Assignment 4

Due date: Wednesday, March 21

Only do three of the problems and respond to the short survey questions at the end.

Quantum state tomography

Use RRR to reconstruct a pure-state density matrix X from a set of measurements $\mu_1, \mu_2, \ldots$ of operators $H_1, H_2, \ldots$. As in the last assignment, your system is comprised of four qubits labeled 1-4. The values $\mu_1, \ldots, \mu_{12}$ of the single qubit polarizations are as follows:

	σ_x	σ_y	σ_z
1	$-1/4$	0	0
2	$-1/4$	0	0
3	0	0	0
4	$1/4$	$1/4$	0

You have also measured the following products of two-qubit polarizations, measurements $\mu_{13}, \ldots, \mu_{39}$:

	σ_x^2	σ_y^2	σ_z^2	σ_x^3	σ_y^3	σ_z^3	σ_x^4	σ_y^4	σ_z^4
σ_x^1	0	$1/4$	$-1/4$	$-1/4$	$1/4$	0	0	0	0
σ_y^1	0	$1/4$	0	$-1/4$	$1/4$	$3/4$	0	0	$1/4$
σ_z^1	0	$-1/4$	0	$-1/4$	$-1/4$	0	0	0	0

For one of the RRR constraints, use the rank-1 (pure-state) constraint on X. The 40 linear equations $\mathrm{Tr}X = 1$, $\mathrm{Tr}XH_i = \mu_i$, $i = 1, \ldots, 39$ will serve as the other RRR constraint. Report your reconstruction of $X_{ij} = x_i^* x_j$ by its first row, $X_{1j} = x_1^* x_j$, since this has no phase ambiguity. Order the 16 qubit basis states as in the previous assignment.

Complex Hadamard matrices

To what extent does knowledge of the magnitudes of all the elements of a unitary matrix determine the matrix itself? For example, do magnitude measurements of the 3×3 CKM[1] matrix elements of particle physics determine this fundamental unitary "mixing" matrix?

As an experiment, consider the case of $n \times n$ unitary matrices U that have equal-magnitude elements. These are called complex Hadamard matrices because they generalize the standard Hadamard matrices, where all elements are real and equal to $\pm 1/\sqrt{n}$. The Fourier matrices $F_{pq} = \exp(2\pi i\, p\, q/n)/\sqrt{n}$ show there exists at least one complex Hadamard for each n. Note that multiplying rows or columns of U by arbitrary phases does not change the magnitudes while keeping U unitary. We therefore do not count these phase modifications as different and adopt the "Fourier convention", where phases are applied to make the first row and column have all elements equal to $1/\sqrt{n}$.

Use RRR to find 7×7 complex Hadamard matrices. Apply the Fourier phase convention and check if you are finding matrices other than the Fourier matrix (or some permutation applied to its rows or columns).

Antipodal kissing spheres

A configuration of $2k$ antipodal kissing spheres in dimension d is a set of vectors $v_1, \ldots, v_k$ in $\mathbb{R}^d$ satisfying:

$$v_i \cdot v_j = 1, \quad i = j, \tag{1}$$

$$|v_i \cdot v_j| \leq \frac{1}{2}, \quad i \neq j. \tag{2}$$

The $2k$ unit-diameter spheres centered at $\pm v_1, \ldots, \pm v_k$ would "kiss" a unit-diameter sphere at the origin and not intersect each other.

Use RRR to find such configurations by searching for symmetric "dot-product" matrices X subject to the rank-d constraint $X_{ij} = v_i \cdot v_j$ and the element-wise equality and inequality constraints implied by (1) and (2).

Check your program for the case $d = 2$, $k = 3$, where you know the answer. Next, try $d = 3$ and $k = 6$ (the maximum possible). The solutions in this case are not rigid — a quirk of three dimensions — so expect to find different dot products in different runs. Finally, see if you can find $k = 12$ and $k = 20$ configurations in $d = 4$ and $d = 5$, both of which are rigid.

[1] Cabibbo-Kobayashi-Maskawa

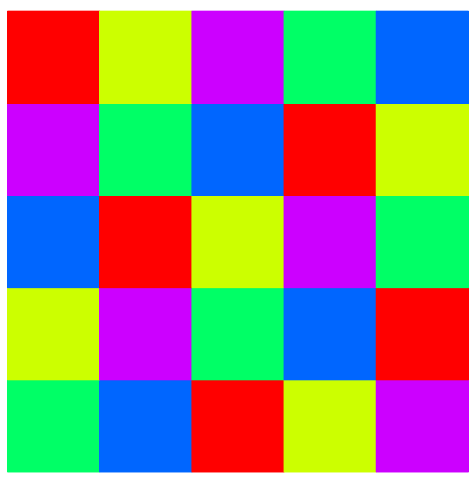
There are two other standard formulations of this problem that can be used. Since the programming for these is slightly more involved, only define the variables and describe (in words and a few equations) the projections to the two constraint sets.

Start with the divide-and-concur formulation, where variables are suitably replicated to enable independent easy projections to the "divide" constraint, and are forced to agree by the "concur" constraint.

In the linear-relation formulation, secondary variables are introduced that are related to the primary variables $(v_1, \ldots, v_k)$ by linear equations, and one of the RRR projections is to that linear relation. For the other RRR constraint you can independently impose constraints on the primary and secondary variables.

Coloring the queens graph

Suppose n^2 queens are placed on an $n \times n$ checkerboard. Can they be colored with k colors so that no two attacking queens have the same color? Here is a $k = 5$ (the minimum possible) solution for $n = 5$:



Use RRR to find colorings of queens graphs. Use either the tricky rank-$(k-1)$ formulation on symmetric matrices from lecture, divide-and-concur, or the method of linear relations. Remember that your Euclidean space encoding of color should have complete permutation symmetry. For example, to encode red/green/blue ($k = 3$) you could use the three vertices of a centered equilateral triangle in a plane, or even the 3D encoding $[1, 0, 0]$, $[0, 1, 0]$, $[0, 0, 1]$.

Try to find colorings for $n = 6$ and $n = 7$, both with $k = 7$ (the minimum possible).

Clique finding

For any pair $i, j \in V$ of vertices in a graph $G(V, E)$, denote by $(i, j) \in E$ the presence of an edge between those vertices. A k-clique of G is a subset $C \subset V$ of size k with the property $(i, j) \in E$ for all distinct pairs $i, j \in C$.

Finding k-cliques is a very natural application of RRR with rank constrained symmetric matrices. Let c be the indicator variable for the k-clique:

$$\forall i \in V: \quad c_i = \begin{cases} 1, & i \in C \\ 0, & i \notin C. \end{cases}$$

Define a symmetric rank-1 matrix, as usual, by $X_{ij} = c_i c_j$. We encode the graph using the symmetric matrix H defined by

$$\forall i, j \in V: \quad H_{ij} = \begin{cases} 1, & (i, j) \in E \\ 0, & (i, j) \notin E. \end{cases}$$

With symmetric matrices X as the search variable, one of the RRR constraints, A, is that X is semi-positive definite and rank 1. The other constraint, B, is the set

$$\begin{aligned} \mathrm{Tr} X &= k \\ \mathrm{Tr} XH &= k(k-1) \\ X_{ij} &\in \{0, 1\}, \quad \forall i, j \in V. \end{aligned}$$

The third property makes computing the projection P_B very easy. For example, in order to satisfy the first condition as well, we simply find the k largest elements on the diagonal of X, set them to 1, and the remainder to zero. A similarly simple operation (for you to work out) applies to the off-diagonal elements.

Try out your clique finder on the 125-vertex benchmark instance `C125.9.clq` located in the data directory of the github site. Ignore the lines in the header and go directly to the listing of the 6963 edges: `e 2 1`, `e 3 1`, ... Start with $k = 25$ and see how much you can increase k before you stop finding solutions. List the k vertices of the largest clique you find.

Ramsey numbers

Ramsey numbers can be thought of as generalizing the pigeonhole principle. Suppose you have complete graph on n vertices and color the edges with three colors. If $n \geq 17$ it turns out that no matter how you color the edges there will always be a mono-chromatic triangle in the graph (all three edges have the same color). Conversely, when $n < 17$ there is a way to color the edges so monochromatic triangles are absent. In this problem you will use RRR to find such edge colorings.

Because there is a constraint, on each triangle, among three — not two — primary variables (edge colors), the linearizing trick of using symmetric matrices cannot be used. You should therefore try one of the two other standard formulations: divide-and-concur or linear relations. Use permutation symmetric, color-to-vector encodings as described in the queens graph problem.

For divide-and-concur you would use color-vector variables with two indices v_{et}, where e is an edge of the complete graph and t is one of the triangles that contains edge e. Interpret v_{et} as, "the color of edge e in the world of triangle t ". The concur projection will be applied to all the v_{et}'s with e fixed as t ranges over all the triangles that include e. The divide projection is applied to only three color variables at a time: the three v_{et}'s for each t. The job of this projection is just to ensure that no triangle is monochromatic.

If instead you use linear relations, you would have a color vector v_e on each edge and define, linearly, triangle variables $w_t = v_{t(1)} + v_{t(2)} + v_{t(3)}$, where $t(1)$, $t(2)$, $t(3)$ are the three edges of triangle t. All these linear equations are one of the RRR constraints. For the other RRR constraint you would constrain each v_e to be one of your three permutation-symmetric color codes, and some other constraint on the w_t's to keep triangles non-monochromatic.

The difficulty of finding colorings grows with n, of course, and reaching $n = 16$ might be a challenge (but it has been done).

Spin glass ground states

Use the rank-1 constrained symmetric matrix method demonstrated in lecture to find low energy states for a 40-spin Ising model. The energy of the model is defined by

$$E = \frac{1}{2} \sum_{i,j} s_i H_{ij} s_j,$$

and you can find the 40×40 matrix H of couplings, `spinglassH`, in the data directory of the github site. Give the lowest energy you find along with the list of spins with value $+1$, using the convention $s_1 = +1$.

Survey questions

1. Which programming language do you prefer for in-class demonstrations?
2. Which programming language are you most motivated to learn?
3. List applications that you would have liked to have seen demonstrated.
4. What would be a good name for the proposed python package?