & b & c & d \end{pmatrix}
\begin{pmatrix}
\widetilde{\mathbb{E}}[x_1^4] & \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] \\
\widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] \\
\widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] \\
\widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] & \widetilde{\mathbb{E}}[x_1x_2^3] & \widetilde{\mathbb{E}}[x_2^4]
\end{pmatrix}
\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}
$$

$$
= \widetilde{\mathbb{E}}[(ax_1^2 + bx_1x_2 + cx_2x_1 + dx_2^2)^2] \geq 0
$$

$$
\begin{array}{c} \\ x_1^2 \\ x_1x_2 \\ x_2x_1 \\ x_2^2 \end{array}
\begin{array}{cccc} x_1^2 & x_1x_2 & x_2x_1 & x_2^2 \\
\left(\begin{array}{cccc}
\widetilde{\mathbb{E}}[x_1^4] & \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] \\
\widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] \\
\widetilde{\mathbb{E}}[x_1^3x_2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] \\
\widetilde{\mathbb{E}}[x_1^2x_2^2] & \widetilde{\mathbb{E}}[x_1x_2^3] & \widetilde{\mathbb{E}}[x_1x_2^3] & \widetilde{\mathbb{E}}[x_2^4]
\end{array}\right)
\end{array}
$$

- We have a matrix $\tilde{\rho}$ where $\tilde{\rho}_{r_1r_2,s_1s_2} = \widetilde{\mathbb{E}}\left[x_{r_1}x_{r_2}x_{s_1}x_{s_2}\right]$.
- Since $\tilde{\rho} \succeq 0$ and $\mathrm{Tr}[\tilde{\rho}] = 1$, $\tilde{\rho}$ is a bipartite density matrix.
- $\tilde{\rho}$ is also invariant under any permutation of row and column indices,

  i.e. supported on symmetric subspace & invariant under partial transpose.
- Recall that $\rho_{r_1r_2,s_1s_2} = \mathbb{E}\left[x_{r_1}x_{r_2}x_{s_1}x_{s_2}\right]$ for our optimal ρ.

  Optimizing over $\tilde{\rho}$ is a (not-very-good) approximation to optimizing over ρ.

Degree-$2k$ pseudo-expectations

- For degree-$2k$ pseudo-expectations, can define k-partite state $\tilde{\rho}^{(k)}$ s.t. $\tilde{\rho}^{(k)}_{r_1 \cdots r_k, s_1 \cdots s_k} = \widetilde{\mathbb{E}}[x_{r_1} \cdots x_{r_k} x_{s_1} \cdots x_{s_k}]$.
- $\tilde{\rho}^{(k)}$ is invariant under permutation of all $2k$ indices, i.e. supported on symmetric subspace and invariant under partial transposes.
- To get pseudo-expectations of lower degree monomials we can "trace out" extra indices; e.g.

$$\begin{aligned} [\operatorname{Tr}_k \tilde{\rho}^{(k)}]_{r_1 \cdots r_{k-1}, s_1 \cdots s_{k-1}} &= \widetilde{\mathbb{E}}\left[x_{r_1} \cdots x_{r_{k-1}} x_{s_1} \cdots x_{s_{k-1}} \left(\Sigma_{r_k} x_{r_k}^2\right)\right] \\ &= \widetilde{\mathbb{E}}\left[x_{r_1} \cdots x_{r_{k-1}} x_{s_1} \cdots x_{s_{k-1}}\right] \end{aligned}$$

Recall the DPS hierarchy…

- Consider the $(k-1)$-th level of the DPS hierarchy.
- For optimal $\rho_{AB_1B_2\cdots B_{k-1}}$, can assume $\rho_{AB_1B_2\cdots B_{k-1}}^{T_S} = \rho_{AB_1B_2\cdots B_{k-1}}$ for all S; otherwise average the two.

- The DPS hierarchy optimizes over states invariant under permutations of all indices.
- So the DPS hierarchy actually optimizes over pseudo-expectations!

Sum-of-Squares (SoS) hierarchy

- In general, we can consider the following optimization problem:

  Maximize $\quad f(x)$

  Subject to $\quad g_1(x) = \cdots = g_m(x) = 0.$

  where $f, g_1, \cdots g_m$ are polynomials in $x = (x_1, \cdots, x_d)$
- A method of upper bounding the optimal value is to optimize over degree-k pseudo-expectations satisfying $\widetilde{\mathbb{E}}[(g_i(x))^2] = 0$.

  This is the Sum-of-Squares (SoS) hierarchy/Lasserre hierarchy.

  [Sho87, Par00, Nes00, Las01, Par03, BBHKSZ12]
- The DPS hierarchy can be seen as a special case of SoS, with the constraint $g(x) = \Sigma_j x_j^2 - 1 = 0$.

Advantages of SoS

- [HNW15, Natarajan-Wed] consider a modification of DPS by adding more constraints. This hierarchy converges *exactly* with $k = 2^{O(d^2)}$.
  This gives a $\exp(\mathrm{poly}(d))\mathrm{polylog}(1/\epsilon)$ algorithm for BSS(ϵ).
- SoS is a useful algorithm for a variety of approximation problems,
  e.g. computing graph expansions, machine learning, …
- SoS is one of the main approaches to refuting the Unique Games Conjecture.

Limitations of SoS

- In many cases there are proven bounds on how well SoS can do.
- These proofs often take the form of explicit examples:

  e.g. the optimal value is small, but the k-th level SoS outputs a large value for k insufficiently large.
- [HNW16, Wu-Monday] give such a proof for BSS($\Theta(1)$),

  establishing that $k = \widetilde{\Omega}(\log d)$ is necessary.

  This was previously known only assuming ETH.

Be careful of notation!

- What we use: d for # of variables, k for degree
- Optimization literature: n for # of variables, r for degree

Net-based methods

The low Schmidt rank case [SW12]

- Consider the problem of computing $h_{\mathrm{Sep}(d)}(M)$, where
$$M = \Sigma_{i=1}^{R} X_i \otimes Y_i$$
with $0 \preceq X_i, Y_i \preceq I$.

- Shi and Wu gave an net-based method that approximates $h_{\mathrm{Sep}(d)}(M)$ to ε precision in $O((R/\varepsilon)^R \times \mathrm{poly}(R, 1/\varepsilon))$ time.

The low Schmidt rank case, cont'd

- $h_{\mathrm{Sep}(d)}(M) = \max_{\rho_1,\rho_2} \Sigma_{i=1}^{R} \mathrm{Tr}[X_i \rho_1] \times \mathrm{Tr}[Y_i \rho_2]$.

  If we knew the optimal $(x_1 = \mathrm{Tr}[X_1 \rho_1], \cdots, x_R = \mathrm{Tr}[X_R \rho_1])$, then

$$h_{\mathrm{Sep}(d)}(M) = \max_{\rho_2} \Sigma_{i=1}^{R} x_i \mathrm{Tr}[Y_i \rho_2]$$

$$= \max_{\rho_2} \mathrm{Tr}[\bar{Y} \rho_2] = \|\bar{Y}\|_{\mathrm{op}}$$

  is easy to compute, where $\bar{Y} = \Sigma_{i=1}^{R} x_i Y_i$.

- So all we need to do is to find the optimal $(x_1, \cdots, x_R)$.

  But this seems hard to do…

Just try everything!

- Let $S = \{(\mathrm{Tr}[X_1\rho], \cdots, \mathrm{Tr}[X_R\rho]) : \rho \in D(d)\}$.
  Want to sample from points in S up to ϵ error
- Since $S \in [0,1]^R$, can try all points $\vec{p}$ from a grid with side length ϵ/R, and then test with a SDP solver if $\vec{p}$ is ϵ-close to S:

$$\vec{q} = (\mathrm{Tr}[X_1\rho], \cdots, \mathrm{Tr}[X_R\rho])$$
$$-a_i \le p_i - q_i \le a_i, \Sigma_{i=1}^{R} a_i \le \epsilon$$
$$\mathrm{Tr}[\rho] = 1, \quad 0 \preceq \rho \preceq I$$

- $O((R/\varepsilon)^R)$ points in the net.

The 1-LOCC case [BH15, Harrow-Monday]

- Consider the problem of computing $h_{\mathrm{Sep}(d)}(M)$, where
$$M = \Sigma_{i=1}^{R} X_i \otimes Y_i$$
with $\Sigma_{i=1}^{R} X_i = I$, and for all i, $X_i \succeq 0$, $0 \preceq Y_i \preceq I$.
- We can again try to use a covering net over $x_i = \mathrm{Tr}[X_i \rho_1]$.
Note in this case that $x_i \geq 0$, $\Sigma_{i=1}^{R} x_i = 1$
- Can think of $(x_1, \cdots, x_R)$ as being the PDF of a probability distribution.
Turns out it's easier to approximate PDFs…

Approximating distributions

- Let $\Delta_R(k) = \left\{\frac{e_{i_1}+e_{i_2}+\cdots+e_{i_k}}{k} : i_1, \cdots, i_k \in \{1, \cdots, R\}\right\}$.
- It suffices to consider points in $\Delta_R(k)$, with $k = O(\log d / \epsilon^2)$.

  Reason: can sample k points from distribution,

  get something close to true distribution with high prob.
- There are $R^{O(k)}$ points in $\Delta_R(k)$.

  So get an algorithm using $\approx R^{O(\log d/\epsilon^2)}$ operations.

The 1-LOCC case, cont'd

- In general, $R \gg d$. However can *sparsify* number of local terms, by a sampling procedure on the terms in the decomposition
- Get algorithm using $\approx d^{O\left(\log d/\epsilon^2\right)} = \exp(O(\log^2 d/\epsilon^2))$ operations. Matches that of SDP algorithm based on k-extensions!
- Downside: requires knowing explicitly a decomposition for M.

$$M = \Sigma_{i=1}^{R} X_i \otimes Y_i$$

- Often with a SoS-based algorithm, there is another algorithm based on covering nets with similar runtime. But see downside…

Open questions

- Coincidence between upper and lower bounds for BSS?
- Better algorithms / hardness results for BSS?
- Algorithms based on convex optimization? See [Poulin-Hastings, ’11]
- Connections between SoS and net-based methods?

Thanks!