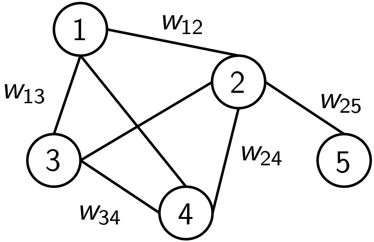
Relax and do Something Random: The MAXCUT Approximation of Goemans and Williamson

Michael McCoy

January 29, 2010

What is a cut?

- Given a graph (V, E) with edge weights $w_{ij} \geq 0$,



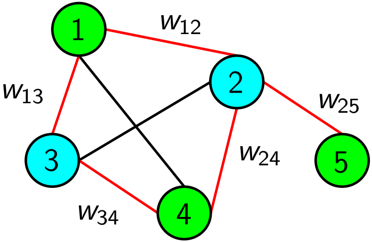
a cut S is a subset of the vertices $S \subset V$.

- The weight of the cut $\omega(S)$ is the sum of the weights of the edges that "cross the cut":

$$\omega(S) = \sum_{i \in S,\, j \notin S} w_{ij}$$

Cut Example

- Here, $S = \{1, 4, 5\}$.



- The weight of the cut $\omega(\{1, 4, 5\})$ is

$$\omega(\{1, 4, 5\}) = w_{12} + w_{13} + w_{24} + w_{25} + w_{34}. \qquad (1)$$

MAXCUT

- Determining a subset $S \subset V$ that maximizes $\omega(S)$ is the MAXCUT problem:

$$\begin{array}{rl} \text{maximize} & \omega(S) \\ \text{subject to} & S \subset V \end{array} \qquad \text{(MAXCUT)}$$

- Equiavelently, we can write MAXCUT as

$$\begin{array}{rl} \text{maximize} & \frac{1}{4} \sum_{i,j} w_{ij}(1 - \sigma_i \sigma_j) \\ \text{subject to} & \sigma_i = \pm 1 \text{ for all } i \in V \end{array} \qquad \text{(MAXCUT')}$$

- Equivalent by setting $i \in S \iff \sigma_i = +1$.

MAXCUT

- Determining a subset $S \subset V$ that maximizes $\omega(S)$ is the MAXCUT problem:

$$\begin{array}{rl} \text{maximize} & \omega(S) \\ \text{subject to} & S \subset V \end{array} \qquad \text{(MAXCUT)}$$

- Equiavelently, we can write MAXCUT as

$$\begin{array}{rl} \text{maximize} & \frac{1}{4}\sum_{i,j} w_{ij}(1-\sigma_i\sigma_j) \\ \text{subject to} & \sigma_i = \pm 1 \text{ for all } i \in V \end{array} \qquad \text{(MAXCUT')}$$

- Equivalent by setting $i \in S \iff \sigma_i = +1$.
- MAXCUT is known to be *NP*-complete.

Relax

- Key Idea: Replace integers $|\sigma_i| = 1$ with norm-1 vectors $\|u_i\| = 1$, and scalar multiplication with vector multiplication.

$$\begin{array}{rl} \text{maximize} & \frac{1}{4}\sum_{i,j} w_{ij}(1 - \langle u_i, u_j \rangle) \\ \text{subject to} & \|u_i\| = 1 \text{ for all } i \text{ in } V \end{array} \qquad \text{(RELAX)}$$

- This is a *relaxation* of MAXCUT since the original problem is contained in this problem, e.g., take $u_i = (\pm 1, 0, \ldots, 0)$.
- We will show later how to compute the u_i's using a semidefinite program.

A key result

Lemma

Let r be a random[1] *vector. For any unit vectors u_i and u_j,*

$$\mathbb{P}\big(\text{sign}(\langle u_i, r\rangle) \neq \text{sign}(\langle u_j, r\rangle)\big) = \frac{\arccos(\langle u_i, u_j\rangle)}{\pi}.$$

[1] By which me mean that r is drawn from a spherically symmetric distribution with zero mass at the origin.

A key result

Lemma

Let r be a random[1] vector. For any unit vectors u_i and u_j,

$$\mathbb{P}\big(\mathrm{sign}(\langle u_i, r\rangle) \neq \mathrm{sign}(\langle u_j, r\rangle)\big) = \frac{\arccos(\langle u_i, u_j\rangle)}{\pi}.$$

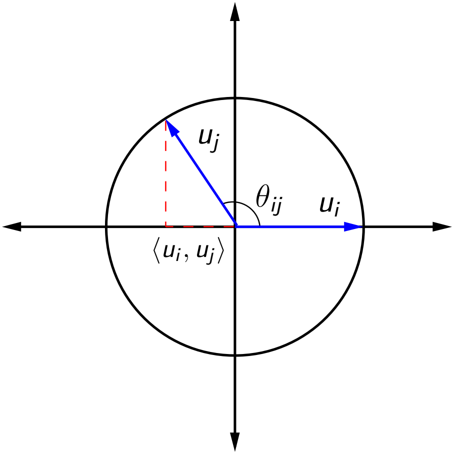
As an immediate consequence of this Lemma, we have that

$$\mathbb{E}\left[\frac{1}{2} - \frac{1}{2}\,\mathrm{sign}(\langle u_i, r\rangle)\,\mathrm{sign}(\langle u_j, r\rangle)\right] = \frac{1}{\pi}\arccos(\langle u_i, u_j\rangle).$$

[1] By which me mean that r is drawn from a spherically symmetric distribution with zero mass at the origin.

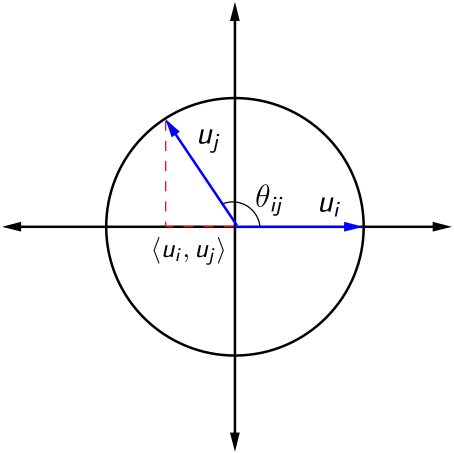
Proof.

Using a suitable rotation, we can assume without loss that $u_i = (1, 0, \ldots, 0)$ and $u_j = (a, b, 0, \ldots, 0)$. (Why?)



Proof.

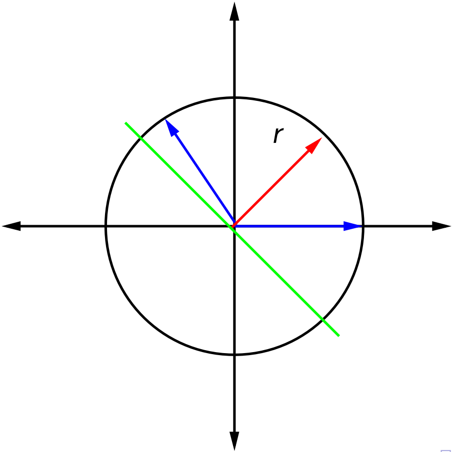
The signs of $\langle u_i, r\rangle$ and $\langle u_j, r\rangle$ are different if and only if the tangent plane of r separates u_i and u_j.



Proof.

The signs of $\langle u_i, r\rangle$ and $\langle u_j, r\rangle$ are different if and only if the tangent plane of r separates u_i and u_j.

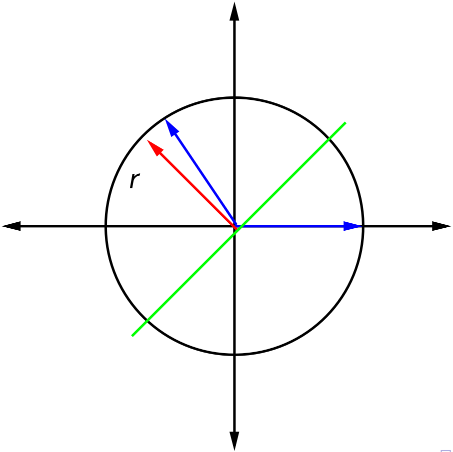
Here, the tangent plane does not separate u_i and u_j.



Proof.

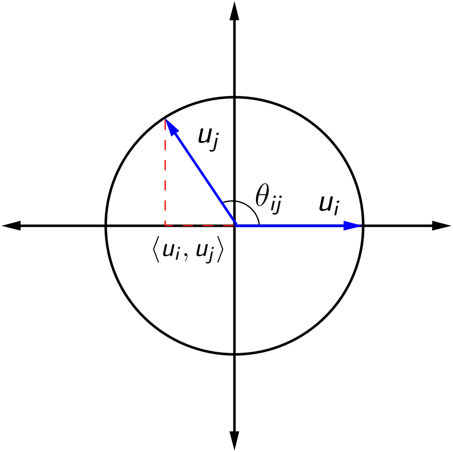
The signs of $\langle u_i, r \rangle$ and $\langle u_j, r \rangle$ are different if and only if the tangent plane of r separates u_i and u_j.

Here, the tangent plane separates u_i and u_j.



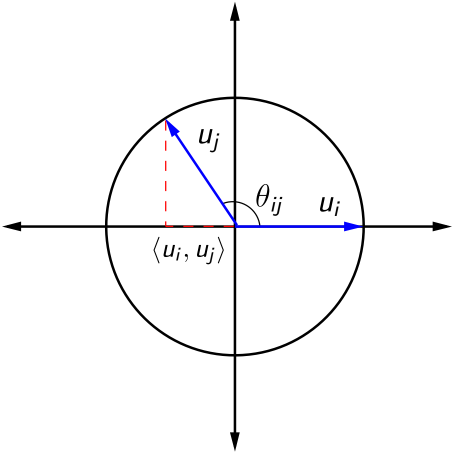
Proof.

A random line bisects an angle of θ_{ij} with probability $\frac{\theta_{ij}}{\pi}$,



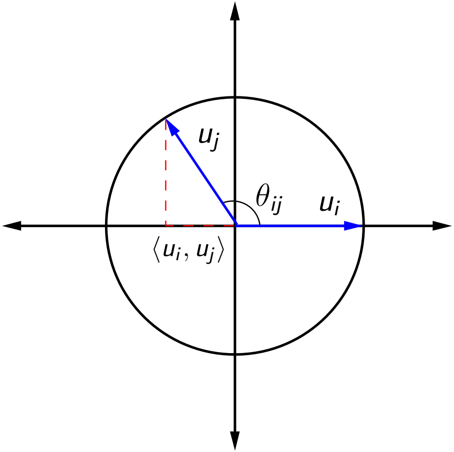
Proof.

A random line bisects an angle of θ_{ij} with probability $\frac{\theta_{ij}}{\pi}$,
but $\cos(\theta_{ij}) = \langle u_i, u_j \rangle$,



Proof.

A random line bisects an angle of θ_{ij} with probability $\frac{\theta_{ij}}{\pi}$,
but $\cos(\theta_{ij}) = \langle u_i, u_j \rangle$,
so that $\frac{\theta_{ij}}{\pi} = \frac{\arccos(\langle u_i, u_j \rangle)}{\pi}$. □



Converting back to scalars

Suppose we have the vectors u_i that solve RELAX. Then do the following:

1. Choose a random vector r.
2. Set $\hat{\sigma}_i = \text{sign}(\langle u_i, r\rangle)$ for all $i \in V$.
3. Equivalently, set $i \in \hat{S}$ if $\text{sign}(\langle u_i, r\rangle) \geq 0$.

Converting back to scalars

Theorem

Let S_* be a cut that optimizes MAXCUT. Then

$$\mathbb{E}[\omega(\hat{S})] \geq \alpha\, \omega(S_*),$$

where $\alpha > 0.87$.

Converting back to scalars

Theorem

Let S_* be a cut that optimizes MAXCUT. Then

$$\mathbb{E}[\omega(\hat{S})] \geq \alpha\, \omega(S_*),$$

where $\alpha > 0.87$.

For the proof, we will use the fact that

$$\frac{\arccos(y)}{\pi} \geq \alpha \frac{1-y}{2} \text{ for all } -1 \leq y \leq 1,$$

where

$$\alpha = \min_{0 \leq \theta \leq \pi} \frac{2\theta}{\pi(1-\cos(\theta))} > 0.87.$$

Proof.

By the corollary to the Lemma and our fact,

$$\mathbb{E}[\omega(\hat{S})] = \frac{1}{2}\sum_{i,j} w_{ij}\frac{\arccos\langle u_i, u_j\rangle}{\pi} \geq \frac{\alpha}{4}\sum_{i,j} w_{ij}(1-\langle u_i, u_j\rangle).$$

Since u_i and u_j maximize maximize the right-hand side over the unit sphere, by restriction, we have

$$\mathbb{E}[\omega(\hat{S})] \geq \frac{\alpha}{4}\sum_{i,j} w_{ij}(1-\sigma_i\sigma_j).$$

for any $\sigma_i = \pm 1$. In particular, the inequality holds for the maximum possible choice of signs, which by definition is $\alpha\,\omega(S_*)$. □

The step to semidefinite

- For a square matrix Σ, following are equivalent:
  1. $\Sigma \succcurlyeq 0$, $\Sigma_{ii} = 1$, and $\text{rank}(\Sigma) = 1$,
  2. $\Sigma = \sigma\sigma^t$, where $\sigma_i = \pm 1$.
- Setting $(W)_{ij} = w_{ij}$, note that

$$\sum_{i,j} w_{ij}\sigma_i\sigma_j = \text{tr}(W\sigma\sigma^t) \tag{2}$$

- Thus, MAXCUT is equivalent to

$$\begin{aligned} &\text{maximize} && \tfrac{1}{4}\textstyle\sum_{i,j} w_{ij} - \tfrac{1}{4}\text{tr}(W\Sigma) \\ &\text{subject to} && \Sigma \succcurlyeq 0,\ \Sigma_{ii} = 1,\ \text{and}\ \text{rank}(\Sigma) = 1. \end{aligned} \tag{3}$$

Semidefinite relaxation

- By dropping the rank-1 restriction, (3) becomes a semidefinite program:

$$\begin{array}{ll} \text{minimize} & \operatorname{tr}(W\Sigma) \\ \text{subject to} & \begin{cases} \Sigma \succcurlyeq 0, \\ \Sigma_{ii} = 1 \end{cases} \end{array} \tag{4}$$

- Setting $\Sigma = U^t U$ via a Cholesky factorization, the restriction $\Sigma_{ii} = 1$ implies that U has unit-norm columns.
- That is, (4) is equivalent to RELAX.

How to compute it

- For large problems, the current state-of-the-art algorithm for computing the solution to these semidefinite programs is available in Burer and Montiero [2].
- For moderately sized problems, use MATLAB's `CVX` package [3].

How to compute it

- For large problems, the current state-of-the-art algorithm for computing the solution to these semidefinite programs is available in Burer and Montiero [2].
- For moderately sized problems, use MATLAB's CVX package [3].

The entire code using CVX:

```
... % define W, N
cvx_begin sdp
    variable
        Sigma(N,N) symmetric
    minimize
        trace(W*Sigma)
    subject to
        Sigma >= 0;
        diag(Sigma) == ones(N,1);
cvx_end

U = chol(Sigma);

sigma = sign(U*randn(N,1));
```

How to compute it

- For large problems, the current state-of-the-art algorithm for computing the solution to these semidefinite programs is available in Burer and Montiero [2].
- For moderately sized problems, use MATLAB's CVX package [3].

The entire code using CVX:

```
... % define W, N
cvx_begin sdp
    variable
        Sigma(N,N) symmetric
    minimize
        trace(W*Sigma)
    subject to
        Sigma >= 0;
        diag(Sigma) == ones(N,1);
cvx_end

U = chol(Sigma); % May fail occasionally due to numerical inaccuracy

sigma = sign(U*randn(N,1));
```

For More Details

- M.X. Goemans and D.P. Williamson.
  Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.
  Journal of the ACM (JACM), 42(6):1145, 1995.

- S. Burer and R. Monteiro.
  Local Minima and Convergence in Low-Rank Semidefinite Programming.
  Mathematical Programming, 103(3):427–444, December 2004.

- M. Grant and S. Boyd.
  CVX: Matlab software for disciplined convex programming (web page and software)
  http://stanford.edu/~boyd/cvx