Cleanup notes on max-cut.

Max-cut:

Given $G = (V, E)$, $w : E \rightarrow \mathbb{Z}^+$,

Determine $S, \overline{S}$ s.t. $S \cup \overline{S} = V$

$S \cap \overline{S} = \emptyset$

$\omega(S, \overline{S})$ is maximised

Randomized strategy:

randomly assign $-1, 1$ to verts

$S = \{ \text{verts w/} 1 \}$

$\overline{S} = \{ \ldots \ldots -13 \}$

In this case,

$P[ u \cdot v \text{ crosses cut}]$

$= P[u = 1, v = -1] + P[u = -1, v = 1]$

$= \frac{1}{2}$.

Now, the value of the cut is

$Z_{y_r} = \frac{1}{2} \sum_{i \neq j} w_{ij} [1 - y_i y_j]$
\[ E[Z_{yr}] = \frac{1}{2} \sum_{i \neq j} w_{ij} (1 - E[y_i y_j]) \]

\[ E[y_i y_j] = P[y_i y_j = 1] \cdot 1 + P[y_i y_j = -1] \cdot 1 \]

\[ = 0 \]

\[ \therefore E[Z_{yr}] = \frac{1}{2} \sum_{i \neq j} w_{ij} \]

Now, the value of \( Z \text{opt} \) is value of best cut. Write \( Z \text{opt} \).

\[ Z \text{opt} \leq \frac{1}{2} \sum_{i \neq j} w_{ij} \]

(because weights are non-negative)

\[ \therefore E[Z_{yr}] = \frac{1}{2} \sum_{i \neq j} w_{ij} \geq \frac{1}{2} Z \text{opt} \]
GKW bound:

we solve

$$\max \frac{1}{2} \sum_{i \neq j} w_{ij} (1 - v_i \cdot v_j)$$

s.t. $v_i \cdot v_j = 1$

(Which is an SDP). Value of this is $Z_p^*$

we now choose a random vector $r$

at vert $i$, if $v_i \cdot r > 0$, $i \rightarrow S$

$$P_i [i, j \in \text{cut}] = \frac{\cos^{-1}(v_i \cdot v_j)}{\pi}$$ (elementary geometry)

write $W$ for the value of the cut.

$$E[W] = \frac{1}{\pi} \sum_{i \neq j} w_{ij} \cos^{-1}(v_i \cdot v_j)$$
Now, for $\theta = 0.878 \cdots$

\[ \frac{\theta}{\pi} > \frac{1}{2} \frac{1}{1 - \cos \theta} \]

which we can get by minimizing

\[ \frac{\theta}{\pi} \cdot \frac{2}{1 - \cos \theta} \]

but write $\theta_{ij} = \cos^{-1}(v_i \cdot v_j)$

we have

\[ E[w] = \frac{1}{\pi} \sum_{i>j} w_{ij} \cos^{-1}(v_i \cdot v_j) \]

\[ = \frac{\theta}{\pi} \sum_{i>j} w_{ij} \frac{\theta_{ij}}{\pi} \]

\[ \geq \alpha \cdot \sum_{i>j} w_{ij} (1 - \cos \theta_{ij}) = \alpha Z_p^* \]

\[ \langle Z_p^* \rangle \geq \alpha Z_{opt} \]

so $E[w] \geq \alpha Z_{opt}$
Doubly stochastic matrices and balancing

Recall $M$ is doubly stochastic if

1. $M_{ij} \geq 0$
2. $\sum_i M_{ij} = \sum_j M_{ij} = 1$

(Notice this means that $M$ is square.)

Recall

Convex hull of permutation matrices = doubly stochastic matrices (easy).

Vertices of doubly stochastic matrix polytope = permutation matrices (deep, non-obvious; Birkhoff).

Above establishes how d.s. matrices could be interesting also, matching paper.
Procedure to produce d.s. matrix from $M$.

Iterate

\[
M_{ij}^* = \frac{M_{ij}^{(n)}}{\sum_j M_{ij}^{(n)}}
\]

\[
M_{ij} = \frac{M_{ij}^*}{\sum_i M_{ij}^*}
\]

and keep going, till convergence.

Facts:

- If procedure converges, it converges to unique d.s. matrix.
- Zeros may cause a failure to converge (conditions under which it converges are known, see papers)
- If doesn't help to make zeros into 3 (e.g. next page)
Example:

\[
\begin{pmatrix}
0 & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 1
\end{pmatrix}
\]
approx as

\[
\begin{pmatrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{pmatrix}
\]

converges to

\[
\begin{pmatrix}
0.25 & 0.25 & 0.5 \\
0.25 & 0.25 & 0.5 \\
0.5 & 0.5 & 0
\end{pmatrix}
\]

\( \varepsilon \to 0 \)

but

\[
\begin{pmatrix}
0 & 0 & 1 \\
0 & 0 & 1 \\
1 & 1 & 1
\end{pmatrix}
\]

\( \varepsilon \to 0 \)

\[
\begin{pmatrix}
0.5 & 0.5 \\
0.5 & 0.5 \\
0 & 0 & 1
\end{pmatrix}
\]

\( \varepsilon \to 0 \)

This procedure has some quite surprising applications.
Sinkhorn can solve Sudoku!

Sudoku: \( N \times N \) grid of cells

- \( N \) blocks of \( N \) elements
- into each cell, insert \( 1 \) to \( N \)
- exactly \( 1 \) in each row, column, block

- \( N \) is usually 9.

- There is initial data (and the solver must fill in the rest).

- There are \( 3N \) vector constraints

  have a doubly stochastic flavor

  e.g., write \( s_{ij}^\ell = 1 \) if \( \ell \) in the cell has \( \ell \)

we have
\[ \sum_i S_{ij}^e = 1 \]
\[ \sum_j S_{ij}^e = 1 \]
\[ \sum_{i,j \in \text{block}} S_{ij}^e = 1 \] (for each block, only one # per table location).

Procedure:
- Write \( p = [p(e_1 = 1), p(oc = 2), \ldots, p(oc = N)] \) (which we interpret as probabilities, but...)
- Now imagine a col of values
  \[ \begin{bmatrix} \mathbf{p}^m \end{bmatrix} \]
  \[ \downarrow \]
  \[ \text{cells} \]
  \[ \begin{bmatrix} \mathbf{p}^{m_1} & \mathbf{p}^{m_2} \end{bmatrix} \]
at the right answer, all $Q^\pi$ would be permutations

for prescribed data, we can adjust the $p$-vector (i.e. it must be $(0 \cdots 0)$

This means that $Q^\pi$ at least are doubly stochastic.

Idea:

- construct each row $Q^\pi$
  - straighten
  - "col
    - straighten
  - "block
    - straighten
  - stop on convergence.
Each balancing step does not move farther from some appropriate distance.

Notation:

\[ S_{ij} \uparrow \downarrow \text{ for a Solution} \]

\[ S_{ij} = \begin{cases} \uparrow & \text{if cell } i,j = \downarrow \\ 0 & \text{otherwise} \end{cases} \]

Notice \( S_{ij} \) has balance properties.

Write \( \hat{S}_{ij} \) for \( e \) the \( n \)th rest of \( \hat{S}_{ij} \).

Start with \( \hat{g}_{ij} \) and \( \sum_{e} \hat{g}_{ij} = 1 \).
Each step of each stage does not increase

\[ D(S \| g) = \sum_{i, j} S_i^* \log \frac{S_i^*}{g_{ij}} \]

**Proof:** (case-by-case)

- **col balancing for a row**

- **eg:**

  - matrix

  - matrix is \( Q \)

  \[ q_{ij} = \frac{S_i^*}{g_{ij}} \]

  - for \( k \)'th row, \( q_{je} = \frac{S_{kj}}{g_{jej}} \) fixed

and **col balancing** takes

\[ q_{je} \rightarrow \frac{q_{je}}{\sum_j q_{je}} = \frac{q_{je}}{\sum_j q_{je}} \]

\[ \rightarrow X_e \]
i.e., this gets

\[(n+1) \ell \quad g_{ij} = g_{ij} \quad \text{for } i \neq k \]

\[(n+1) \ell \quad g_{kj} = \frac{\ell g_{kj}}{\chi_e} \quad i \neq k \]

Now

\[D(\hat{S}^{(n+1)} / g) < D(\hat{S}^{(n)} / g) \]

\[= \sum_{ij} \hat{S}_{ij} \log \frac{\hat{S}_{ij}}{g_{ij}} + \sum_{je} \hat{S}_{kj} \log \chi_e \]

the change is

\[\sum_{je} \hat{S}_{kj} \log \chi_e \leq \sum_{je} \hat{S}_{kj} (\chi_e - 1) \]

because \(\log x < x - 1\)

\[= \sum_{je} \hat{S}_{kj} \chi_e - N \]

\[= N - N \]

\[= 0 \]
Other cases follow same line.

In practice, works OK.