Projection Methods for Quantum Channel Construction

Henry Wolkowicz

(work with: Yuen-Lam Vris Cheung, Dmitriy Drusvyatskiy,
Chi-Kwong Li, Diane Christine Pelejo)

Dept. Combinatorics and Optimization
University of Waterloo

At: Quantum Optimization Workshop
Fields Institute, October 27-29, 2014

FIELDS

Outline

- Find a quantum channel, if it exists, that maps a given set of quantum states to another given set of quantum states
- Use (for large scale):
  MAP, Method of alternating projections
  DR, Douglas-Rachford reflect-reflect-average
- find low rank solutions
- find high rank solutions

Reference

http://arxiv.org/abs/1407.6604

Motivation: Quantum Channels/Quantum States

Basic problem in quantum information science

construct, if it exists, a quantum channel T that
maps given sets of quantum states

$$T : \{\rho_1, \ldots, \rho_k\} \longrightarrow \{\hat{\rho}_1, \ldots, \hat{\rho}_k\}$$

- Quantum channel: trace preserving completely positive linear map — $T : \mathcal{M}^n \to \mathcal{M}^m$ (e.g. Choi '75)

$$T(X) = \sum_{j=1}^{r} F_j X F_j^*, \text{ for some } F_j, m \times n, \sum_{j=1}^{r} F_j^* F_j = I_n$$

- Quantum states: $\rho_j, \hat{\rho}_i$ are density matrices A_j, B_i
  Hermitian, positive semidefinite, trace 1.

References:

Choi, Fung/C.K.Li/, Huang/Poon, etc...

Background/Notation/Problem Formulation

Given density matrices: $A_j \in \mathcal{M}^n, B_i \in \mathcal{M}^m$

Find a completely positive linear map T with

$$T(A_i) = B_i, \quad \text{trace}(A_i) = \text{trace}(B_i), \qquad i = 1, \ldots, k.$$

Celebrated Choi matrix, $C(T)$

$E_{ij} = e_i e_j^T$ *standard orthonormal basis* of $\mathcal{M}^n$; $P_{ij} \in \mathcal{M}^m$;
T is trace preserving, completely positive, TPCP, iff the following block matrix is positive semidefinite ($\succeq 0$)! and trace preserving constraints $\text{trace}(P_{ij}) = \delta_{ij}$ hold (δ_{ij} Kronecker delta)

$$C(T) := \begin{bmatrix} P_{11} & \ldots & P_{1n} \\ \vdots & P_{ij} & \vdots \\ P_{n1} & \ldots & P_{nn} \end{bmatrix} := \begin{bmatrix} T(E_{11}) & \ldots & T(E_{1n}) \\ \vdots & T(E_{ij}) & \vdots \\ T(E_{n1}) & \ldots & T(E_{nn}) \end{bmatrix} \succeq 0$$

Equivalent Feasibility Problem for $C(T)$

By abuse of notation:

$T(A_\ell) = T\left(\sum_{ij}(A_\ell)_{ij}E_{ij}\right) = \sum_{ij}(A_\ell)_{ij}T(E_{ij}) = \sum_{ij}(A_\ell)_{ij}P_{ij}$

Positive semidefinite feasibility problem for $P = (P_{ij})$

feas. prob.

$$\left\{ \begin{array}{ll} \sum_{ij}(A_\ell)_{ij}P_{ij} = B_\ell, & \ell = 1, \ldots, k \\ \operatorname{trace}(P_{ij}) = \delta_{ij}, & 1 \le i \le j \le n \\ P \in \mathcal{H}_+^{nm} & \text{Hermitian psd } nm \times nm \end{array} \right\}$$

$\mathcal{H}^{nm}$ inner product space of Hermitians over reals, Frobenius norm

- $\operatorname{rank}(P)$ - minimal number of summands in representation $T(X) = \sum_{j=1}^{r} F_j X F_j^*$
- from compactness, feas. prob. is never weakly infeasible
- $m = n = 100$ implies number complex variables is $10^8/2$

MAP Algorithm

Method of alternating projections

Given $A, B \subseteq \mathcal{E}, x \in \mathcal{E}$ (Euclidean space)

$$\mathrm{proj}_A(x) = \operatorname{argmin}_{a \in A}\{\|x - a\|\} \text{ nearest point set}$$

A, B closed convex implies $\mathrm{proj}_A(x), \mathrm{proj}_B(x)$ are singletons.
iterates the following two steps

$$\begin{aligned} &\text{choose} \quad b_l \in \mathrm{proj}_B(a_l) \\ &\text{choose} \quad a_{l+1} \in \mathrm{proj}_A(b_l) \end{aligned}$$

When relative interiors intersect, then convergence is R-linear; rate governed by cosines of angles between vectors $a_{l+1} - b_l$ and $a_l - b_l$.

DR Algorithm

Douglas-Rachford reflect-reflect-average

given $x \in \mathcal{E}$

$$\mathrm{refl}_A(x) = \mathrm{proj}_A(x) + (\mathrm{proj}_A(x) - x), \qquad \text{reflection operator}$$

iterate

$$x_{l+1} = \frac{x_l + \mathrm{refl}_A(\mathrm{refl}_B(x_l))}{2}.$$

convergence known for convex instances; rate of convergence not well-understood.

- for MAP, DR to be effective, nearest point mappings proj_A and proj_B must be easy to evaluate.
- for quantum channel construction problem, mappings are easy to compute (especially projection onto affine subspace).

Solve for P Hermitian psd and in Linear Manifold

Recall: feasibility problem

$$\left\{\begin{array}{ll} \sum_{ij}(A_\ell)_{ij}P_{ij} = B_\ell, & \ell = 1,\ldots,k \\ \text{trace}(P_{ij}) = \delta_{ij}, & 1 \leq i \leq j \leq n \\ P \in \mathcal{H}_+^{nm} & \text{Hermitian psd } nm \times nm \end{array}\right\}$$

define linear mappings and vectors

$$\mathcal{L}_A(P) := \left(\sum_{ij}(A_\ell)_{ij}P_{ij}\right)_\ell \quad \text{and} \quad \mathcal{L}_T(P) = \left(\text{trace}(P_{ij})\right)_{i\leq j},$$

$$\mathcal{L}(P) = (\mathcal{L}_A(P), \mathcal{L}_T(P)).$$

$$B = (B_1,\ldots,B_k) \quad \text{and} \quad \Delta = (\delta_{ij})_{i\leq j}.$$

Projection onto Semidefinite Cone $\mathrm{proj}_{\mathcal{H}_+^{mn}}(P)$

Eckart-YoungTheorem

$P = U\,\mathrm{Diag}(\lambda_1, \ldots, \lambda_{mn})U^*$ (unitary) eigenvalue decomposition

$$\mathrm{proj}_{\mathcal{H}_+^{mn}}(P) = U^*\,\mathrm{Diag}(\lambda_1^+, \ldots, \lambda_{mn}^+)U,$$

$r^+ = \max\{0, r\}$

- requires a single eigenvalue decomposition (efficient, accurate)
- can use Lanczos (Krylov method) if low rank projection is desired

Projection onto Linear Manifold

$\mathcal{A} := \{P \in \mathcal{H}^{nm} : \mathcal{L}(P) = (B, \Delta)\}$

$$\min\left\{\frac{1}{2}\|P - \hat{P}\|^2 : \mathcal{L}(\hat{P}) = (B, \Delta)\right\}.$$

Classically,

$$\text{proj}_{\mathcal{A}}(P) = P + \mathcal{L}^{\dagger}R,$$

$\mathcal{L}^{\dagger}$ Moore-Penrose generalized inverse
$R := (B, \Delta) - \mathcal{L}(P)$ is residual at current P.

- Finding Moore-Penrose generalized inverse usually time consuming and error prone.
- Luckily, we can exploit special structure;
- time to compute the projection onto $\mathcal{A}$ is negligible compared to computational effort for eigenvalue decompositions.

Exploiting Structure

$\text{sHvec}(A_k)$ denotes vectorization of A_k with respect to a certain basis of $\mathcal{H}^{nm}$.
construct special matrix $M \in \mathbb{R}^{k \times m^2}$

$$M^T = \left[\text{sHvec}(A_1) \quad \text{sHvec}(A_2) \quad \dots \quad \text{sHvec}(A_k)\right].$$

separate M into three blocks

$$M = \left[M_{Re} \quad M_{Im} \quad M_D\right]$$

where: $M_D \in \mathbb{R}^{k \times m}$ rows formed from diagonals of A_i
M_{Re}, M_{Im} rows from real, imaginary parts A_i, resp. $i = 1, \dots, k$.

Further Structure

$$
\begin{aligned}
M_{ReImD} &:= \begin{bmatrix} M_{Re} & -M_{Im} & M_D \end{bmatrix} \\
N_{\Re\Im D} &:= \left[\frac{1}{\sqrt{2}} \begin{bmatrix} M_{\mathbb{R}} & M_{Re} & -M_{Im} & -M_{Im} \\ -M_{Im} & M_{Im} & -M_{Re} & M_{Re} \end{bmatrix} \begin{bmatrix} M_D & 0 \\ 0 & M_D \end{bmatrix} \right]
\end{aligned}
$$

Permute rows and columns of $N_{\Re\Im D}$ we obtain matrix N_{final}.

Special simplified structured L

$$
L := \begin{bmatrix} I_{t(n-1)} \otimes N_{final} & 0 \\ 0 & \begin{bmatrix} \begin{bmatrix} I_{n-1} \otimes M_{ReImD} & 0_{k(n-1),n^2} \end{bmatrix} \\ \begin{bmatrix} e_n \otimes I_{n^2} \end{bmatrix}^T \end{bmatrix} \end{bmatrix},
$$

$\otimes$ Kronecker product; triangular number $t(n-1) = \frac{n(n-1)}{2}$.

Simplified $L^\dagger$ and Vectorization

let $(M_{ReImD})_{null}$ have orthonormal columns for a basis of $\mathrm{null}(M_{ReImD})$, i.e., $\mathrm{null}(M_{ReImD}) = \mathrm{range}((M_{ReImD})_{null})$

$$L^\dagger = \begin{bmatrix} I_{t(n-1)} \otimes N_{final}^\dagger & 0 \\ 0 & \begin{bmatrix} I_{n-1} \otimes M_{ReImD}^\dagger & e_{n-1} \otimes (M_{ReImD})_{null} \\ e_{n-1}^T \otimes -M_{ReImD}^\dagger & I_{n^2} - (n-1)(M_{ReImD})_{null} \end{bmatrix} \end{bmatrix},$$

- $L^\dagger$ easy to construct by simply stacking various small matrices together in blocks.
- both evaluations Lp and $L^\dagger R$ can be *vectorized* and evaluated efficiently and accurately in MATLAB.

Numerical Tests

Computers:

Large Problems

AMD Opteron(tm) Processor 6168, 1900.089 MHz cpu running LINUX

Smaller Problems

Optiplex 9020, Intel(R) Core(TM), i7-4770 CPUs, 3.40GHz,3.40 GHz, RAM 16GB running Windows 7.

For simplicity:

- $m = n$
- unital constraint $T(I_n) = I_n$ is assumed.

Generating Instances

Generate Choi matrix P

- random unitary matrices $F_i, i = 1, \ldots, r$; and positive probability distribution d, $\quad P = \sum_{i=1}^r d_i F_i F_i^*$
- given density matrix X, TPCP evaluation $T(X)$ can be done using blocked form $T(X) = \sum_{ij} X_{ij} P_{ij}$

generate random density matrices $A_i, i = 1, \ldots, k$

set B_i as image of TPCP T
(guarantees a feasible instance of rank r)

m=n,k,r	iters	norm-residual	max-cos	PSD-proj-CPUs
90,50,90	6	5.88e-15	.7014	233.8
100,60,90	7	7.243e-15	0.8255	821.7
110,65,90	7	7.983e-15	0.8222	1484
120,70,90	8	8.168e-15	0.8256	2583
130,75,90	8	7.19e-15	0.8288	3607
140,80,90	9	8.606e-15	0.8475	5832
150,85,90	11	8.938e-15	0.8606	6188
160,90,90	11	9.295e-15	0.8718	1.079e+04
170,95,90	12	9.412-15	0.8918	1.139e+04

Table: Using DR algorithm; for solving huge problems

- $m = n = 10^2$ finds PSD matrix order 10^4; approximately $10^8/2$ variables; high accuracy
- CPU time depends approximately linearly in the size $m = n$.

High Rank Solutions

Apply DR algorithm (more efficient than MAP here)

- current feasible solution $P_c = mnI_{mn}$ barycenter of all the feasible points currently found.
- change starting point to *other side and outside* PSD cone using direction $d = mnI_{mn} - \operatorname{trace}(P_c)P_c$; new starting point $P_n := P_c + \alpha d$ (not PSD)
- repeat till $P \succ 0$ or no increase in rank occurs

Using DR for Max Rank Problems

Using DR algorithm; with
$[m\ n\ k\ mn\ toler\ iterlimit] = [30\ 30\ 16\ 900\ 1e-14\ 3500]$;
max/min/mean iter and number rank steps for finding max-rank of P. The 3500 here means 9 decimals accuracy attained for last step.

	rank steps	min-iters	max-iters	mean-iters	max-cos	max rank
r=30	1	6	6	6	7.008801e-01	900
r=28	1	7	7	7	7.323953e-01	900
r=26	1	7	7	7	7.550174e-01	900
r=24	1	8	8	8	7.911440e-01	900
r=22	1	9	9	9	8.238539e-01	900
r=20	1	9	9	9	8.454781e-01	900
r=18	1	11	11	11	8.730321e-01	900
r=16	1	15	15	15	8.995266e-01	900
r=14	1	23	23	23	9.288445e-01	900
r=12	8	194	3500	1.916375e+03	9.954262e-01	900
r=10	9	506	3500	2.605778e+03	9.968120e-01	900
r=8	12	2298	3500	3.350833e+03	9.986002e-01	900

m=n,k	initial rank r	facial red. ranks	final rank	final norm-residual
12,10	11	100,50,44,39	39	1.836e-15
12,10	10	92,61,43,44	44	1.786e-15
20,14	20	304,105,71	71	9.648e-15
22,13	20	374,121,75	75	9.746e-15

Table: Using MAP algorithm with facial reduction for decreasing the rank

$m = n, k$	initial rank r	starting constr. rank r_s	final constr. rank r_f
12,9	15	20	7
25,16	35	45	19
30,21	38	48	27

Table: Using DR algorithm for rank constrained problems with ranks r_s to r_f nonconvex problem

Summary

- We have used MAP and DR to accurately construct quantum channels
- solved huge semidefinite problems accurately (DR more efficient than MAP)
- found high rank solutions (DR more efficient than MAP)
- found low rank solutions (MAP better for facial reduction; DR better for nonconvex rank constrained problem)

Projection Methods for Quantum Channel Construction

Henry Wolkowicz

(work with: Yuen-Lam Vris Cheung, Dmitriy Drusvyatskiy, Chi-Kwong Li, Diane Christine Pelejo)

Dept. Combinatorics and Optimization
University of Waterloo

At: Quantum Optimization Workshop
Fields Institute, October 27-29, 2014