Problem Set 2 Solutions

Sum of Squares Seminar

October 9, 2017

Adjacency matrix

Several problems involve the adjacency matrix of a graph, so we recall the definition here. The adjacency matrix A of a graph G is defined to be the matrix whose entries are as follows:

1. $\forall i, A_{ii} = 0$
2. $\forall i < j, A_{ij} = A_{ji} = 1$ if $(i,j) \in E(G)$ and $A_{ij} = A_{ji} = 0$ if $(i,j) \notin E(G)$

Problem 1: Vectors for PSD Matrices (20 points)

Recall the following very useful characterization of PSD matrices. A matrix M is PSD if and only if there are vectors $\{v_i\}$ such that $\forall i,j, M_{ij} = v_i \cdot v_j$. Further recall the Cholesky-Banachiewicz or Cholesky-Crout algorithm for finding the Cholesky decomposition (which gives us such a set of vectors).

1. Let c_{ia} be the ath coordinate of v_i. Set $c_{11} = \sqrt{M_{11}}$ and set $c_{ia} = 0$ whenever $a > i$.
2. For all $i < k$, take $c_{ki} = \frac{M_{ik} - \sum_{a=1}^{i-1} c_{ka}c_{ia}}{c_{ii}}$ (take $c_{ki} = 0$ if $M_{ik} - \sum_{a=1}^{i-1} c_{ka}c_{ia} = c_{ii} = 0$)
3. For all k, take $c_{kk} = \sqrt{M_{kk} - \sum_{a=1}^{k-1} c_{ka}^2}$

(a) 5 points: Let $M = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 6 & 10 \\ 1 & 4 & 10 & 20 \end{bmatrix}$.

Find vectors v_1, v_2, v_3, v_4 such that $\forall i,j \in [1,4], M_{ij} = v_i \cdot v_j$. Can you see the pattern?

(b) 15 points: Show that if this algorithm fails on a matrix M then M is not PSD. (hint is available)

Solutions

Part a: Running the algorithm for the Cholesky decomposition gives us $v_1 = (1,0,0,0)$, $v_2 = (1,1,0,0)$, $v_3 = (1,2,1,0)$ and $v_4 = (1,3,3,1)$.

The pattern is that $M_{jk} = \binom{j+k-2}{j-1}$ and the ith entry of v_j is $\binom{j-1}{i-1}$, so the entries of these vectors look like Pascal's triangle. To show that this pattern always holds (which is nice but not required), we need to show that

$$v_j \cdot v_k = \sum_{i=1}^{j} \binom{j-1}{i-1}\binom{k-1}{i-1} = \sum_{i=0}^{j-1} \binom{j-1}{i}\binom{k-1}{i} = \binom{j+k-2}{j-1}$$

To see this, observe that to choose a set A of $j-1$ elements out of $j+k-2$ elements, it is sufficient to choose how many of the first $j-1$ elements are in A (which will also be the number of the last $k-1$ elements which are not in A) and then choose which elements these are. This is precisely the sum $\sum_{i=0}^{j-1}\binom{j-1}{i}\binom{k-1}{i}$.

Part b: We can show this by induction. Assume that the algorithm fails on an $n \times n$ matrix M but succeeds on all $(n-1) \times (n-1)$ matrices. There must be an index k such that we have $M_{kk} - \sum_{a=1}^{k-1} c_{ka}^2 \leq 0$, take k to be the first such index. Run the algorithm to obtain vectors $v_1, \cdots, v_k$ such that $\forall i,j \in [1,k], M_{ij} = v_i \cdot v_j$. If $M_{kk} - \sum_{a=1}^{k-1} c_{ka}^2 = 0$ then the last entry of v_k will be 0 and if $M_{kk} - \sum_{a=1}^{k-1} c_{ka}^2 < 0$ then the last entry of v_k will be imaginary.

If $M_{kk} - \sum_{a=1}^{k-1} c_{ka}^2 = 0$ then v_k will be a linear combination of $v_1, \cdots, v_{k-1}$ so we can write $v_k = \sum_{j=1}^{k-1} b_j v_j$. Letting r_j be the jth row of M, there are two cases to consider:

1. $r_k = \sum_{j=1}^{k-1} b_j r_j$
2. $r_k \neq \sum_{j=1}^{k-1} b_j r_j$

In the first case, the algorithm will take $v_k = \sum_{j=1}^{k-1} b_j v_j$, which succeeds as far as the entries in the kth row and column are concerned. In this case, the algorithm must fail if we run it on M with the kth row and column removed. By the inductive hypothesis, M is not PSD if we remove the kth row and column, which implies that M was not PSD.

In the second case, taking w to be the vector $(b_1, \cdots, b_{k-1}, -1, 0, \cdots, 0)$, $Mw \neq 0$ but the first k entries of Mw are 0, so $w^T M w = 0$ which immediately implies M is not PSD. To see this, choose a $j > k$ such that the value of Mw is nonzero in coordinate k. Now observe that for all B, $(Bw + e_j)^T M (Bw + e_j) = 2B(Mw)_j + M_{jj}$, so we can easily find a value of B which makes this negative.

If $M_{kk} - \sum_{a=1}^{k-1} c_{ka}^2 < 0$ then we can write $v_k = \sum_{j=1}^{k-1} b_j v_j + cie_k$ for some constant c. Taking w to be the vector $(b_1, \cdots, b_{k-1}, -1, 0, \cdots, 0)$,

$$w^T M w = \left(\sum_{j=1}^{k} w_j v_j\right) \cdot \left(\sum_{j=1}^{k} w_j v_j\right) = -c^2 < 0$$

Problem 2: Goemans-Williamson Dual (15 points)

(a) 10 points: What is the dual of the Geomans-Williamson semidefinite program?

(b) 5 points: Deduce that the value of the Goemans-Williamson semidefinite program is always at most $\frac{|E(G)|}{2} + \frac{n\lambda_{max}(-A)}{4}$ where $\lambda_{max}(-A)$ is the maximum eigenvalue of $-A$.

Solutions

Part a: The Goemans-Williamson program is as follows. Maximize $\sum_{i<j:(i,j)\in E(G)} \frac{1-M_{ij}}{2}$ subject to:

1. $\forall i, M_{ii} = 1$
2. $M \succeq 0$

We can either find the dual by putting this program in canonical form and and taking the canonical dual or by working with the program directly. Here we'll work with the program directly. Consider

$$\begin{aligned} f(M,y) &= \sum_{i<j:(i,j)\in E(G)} \frac{1-M_{ij}}{2} + (Id - M) \bullet (\sum_i y_i e_i e_i^T) \\ &= \frac{|E(G)|}{2} - M \bullet \left(\sum_i y_i e_i e_i^T + \frac{A}{4}\right) + \sum_i y_i \end{aligned}$$

where the M player is trying to maximize f, the y player is trying to minimize f, and we require that $M \succeq 0$. If the M player goes first, this is the Goemans-Williamson program. Note that we have $\frac{A}{4}$ rather than $\frac{A}{2}$ because this entrywise dot product counts each edge $(i,j) \in E(G)$ twice, once for the i,j entry and once for the j,i entry. If the y player goes first, this gives us the dual, which is as follows. Minimize $\frac{|E(G)|}{2} + \sum_i y_i$ subject to

1. $\sum_i y_i e_i e_i^T + \frac{A}{4} \succeq 0$

This program can be reexpressed as follows. Minimize $\frac{|E(G)|}{2} + tr(Q)$ where Q is a diagonal matrix such that $Q \succeq -\frac{A}{4}$

Part b: We can always take $Q = \lambda_{max}(-A)Id$ so the optimal value of the dual is always at most $\frac{|E(G)|}{2} + \frac{n\lambda_{max}(-A)}{4}$. Thus, the value of the Goemans-Williamson program is always at most $\frac{|E(G)|}{2} + \frac{n\lambda_{max}(-A)}{4}$, as needed.

Problem 3: Goemans-Williamson on the Cycle (15 points)

Let G be the cycle graph on n vertices, i.e. $E(G) = \{(i, i+1) : i \in [1, n]\} \cup \{(1, n)\}$

(a) 10 points: What are the eigenvalues and eigenvectors of A? (hint is available)

(b) 5 points: What will the Goemans-Williamson program output when gven G? In particular, what vector v_i will be associated to each vertex i and what will the final value be?

Solutions

Part a: For one set of eigenvectors of A, we can take the vectors $\{v_j : j \in [0, n-1]\}$ where $(v_j)_k = e^{\frac{2\pi ijk}{n}}$. Now observe that for each v_j, $(v_j)_{k-1} = e^{-\frac{2\pi ij}{n}}(v_j)_k$ and $(v_j)_{k+1} = e^{\frac{2\pi ij}{n}}(v_j)_k$. Thus, $A(v_j) = \left(e^{-\frac{2\pi ij}{n}} + e^{\frac{2\pi ij}{n}}\right)(v_j) = 2cos\left(\frac{2\pi j}{n}\right)(v_j)$ so v_j has eigenvalue $2cos\left(\frac{2\pi j}{n}\right)$.

To obtain a set of real eigenvectors of A, observe that for all $j \in [1, \frac{n-1}{2}]$, v_j and v_{n-j} both have eigenvalue $2cos\left(\frac{2\pi j}{n}\right)(v_j)$ Thus, we can replace these eigenvectors with the eigenvectors $u_j = \frac{v_j + v_{n-j}}{2}$ and $w_j = \frac{v_j - v_{n-j}}{2i}$ which have coordinates $(u_j)_k = cos\left(\frac{2\pi jk}{n}\right)$ and $(w_j)_k = sin\left(\frac{2\pi jk}{n}\right)$.

Part b: The Goemans-Williamson program can map each vertex j to $\left(cos\left(\frac{2\pi j\lfloor\frac{n}{2}\rfloor}{n}\right), sin\left(\frac{2\pi j\lfloor\frac{n}{2}\rfloor}{n}\right)\right)$. The resulting value of the Goemans-Williamson SDP is

$$\sum_{i<j:(i,j)\in E(G)} \frac{1 - M_{ij}}{2} = \frac{n\left(1 - cos\left(\frac{2\pi\lfloor\frac{n}{2}\rfloor}{n}\right)\right)}{2}$$

This equals n if n is even and $\frac{n\left(1+cos\left(\frac{\pi}{n}\right)\right)}{2}$ if n is odd.

Problem 4: Eigenvalues of the Hypercube Graph (15 points)

Let G be the graph with one vertex for each point of the hypercube $\{-1, 1\}^n$ and edges $E(G) = \{(x, y) : x, y \text{ differ in } k \text{ coordinates}\}$ where k is even.

(a) 10 points: What are the eigenvectors and eigenvalues of the adjacency matrix A? (hint is available)

Note: the expression for the eigenvalues will be somewhat messy. Challenge question: It is intuitively clear that when k is significantly greater than $\frac{n}{2}$, $\lambda_{max}(-A)$ is given by the coordinate cuts. Can you prove it?

(b) 5 points: Assuming that $\lambda_{max}(-A)$ is given by the coordinate cuts, deduce that Goemans-Williamson gives the correct value on G. Letting $\Theta = cos^{-1}(\frac{n-2k}{n})$, what happens when we round the feasible solution (to the SDP) where each vertex x is mapped to $\frac{x}{\sqrt{n}}$ (viewing x as a vector)?

Solutions

Part a: One basis of eigenvectors is given as follows. For each $A \subseteq [1, n]$, take v_A to be the vector with coordinates $(v_A)_x = (-1)^{1_A \cdot \frac{1-x}{2}}$ where 1_A is the vector with coordinates $(1_A)_i = 1$ if $i \in A$ and 0 otherwise and 1 is the all ones vector. The eigenvalue of v_A is

$$|\{B \subseteq [1, n], |B| = k, |B \cap A| \text{ is even}\}| - |\{B \subseteq [1, n], |B| = k, |B \cap A| \text{ is odd}\}|$$

Letting $a = |A|$, v_A has eigenvalue

$$\sum_{j=0}^{a} (-1)^j \binom{a}{j}\binom{n-a}{k-j}$$

When $|A| = 1$ or $|A| = n - 1$, v_A has eigenvalue

$$\binom{n-1}{k} - \binom{n-1}{k-1} = \left(\frac{n-k}{n} - \frac{k}{n}\right)\binom{n}{k} = \frac{n-2k}{n}\binom{n}{k}$$

Part b: In fact, the Goemans-Williamson program always gives the correct answer on this kind of hypercube graph. From problem 2, part b, the value of the primal program is at most

$$\frac{|E(G)|}{2} + \frac{2^n \lambda_{max}(-A)}{4}$$

For any actual solution on a graph G, we will have that $M = vv^T$ where $v \in \{-1, 1\}^{|V(G)|}$ and thus $||v||^2 = |V(G)|$. In this case, the size of the cut is

$$\sum_{i<j:(i,j)\in E(G)} \frac{1 - M_{ij}}{2} = \frac{|E(G)|}{2} - \frac{A \bullet M}{4} = \frac{|E(G)|}{2} - \frac{|V(G)|}{4}\frac{v^T A v}{v^T v}$$

For this kind of hypercube graph, every eigenvalue of $-A$ has an eigenvector $v \in \{-1, 1\}^{2^n}$. Taking the vector v corresponding to the maximal eigenvector $\lambda_{max}(-A)$, if $M = vv^T$ then

$$\sum_{i<j:(i,j)\in E(G)} \frac{1 - M_{ij}}{2} = \frac{|E(G)|}{2} + \frac{2^n \lambda_{max}(-A)}{4}$$

Thus, there is indeed a cut of this size, so the maximal cut size and the value of the Goemans-Williamson SDP are both $\frac{|E(G)|}{2} + \frac{2^n \lambda_{max}(-A)}{4}$. However, the solution to the SDP may be an average of many solutions in which case rounding it can give a cut with a worse value.

Problem 5: Optimization Versus Feasibility Testing (10 points)

Let's say that we want to find the minimum value of some function h subject to a set of polynomial constraints. We have seen two alternative ways to use SOS to lower bound this. The two ways are as follows:

1. The first way (which was discussed briefly in Lecture 1) is to add $h \leq c$ (equivalently, $h = c - z^2$ for a new variable z) as a problem constraint and use SOS as a feasibility test to determine if this is feasible. We can then use binary search to find the minimal value of c where SOS thinks the equations are feasible and output this value.
2. The second way (which we saw in Lecture 3) is find the minimal possible value of $\tilde{E}[h]$ over any pseudo-expectation values $\tilde{E}$ which respect the problem constraints and output this value.

Does one of these alternatives give a better bound than the other? If so, why? If not, why not?

Solution

The first method is stronger for the following reason. If we have $h = c - z^2$ as a problem constraint, then we can multiply this constraint by any polynomial f whose degree is not too high to obtain the constraint $\tilde{E}[fh] = \tilde{E}[cf - z^2 f]$. However, with the second method we only have the constraint that $\tilde{E}[h] \leq c$. For an example where this makes a difference, see problem set 3.

Problem 6: Applying Goemans-Williamson (25 points)

Apply the Goemans-Williamson semidefinite program and rounding algorithm to the following graphs (the adjacency matrices for the graphs are provided as .txt files on the course website):

(a) A random $G(n, \frac{1}{2})$ graph on $n = 30$ vertices.

(b) Half of the hypercube graph described in problem 4 with $n = 6$ and $k = 4$.

(c) A graph formed by taking two communities of size 15, adding each edge within a community with probability .3, and adding each edge between communities with probability .7.

(d) The Peterson graph.

For your answers, please give the value of the Goemans-Williamson progam and a cut obtained by the rounding algorithm. Optionally, you may also give the matrices outputted by the program and the radom vector used for the rounding, as this allows each step of the algorithm to be checked.

Solution

The values given by the Goemans-Williamson SDP on these graphs are as follows:

1. 161.051
2. 160
3. 134.633
4. 12.5

Hints

1b. To show that M is not PSD, find a vector w such that $w^T M w < 0$. If the algorithm fails because $M_{kk} < \sum_{a=1}^{k-1} c_{ka}^2$, we can instead take $c_{kk} = i\sqrt{\sum_{a=1}^{k-1} c_{ka}^2 - M_{kk}}$. Now write ie_k as a linear combination of the vectors $v_1, \cdots, v_k$. If the algorithm fails because $c_{ii} = 0$ and $M_{ik} \neq \sum_{a=1}^{i-1} c_{ka} c_{ia}$, show that there is a vector w on the first i coordinates such that

1. w is an eigenvector of the submatrix of M consisting of the first i rows and columns which has eigenvalue 0.
2. The inner product of w with the kth row/column of M is nonzero.

Now use this to find a vector w' such that ${w'}^T M w' < 0$

3a. This is one of the few times in this seminar where it is very useful to use complex numbers! Show that $\forall k \in [0, n-1]$, the vector v_k with jth coordinate $v_{kj} = e^{\frac{2\pi ijk}{n}}$ is an eigenvector of the adjacency matrix. To show this and find its eigenvalue, think geometrically! In other words, view these complex numbers as vectors in the complex plane.

4a. Use discrete Fourier analysis over the hypercube! In particular, by symmetry, all vectors of the following form are eigenvectors: Let $A \subseteq [1, n]$ and set $v_x = (-1)^{|A \cap L_x|}$ where $L_x = \{i \in [1, n] : x_i = -1\}$.