Google Summer of Code 2022 : NumFOCUS Final Report on Improving CVXPY's Support for Quantum Information Modeling

Aryaman Jeendgar

October 24, 2022

Contents

1 Introduction:

This document details the contributions I made to *CVXPY's* codebase over the duration of my GSoC, '22 project titled: **Improve CVXPY's capabilities for quantum information modeling**. Largely speaking, the intent of my work was to implement the quadrature-based approximation methods introduced in the recent groundbreaking paper, Semidefinite approximations of the matrix logarithm (referred to as **FSP17** from now on). From a code perspective, this came down to reproducing the functionality provided in the CVXQUAD matlab package developed for the aforementioned paper by the authors within CVXPY. While a good chunk of the intended `Atom` and `Constraint` classes have been implemented, there is still some more work that remains to be done, I provide details about the same and more in this report.

2 Pre-GSoC period:

I made one significant PR to the CVXPY codebase before the GSoC period began, namely:

- #1740: Adding a new `Constraint` class, `FiniteSet`.

The PR added support for a new `Constraint` class named, `FiniteSet`, which allowed the user to deal with constraints over optimization variables which are to take on values from a finite set. It can be shown that such constraints can be reduced to linear inequality constraints over an auxiliary set of variables, and it is the implementation of this Canonicalization pathway that was the subject of this PR.

3 Community Bonding period:

This time was mostly spent coming up with a plan for tackling the project, and selecting what functionality we want to implement. Additionally, during this time I also started work on a slightly tangential PR, which was an implementation of the reduction for the **Von Neumann entropy** outlined in *proposition-4* of the paper Relative entropy optimization and its applications. I talk more about the particulars of this contribution in later sections.

4 Coding period:

4.1 Pre-midterm evaluation:

Before the midterm evaluation, I made the following PR's:

1. #1789: Implementation of the *exact* reduction for the Von Neumann entropy mentioned above.
2. #1833: Implementation of the quadrature approximation scheme outlined in FSP17 for the `ExpCone` class (under the guise of the so-called, Scalar Relative Entropy cone, which is equivalent to the exponential cone under a permutation and sign change of the arguments)

4.1.1 #1789, *Exact* `von_neumann_entr`:

The reduction for the `von_neumann_entr` implemented in this particular PR is *exact*, and hence isn't directly related to the goal of implementing the approximation methods mentioned above, but it was a great learning exercise for:

1. Getting used to implementing new `Atom` classes.
2. Learning some more Optimization theory, and most importantly
3. Working through the process of coming up with *verifiable* test cases (when none exist in literature) for any such functionality that I implement.

(3) in particular was really important since that is something we have to do on a per-case basis for every new `Constraint` and `Atom` class that we implement since there aren't a lot of "off-the-shelf" examples involving the constraint sets and functions that we are interested in implementing as a part of this work in the literature.

A write up which goes into much more technical detail about the process of implementing this `Atom` class may be found in this blogpost **HERE**.

4.1.2 #1833, a prelude to the *big* one:

This PR implements the reduction of the `ExpCone` class into a series of semidefinite constraints, which are motivated from the initial quadrature approximation of the logarithm. This contribution served the following purposes:

1. It was a great first step towards the upcoming `OpRelConeQuad` class, since here, we are just concerned with constraining variables that take on values in $\mathbb{R}_{++}$, hence, we get to see these complicated semidefinite constraints for a simpler particular case.
2. Some sub-functionality that had to be implemented for this PR carried over to all future implementations of this approximation scheme (such as the implementation of the `gauss_legendre` quadrature)
3. More interestingly, this also introduced an alternate canonicalization pathway for `ExpCone`, which is important in this case because it allows for the use of *Symmetric Cone Program* solvers (like CVXOPT) when the problem has exponential/logarithmic constraints.

Again, a writeup that goes into much more technical detail about the contribution can be found in the form of a blog-post, **HERE**

4.2 Post-midterm:

I made the following PR's in the post-midterm period:

1. #1875: In this PR I implemented the (Full) *Operator Relative entropy Cone*, which is the central ingredient for most of this work, since almost all specific functions (like the all important *Quantum Relative entropy*) can all be implemented using the `OpRelCone`
2. #1899: This PR implements the quadrature-based approximation for the `von_neumann_entr`, this was relatively straightforward since we already had a verified implementation (done in #1789) to check against.

Intuitively, #1875 is #1833, but where the inputs are general hermitian matrices (i.e. $\in \mathbb{H}_{++}$) instead of positive scalars. We first decided to work with the matrices with real inputs (i.e. in this case, *symmetric* matrices), and then later adapt the same for complex hermitian inputs (will be a part of the next PR).

4.3 Future work:

There are still some key features that we want to implement which are currently a `WiP`, these are:

1. Implement `quantum_rel_entr` (this is another atom, for whom a reduction may be derived using the approximation for the operator relative entropy cone) and make sure it works with Hermitian inputs.

2. Add a couple of example notebooks reproducing a series of examples provided in Efficient optimization of the quantum relative entropy and FSP17.

3. Add dual variable recovery for the approximations to the scalar and operator relative entropy cones, i.e. compute the dual representation of each of the variables involved in the semidefinite representation of $K^n_{m,k}$, namely, the T_i's ($i = 1, \dots, m$) and Z_j ($j = 0, \dots, k$) and (X, Y, T). This is challenging because of the large number of variables involved in this representation (i.e. $m + k + 3$). This would also open up the possibility of using CVXPY verifying accuracy for the approximation after solving a problem, especially in the cases when ***Strong Duality*** holds. Since this would end up being a nice mathematical contribution, there is a chance of it leading to a paper!

5 Executive summary:

Type	Name of `CVXPY` class	`CVXQUAD` equivalent
Implemented	`OpRelConeQuad`	`op_rel_entr_epi_cone.m`
Functionality	`von_neumann_entr`	`quantum_entr.m`
	`RelEntrConeQuad`	-
`WiP`	`quantum_rel_entr`	`quantum_rel_entr.m`
	`trace_logm`	`trace_logm.m`
Not started	-	`lieb_ando.m`
	-	`matrix_geo_mean_hypo_cone.m`
	-	`matrix_geo_mean_epi_cone.m`

5.1 Implemented functionality:

5.1.1 `OpRelConeQuad`:

The Operator Relative entropy cone is defined as:

$$K^n_{re} = \text{cl}\{(X, Y, T) \in \mathbb{H}^n_{++} \times \mathbb{H}^n_{++} \times \mathbb{H}^n : T \succeq D_{op}(X||Y)\}$$

Which is approximated by the below cone in FSP17:

$$K^n_{m,k} = \{(X, Y, T) \in \mathbb{H}^n_{++} \times \mathbb{H}^n_{++} \times \mathbb{H}^n : T \succeq -P_{r_{m,k}}(Y, X)\} \qquad (1)$$

Where, $P_{r_{m,k}}$ is the non-commutative perspective of the function $r_{m,k}$ — $r_{m,k}$ is defined in equation (11) of FSP17.

As per *theorem*- 3 of FSP17, $K^n_{m,k}$ has the following representation that was shown to be "ϵ -good" in *theorem*- 7 of the same:

Theorem 1 *A triple of matrices (X, Y, T) belongs to $K^n_{m,k}$ if and only if*

$$\begin{cases} Z_0 = Y, & \begin{bmatrix} Z_i & Z_{i+1} \\ Z_{i+1} & X \end{bmatrix} \succeq 0 \quad (i = 0, \ldots, k-1) \\ \sum_{j=1}^{m} w_j T_j = -2^{-k} T, & \begin{bmatrix} Z_k - X - T_j & -\sqrt{t_j} T_j \\ -\sqrt{t_j} T_j & X - t_j T_j \end{bmatrix} \succeq 0 \quad (j = 1, \ldots m) \end{cases}$$

holds for some $T_1, \ldots, T_m, Z_0, \ldots, Z_k \in \mathbb{H}^n$.

5.1.2 `von_neumann_entr`:

As discussed earlier, the Von Neumann Entropy has been implemented via two distinct canonicalization pathways, I'll summarize their mathematical descriptions below:

1. *Exact*: I reproduce *Proposition-4* from Relative entropy optimization and its applications here:

   Theorem 2 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function that is invariant under permutation of its argument, and let $g : \mathbb{S}^n \to \mathbb{R}$ be the convex function defined as $g(N) = f(\lambda(N))$. Here $\lambda(\boldsymbol{N})$ refers to the list of eigenvalues of the matrix $\boldsymbol{N}$. Then we have that:*

   $$g(\boldsymbol{N}) \leq t$$
   $$\Updownarrow$$
   $$\exists \boldsymbol{x} \in \mathbb{R}^n s.t. \quad \begin{aligned} & f(\boldsymbol{x}) \leq t \\ & \boldsymbol{x}_1 \geq \cdots \geq \boldsymbol{x}_n \\ & s_r(\boldsymbol{N}) \geq \boldsymbol{x}_1 + \cdots + \boldsymbol{x}_r, r = 1, \ldots, n-1 \\ & Tr(\boldsymbol{N}) = \boldsymbol{x}_1 + \cdots + \boldsymbol{x}_n \end{aligned}$$

   where, $s_r(\boldsymbol{N})$ is the sum of the r largest eigenvalues of $\boldsymbol{N}$

   The above is the representation that I implemented for the exact `von_neumann_entr`

2. Quadrature approximation: The epigraph of `von_neumann_entr` can be expressed in terms of the Operator Relative entropy cone, the basis for which is the observation that:

   $$S = \mathrm{tr}(D_{op}(X||I)) \tag{2}$$

Where, D_{op} is the operator relative entropy defined as the *noncommutative perspective* of the negative logarithm, specifically:

$$D_{op}(X||Y) := -X^{1/2}\log(X^{-1/2}YX^{-1/2})X^{1/2}$$

From the above representation, a very natural representation of the `von_neumann_entr` in terms of `OpRelConeQuad` follows, i.e.:

$$(N,T) \in \{(X,Q) : Q \succeq S\} \iff (N,I,T) \in K^n_{re}$$

And hence, the approximate, semidefinite representation given in *Theorem-1* holds.

5.1.3 `RelEntrConeQuad`:

From a mathematical perspective, the underlying representation is equivalent to one for $K^1_{m,k}$ (where $K^n_{m,k}$ is defined above in section 5.1.1).

5.2 Work in Progress:

5.2.1 `quantum_rel_entr`:

This can be characterized via *Corollary-1* in FSP17 (reproduced below), which again draws upon a representation of the Relative Quantum entropy in terms of the Operator relative entropy, which turns out to be:

$$D(A||B) = \phi(D_{op}(A\otimes I, I\otimes \bar{B}))$$

Further, the epigraph of the QRE and the Operator Relative entropy cone may be related as follows:

Theorem 3 *For any $A, B \succ 0$ and $\tau \in \mathbb{R}$, we have:*

$$\phi(D_{op}(A\otimes I, I\otimes \bar{B})) \le \tau \iff \exists T \in \mathbb{H}^{n^2} : (A\otimes I, I\otimes \bar{B}, T) \in K^{n^2}_{re} \text{ and } \phi(T) \le \tau$$

This is currently a `WiP`, because of some issues that I have been facing in getting with a disagreement in output between CVXQUAD and my implementation — further, this also needs to be adapted to allow for complex (Hermitian) inputs. This can be done via a proper initialization with the `Complex2Real` class (there are currently some circular import errors that I have been running into when the initialization is done for both the `Complex2Real` and `Dcp2Cone` classes)

5.2.2 `trace_logm`:

The `trace_logm` is also fairly similar to the `von_neumann_entr` in terms of its representation in terms of the Operator Relative entropy, namely we have:

$$\texttt{trace\_logm}(X, C) = -\operatorname{tr}\left(C \cdot D_{op}(I||X)\right)$$

It is currently also a `WiP`, with it's canonicalization procedure written and ready, since I haven't thought of any test cases for the same as of yet.

5.3 Not started yet:

1. `lieb_ando`
2. `matrix_geo_mean_[hypo|epi]_cone`