# Lecture 4: Goemans-Williamson Algorithm for MAX-CUT

# Lecture Outline

- Part I: Analyzing semidefinite programs
- Part II: Analyzing Goemans-Williamson
- Part III: Tight examples for Goemans-Williamson
- Part IV: Impressiveness of Goemans-Williamson and open problems

# Part I: Analyzing semidefinite programs

# Goemans-Williamson Program

- Recall Goemans-Williamson program: Maximize $\sum_{i,j:i<j,(i,j)\in E(G)} \frac{1-M_{ij}}{2}$ subject to $\mathrm{M} \succcurlyeq 0$ where $M \succcurlyeq 0$ and $\forall i, M_{ii} = 1$
- Theorem: Goemans-Williamson gives a .878 approximation for MAX-CUT
- How do we analyze Goemans-Williamson and other semidefinite programs?

# Vector Solutions

- Want: matrix $M$ such that $M_{ij} = x_i x_j$ where $\{x_i\}$ are the problem variables.
- Semidefinite program: Assigns a vector $v_i$ to each $x_i$, gives the matrix $M$ where $M_{ij} = v_i \cdot v_j$
- Note: This is a relaxation of the problem. To obtain an actual solution, we need a rounding algorithm to round this vector solution into an actual solution.

# Vector Solution Justification

- Theorem: $M \succcurlyeq 0$ if and only if there are vectors $\{v_i\}$ such that $M_{ij} = v_i \cdot v_j$
- Example: $M = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix}$, $\begin{matrix} v_1 = \langle 1,0,0 \rangle \\ v_2 = \langle -1,1,0 \rangle \\ v_3 = \langle 1,0,1 \rangle \end{matrix}$
- One way to see this: take a “square root” of $M$
- Second way to see this: Cholesky decomposition

# Square Root of a PSD Matrix

- If there are vectors $\{v_i\}$ such that $M_{ij} = v_i \cdot v_j$, take $V$ to be the matrix with rows $v_1, \cdots, v_n$. $M = VV^T \succcurlyeq 0$
- Conversely, if $M \succcurlyeq 0$ then $M = \sum_{i=1}^{n} \lambda_i u_i u_i^T$ where $\lambda_i \geq 0$ for all $i$. Taking $V$ to be the matrix with columns $\sqrt{\lambda_i} u_i$, $VV^T = M$. Taking $v_i$ to be the ith row of $V$, $M_{ij} = v_i \cdot v_j$

# Cholesky Decomposition

- Cholesky decomposition: $M = CC^T$ where $C$ is a lower triangular matrix.
- $v_i = \sum_a C_{ia} e_a$ is the ith row of $C$
- We can find the entries of $C$ one by one.

# Cholesky Decomposition Example

- Example: $M = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix}$
- $v_1 = \langle 1,0,0 \rangle$
- Need $C_{21} = -1$ so that $v_2 \cdot v_1 = -1$. $v_2 = \langle -1, C_{22}, 0 \rangle$
- Taking $C_{22} = 1$, $v_2 \cdot v_2 = 2$. $v_2 = \langle -1,1,0 \rangle$
- Need $C_{31} = 1$ and $C_{32} = 0$ so that $v_3 \cdot v_1 = 1$, $v_3 \cdot v_2 = -1$. $v_3 = \langle 1,0,C_{33} \rangle$.
- Taking $C_{33} = 1$, $v_3 \cdot v_3 = 1$. $v_3 = \langle 1,0,1 \rangle$

# Cholesky Decomposition Example

- $\begin{bmatrix} 1 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
- $v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, v_2 = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}, v_3 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$

# Cholesky Decomposition Formulas

- $\forall i < k$, take $C_{ki} = \frac{M_{ik} - \sum_{a=1}^{i-1} c_{ka} c_{ia}}{c_{ii}}$
- Take $C_{ki} = 0$ if $M_{ik} - \sum_{a=1}^{i-1} C_{ka} C_{ia} = C_{ii} = 0$
- Note that $v_k \cdot v_i = \sum_{a=1}^{i-1} C_{ka}\, C_{ia} + C_{ki} C_{ii} = M_{ik}$
- $\forall k$, take $C_{kk} = \sqrt{M_{kk} - \sum_{a=1}^{k-1} C_{ka}^2}$
- These formulas are the basis for the Cholesky-Banachiewicz algorithm and the Cholesky-Crout algorithm (these algorithms only differ in the order the entries are evaluated)

# Cholesky Decomposition Failure

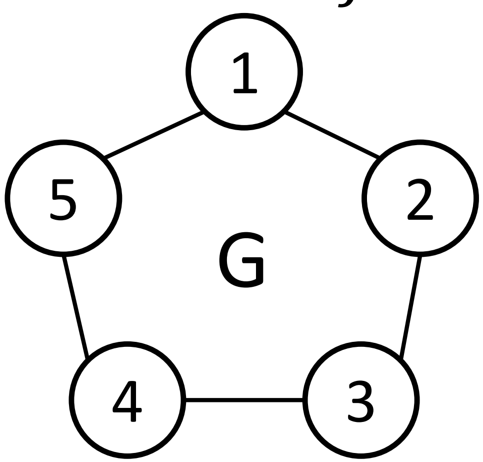
1. $\forall i < k, C_{ki} = \frac{M_{ik} - \sum_{a=1}^{i-1} C_{ka}C_{ia}}{C_{ii}}$
2. $\forall k,\ C_{kk} = \sqrt{M_{kk} - \sum_{a=1}^{k-1} C_{ka}^2}$

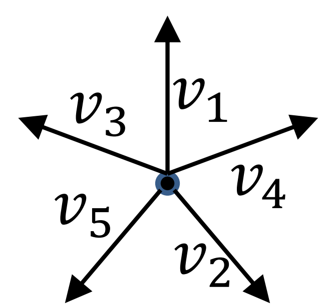
- If the Cholesky decomposition succeeds, it gives us vectors $\{v_i\}$ such that $M_{ij} = v_i \cdot v_j$
- The formulas can fail in two ways:
  1. $M_{kk} - \sum_{a=1}^{k-1} C_{ka}^2 < 0$ for some $k$
  2. $C_{ii} = 0$ and $M_{ik} - \sum_{a=1}^{i-1} C_{ka}C_{ia} \neq 0$ for some $i, k$
- Failure implies $M$ is not PSD (see problem set)

# Part II: Analyzing Goemans-Williamson

# Vectors for Goemans-Williamson

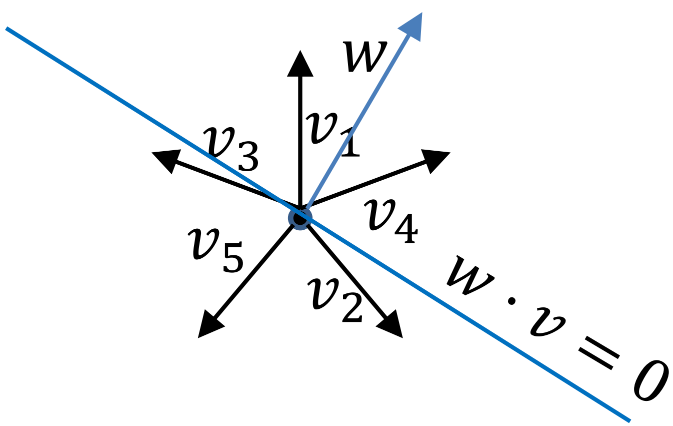
- Goemans-Williamson: Maximize $\sum_{i,j:i<j,(i,j)\in E(G)} \frac{1-M_{ij}}{2}$ subject to M $\succcurlyeq 0$ where $M \succcurlyeq 0$ and $\forall i, M_{ii} = 1$
- Semidefinite program gives us vectors $\{v_i\}$ where $v_i \cdot v_j = M_{ij}$

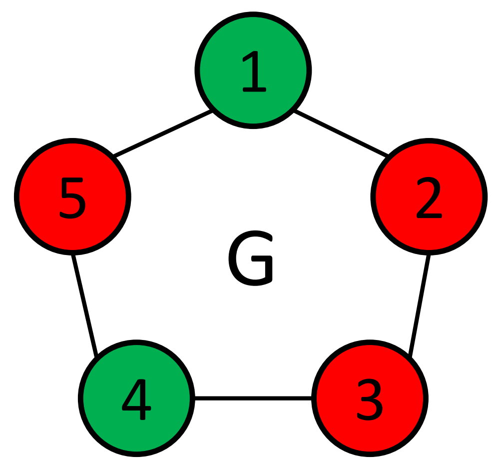


# Rounding Vectors

- Beautiful idea: Map each vector $v_i$ to $\pm 1$ by taking a random vector $w$ and setting $x_i = 1$ if $w \cdot v_i > 0$ and setting $x_i = -1$ if $w \cdot v_i < 0$
- Example:




$x_1 = x_4 = 1, x_2 = x_3 = x_5 = -1$

# Expected Cut Value

- Consider $E\left[\sum_{i,j:i<j,(i,j)\in E(G)} \frac{1-x_i x_j}{2}\right]$
- For each $i, j$ such that $i < j, i, j \in E(G)$, $E\left[\frac{1-x_i x_j}{2}\right] = \frac{\Theta}{\pi}$ where $\Theta \in [0, \pi]$ is the angle between $v_i$ and $v_j$
- On the other hand $\frac{1-M_{ij}}{2} = \frac{1-cos\Theta}{2}$

# Approximation Factor

- Goemens-Williamson gives a cut with expected value at least

$$\left(\min_{\Theta} \frac{\left(\frac{\Theta}{\pi}\right)}{\left(\frac{1-cos\Theta}{2}\right)}\right) \sum_{i,j:i<j,(i,j)\in E(G)} \frac{1-E_{ij}}{2}$$

- The first term is $\approx .878$ at $\Theta_{crit} \approx 134°$ $\sum_{i,j:i<j,(i,j)\in E(G)} \frac{1-E_{ij}}{2}$ is an upper bound on the max cut size, so we have a .878 approximation.

# Part III: Tight Examples

# Showing Tightness

- How can we show this analysis is tight?
- We give two examples where we obtain a cut of value $\approx .878\sum_{i,j:i<j,(i,j)\in E(G)} \frac{1-E_{ij}}{2}$
- In one example, $\sum_{i,j:i<j,(i,j)\in E(G)} \frac{1-E_{ij}}{2}$ is the value of the maximum cut. In the other example, $.878\sum_{i,j:i<j,(i,j)\in E(G)} \frac{1-E_{ij}}{2}$ is the value of the maximum cut.

# Example 1: Hypercube

- Have one vertex for each point $x_i \in \{\pm 1\}^n$
- We have an edge between $x_i$ and $x_j$ in $G$ if

$$\left|\cos^{-1}\left(\frac{x_i \cdot x_j}{n}\right) - \Theta_{crit}\right| < \delta$$

for an arbitrarily small $\delta > 0$

- Goemans-Williamson value $\approx \frac{1-\cos(\Theta_{crit})}{2} E(G)$
- This is achieved by the coordinate cuts.
- Goemans-Williamson rounds to a random cut which gives value $\approx \frac{\Theta_{crit}}{\pi} E(G)$

# Example 2: Sphere

- Take a large number of random points $\{x_i\}$ on the unit sphere
- We have an edge between $x_i$ and $x_j$ in $G$ if
$$\left|\cos^{-1}(x_i \cdot x_j) - \Theta_{crit}\right| < \delta$$
for an arbitrarily small $\delta > 0$
- Goemans-Williamson value $\approx \frac{1-\cos(\Theta_{crit})}{2} E(G)$
- A random hyperplane cut gives value $\approx \frac{\Theta_{crit}}{\pi} E(G)$ and this is essentially optimal.

# Proof requirements

- How can we prove the above examples behave as claimed?
- For the hypercube, have to upper bound the value of the Goemans-Williamson program.
- This can be done by determining the eigenvalues of the hypercube graph and using this to analyze the dual (see problem set)
- For the sphere, have to prove that no cut does better than a random hyperplane cut (this is hard, see Feige-Schechtman [FS02])

# Part IV: Impressiveness of Goemans-Williamson and Open Problems

# Failure of Linear Programming

- Trivial algorithm: Randomly guess which side of the cut each vertex is on.
- Gives approximation factor $\frac{1}{2}$
- Linear programming doesn't do any better, not even polynomial sized linear programming extensions [CLRS13]!

# Hardness of beating GW

- Only know NP-hardness for a $\frac{16}{17}$ approximation [Hås01], [TSSW00]
- Unique-Games hard to beat Goemans-Williamson on MAX-CUT [KKMO07]

# Open problems

- Can we find a subexponential time algorithm beating Goemans-Williamson on max cut?
- Can we prove constant degree SOS lower bounds for obtaining a better approximation than Goemans-Williamson?

# References


- [CLRS13] S. Chan, J. Lee, P. Raghavendra, and D. Steurer. Approximate constraint satisfaction requires large lp relaxations. FOCS 2013.
- [FS02] U. Feige and G. Schechtman. On the optimality of the random hyperplane rounding technique for max cut. Random Structures & Algorithms - Probabilistic methods in combinatorial optimization, 20 (3), p. 403 – 440. 2002
- [GW95] M. X. Goemans and D. P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. J. ACM, 42(6):1115-1145, 1995.
- [Hås01] J. Håstad. Some optimal inapproximability results. JACM 48: p.798-869, 2001.
- [KKMO07] S. Khot, G. Kindler, E. Mossell, R. O'Donnell. Optimal Inapproximability Results for MAX-CUT and Other 2-Variable CSPs? SIAM Journal on Computing, 37 (1): p. 319-357, 2007.
- [TSSW00] L. Trevisan, G. Sorkin, M. Sudan, and D. Williamson. Gadgets, approximation, and linear programming. SIAM Journal on Computing, 29(6): p. 2074-2097, 2000.