We often think of graphs geometrically, i.e. we see vertices as points on a plane or in space, with edges as lines connecting them. When specified algebraically (e.g. by its adjacency matrix), the geometrical imagery is absent. Can we somehow create it from the algebraic description? The geometric representation might be useful in itself, say because we want to draw the graph on paper. But once a graph is embedded in space, in a *nice* manner, say whereby neighbouring vertices are placed close, perhaps problems such as partitioning a graph might be solved by partitioning the associated volume.

We can start with any algebraic representation of the graph, but a convenient one is the *node-edge incidence matrix* B of a graph G with n vertices and m edges. B has n rows and m columns, and $b_{ie} = 1$ if vertex i is an endpoint of edge e, and 0 otherwise. Clearly, every column of B has exactly two 1s and rest 0s. Note that[1] $BB^T = A + D$ where A is the adjacency matrix, in which $a_{ij} = 1$ if (i, j) is an edge and 0 otherwise, and D a diagonal matrix in which the iith entry equals the degree d_i of vertex i.

For d-regular graphs, the matrix B readily yields a nice geometric representation: we consider the elements of the ith row B_i of the incidence matrix B as the coordinates of vertex i in m dimensional space. If (i, j) is an edge, then the distance between i, j is $\sqrt{2d-1}$, whereas otherwise it is $\sqrt{2d}$. That neighbours are located nearer than non neighbours seems like a good property.

Suppose now that we want to partition this graph into equal sized subgraphs (as possible) by removing minimum number of edges. We instead look to partition the *point cloud* formed by the vertices. It turns out (exercises) that simply by slicing the cloud by a hyperplane, we can generate essentially any partition. The key question then is how do we find the hyperplane which gives us the partition we want. If our point cloud is in some sense homogeneous, it may seem natural to find the direction in which the point cloud is long, and cut perpendicular to that direction, expecting we will cut a small cross section of the cloud and hence presumably a small number of edges.[2] Likewise, suppose we want to draw a picture of the graph,

[1] Let B_i denote the ith row of B. Then $(BB^T)_{ij} = B_i B_j^T$. This evaluates to the degree d_i of vertex i if $i = j$, to 1 if (i, j) is an edge, and 0 otherwise.

[2] It is not clear, a priori, that it is even possible to get the best cut using just a

we could project the cloud onto two dimensions. Presumably the dimensions on which to project might correspond to the directions in which the cloud is longest, thereby reducing the amount of superposition of the vertices, i.e. the likelihood of unrelated vertices projecting on top of each other.

The key questions in all this are to find the directions in which the point cloud is long – this is precisely the question answered in Singular value decomposition (SVD), which we study next. SVD of the node-edge incidence matrix is intimately related to eigenvalue decomposition of the adjacency matrix, and also another matrix called the Laplacian matrix which we also study.

For non-regular graphs the SVD as above doesnt seem to be of much value. However, the situation can be fixed as follows: we define a normalized incidence matrix C where $c_{ie} = b_{ie}/\sqrt{d_i}$. Alternately $C = D^{-1/2}B$. The matrix C can be seen to have the property that non-adjacent vertices are located farther than adjacent vertices. Note however, that lengths of different edges is different, whereas it was the same for the regular case. Alternatively, we can directly study the eigenvalue decompositions of the adjacency matrix and the Laplacian matrix. These decompositions are still useful, but they do not have as intuitive a rationale as the node-edge incidence embedding.

The eigenvalues of a matrix (or its singular values) are said to constitute its spectrum. Hence the name Spectral Graph Theory.

1 Singular value decomposition

Note that the discussion in this section applies to any matrix B, not just the incidence matrix considered above.

Given an $n \times m$ matrix B, its first (right) singular vector, customarily denoted as v_1, is defined as a unit vector that maximizes $||Bv_1||_2$. In order for the norm to be maximized, it is necessary, in some sense, that v_1 approximate the "dominant" direction among the rows of B. If our point cloud were ellipsoidal and centered at the origin, v_1 would indeed be vaguely parallel to the longest axis. If the point cloud is far away from the origin, then v_1 generally points towards the center of the cloud, and the second singular vector would then (roughly) identify the longest axis, as we will see.

The first left singular vector could be defined as the right singular vector

hyperplane. See exercise 11.

of the matrix B^T. However, there is an elegant joint definition of both left and right singular vectors:

$$u_1, v_1 = \text{unit vectors that maximize } u_1^T B v_1$$

Note that this immediately shows (exercise) that the direction of Bv_1 must be the same as that of u_1, and that of $u_1^T B$ the same as that v_1^T. Further, we define $\sigma_1 = ||Bv_1||_2 = ||u_1^T B||_2$ as the first singular value.

Next we formalize the intuition that v_1 is a "rough" estimate of the directions of the rows of B. In particular, noting that Bv_1 gives the projections of the rows of B on v_1, the rank 1 matrix $B_1 = Bv_1 v_1^T = \sigma_1 u_1 v_1^T$ can be considered to be a rank 1 approximation to B. In fact we can prove that B_1 is the best such matrix in the sense of the Frobenius norm:

$$||B - B_1||_F = \min_R ||B - R||_F$$

where R is any rank 1 matrix. Note that the Frobenius norm $||B||_F$ of a matrix is simply the sum of the squares of all its entries.[3]

Writing $B' = B - B_1$, we simply repeat the above procedure on B' to get u_2, v_2, σ_2, B_2, where $B'v_2 = \sigma_2 u_2$, $u^T B' = \sigma_2 v_2^T$ and $B_2 = \sigma_2 u_2 v^T$. Thus we have:[4]

1. $B'v_1 = Bv_1 - Bv_1 v_1^T v_1 = 0$, and hence the rank of B' will be 1 less than that of B, and hence the process will terminate after r steps, where r is the rank of B.

2. We know that rows of B' are orthogonal to v_1. But $\sigma_2 v_2 = u_2^T B'$ is a linear combination of the rows of B', hence v_1, v_2 are orthogonal. Similarly u_1, u_2 are orthogonal.

[3]For a proof, suppose that we write the rank 1 matrix R as $R = xy^T$ where y is a unit vector. Then we want $\min_{x,y} ||B - xy^T||_F^2 = \min_{x,y} \sum_i ||B(i) - x_i y^T||_2^2$, where we have used $B(i)$ to mean the ith row of B, and x_i is the ith component of x. But now for fixed y, the sum can be minimized term wise. But $||B(i) - x_i y^T||_2^2$ will be minimized if $B(i) - x_i y^T$ is the component of $B(i)$ perpendicular to y^T. This happens when we choose $x_i = B(i)y$, or $x = By$. Applying the Pythagorean theorem to the components of $B(i)$ perpendicular and parallel to y we get the perpendicular component squared to be $||B(i)||_2^2 - ||B(i)yy^T||_2^2$. So $\min_{x,y} ||B - xy^T||_F^2 = \min_y \sum_i ||B(i)||_2^2 - ||B(i)yy^T||_2^2 = \min_y ||B||_F^2 - ||By||_2^2 = ||B||_F^2 - \max_y ||By||_2^2$. But this maximum is attained when $y = v_1$, which gives $R = Byy^T = Bv_1 v_1^T = \sigma_1 u_1 v_1^T$ as desired.

[4]In class we defined $B_k = \sum_{i=1}^k \sigma_2 u_i v_i^T$.

3. We have $\sigma_2 u_2 = Bv_2$ because $Bv_2 = B'v_2 + B_1 v_2 = B'v_2 + \sigma_1 u_1 v_1^T v_2 = B'v_2 = \sigma_2 u_2$.

Let u_i, v_i, σ_i, B_i, for $i = 1$ to r be obtained in this manner. These respectively give the ith left singular vector, the ith right singular vector, and the ith singular value. Obviously $B = \sum_{k=1}^{r} B_k$. Reasoning as above we have that all v_i are mutually orthogonal, all u_i are mutually orthogonal, and $\sigma_i u_i^T = Bv_i$.

It is customary to define U_k as consisting of the matrix made up by using $u_1, \ldots, u_k$ as its columns. Likewise V_k. Σ_k is defined to be a $k \times k$ diagonal matrix with $\sigma_1, \ldots, \sigma_k$ along the diagonal. Then clearly, $BV_r = U_r \Sigma_r$. Let V be any orthogonal matrix obtained by extending V_r by including any columns that are orthogonal to the columns in V_r and to each other. Note that these newly added columns must be orthogonal to the rows of B. Likewise add columns to U_r to extend it to an orthogonal matrix. Also extend Σ_r into an $n \times m$ matrix by adding 0s. Then we have $BV = U\Sigma$. Alternately, we can write this as a decomposition of B, noting that $V^T = V^{-1}$.

$$B = U\Sigma V^T$$

This is the Singular Value Decomposition (SVD) of B and it can be computed in time $O(mn^2 + m^2 n)$, see [2].

We note that it is easily proved that $\sum_{i=1}^{k} B_i$ is the best rank k approximation to B:

$$||B - \sum_{i=1}^{k} B_i||_F = \min_{R|rank(R)=k} ||B - R||_F$$

Further the error $||B - \sum_{i=0}^{k} B_i||_F = \sqrt{\sigma_{k+1}^2 + \ldots + \sigma_r^2}$.

1.1 Examples

Our first example decomposition is

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0.85 & -0.53 \\ 0.53 & 0.85 \end{pmatrix} \begin{pmatrix} 1.62 & 0 \\ 0 & 0.62 \end{pmatrix} \begin{pmatrix} 0.53 & 0.85 \\ -0.85 & 0.53 \end{pmatrix}$$

Note here that direction of the first right singular vector $\begin{pmatrix} 0.53 \\ 0.85 \end{pmatrix}$ is somewhere between the directions of the rows of A.

Our second decomposition example is

$$B = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

In this case note that the first right singular vector $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ snaps to the direction of the first row, does not point somewhere in between the directions of the two rows. In some sense it can be seen to identify the *dominant* direction. This happens when the rows are orthogonal, unlike the case for matrix A earlier.

Our final example consists of the matrrix

$$C = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

C has a block diagonal structure, with the blocks being of size 3×3 and 3×4. In this case it turns out that the matrix inherits the singular values of the blocks. The singular vectors are also suitably padded by zeros and inherited. In this case the largest singular value 2.36 comes from the second block, and the second largest, 2.00 from the first block. Both these are singular values of C. The first singular vector of the second block is $(.56\ .44\ .44\ .56)^T$, and so the first singular vector of C will be $(0\ 0\ 0\ .56\ .44\ .44\ .56)^T$. The next singular value 2 comes from block 1. The singular vector in that block is $(0.58\ 0.58\ 0.58)^T$. So the second singular vector for C will be $(0.58\ 0.58\ 0.58\ 0\ 0\ 0\ 0)^T$.

1.2 Singular values vs. Eigenvalues

The problem of finding the first singular vector v_1, i.e. the problem of maximizing $||Bv_1||_2^2$ can be analyzed using Lagrangian multipliers (for simplicity we maximize the square of the length rather than the length itself). We are maximizing $f(x) = ||Bx||_2^2 = x^T B^T B x$ subject to $g(x) = ||xx^T||_2^2 - 1 = 0$. Thus we need $\nabla f = \lambda \nabla g$. Let $P = B^T B$. Then $f(x) = x^T P x$, and $\frac{\partial f}{\partial x_i} = \frac{\partial}{\partial x_i}(p_{ii}x_i^2 + \sum_{j \neq i}(p_{ij} + p_{ji})x_i x_j) = 2\sum_j p_{ij}x_j$, noting that $p_{ij} = p_{ji}$. Thus

$\nabla f = 2Px$. Similarly $\nabla g = 2x$. Thus we need $Px = \lambda x$, i.e. $B^TBx = \lambda x$. Thus v_1 must simply be the largest eigenvector of B^TB. In a similar manner we argue that u_1 is the largest eigenvector of BB^T.

We have that $u_i^TBB^T = \sigma_i v_i^TB^T = \sigma_i(Bv_i)^T = \sigma_i^2 u_i^T$. Thus every u_i is an eigenvector of BB^T of eigenvalue σ_i^2. Likewise v_i is a eigenvector of B^TB of eigenvalue σ_i^2. Note that the eigenvalues are squares of the singular values.

The matrices BB^T and B^TB are symmetrical matrices, $n \times n$ and $m \times m$ respectively. So we expect them to have respectively n and m orthogonal eigenvectors. On the other hand, our process of determining singular vectors only give r singular vectors, where r is the rank of B. How do we get the additional eigenvectors? This may be resolved as follows. We simply interprete the extra columns (if any) added to V also as right singular vectors of singular value 0 – notice that these columns are orthogonal to each row of B and hence their product with B is indeed 0. Likewise we may interprete the additional columns added to U as left singular vectors, also of singular value 0.

The following facts are important for finding eigenvalues and eigenvectors and are easily verified.

1. If λ, x are respectively an eigenvalue and eigenvector of a matrix M, then $k + l\lambda, x$ are of $kI + lM$, for any real l, k.

2. If λ, x are respectively an eigenvalue and eigenvector of a matrix M, then $\lambda, Q^{-1}x$ are of $Q^{-1}MQ$, for any matrix Q.

Using the first and noting that $BB^T = A + D$, and noting that $D = dI$ for d-regular graphs, we have that the adjacency matrix of a d-regular graph has eignevalues $\sigma_i^2 - d$, and the eigenvectors are same as the left singular vectors u_i of BB^T.

The second will be useful for normalized adjacency matrix, see the exercises.

While singular vectors can be thought of as approximating the directions of the rows/columns of a matrix, such an interpretation is not interesting for eigenvectors.

2 Drawing regular graphs

Let $u_1, u2, \ldots$ and $v_1, v_2, \ldots$ and $\sigma_1, \sigma_2, \ldots$ respectively denote the left singular vectors, right singular vectors and the singular values of B. Then v_1 gives an

approximation to the direction of the rows of B in the Frobenius norm. It will be seen that the point cloud is relatively far from the origin, and hence v_1 merely gives the general direction in which the point cloud lies. From the point of view of understanding the shape of the point cloud, v_1 is not interesting.

The second right singular vector v_2 is an approximation to the general direction of the rows of A after removing the components in the direction of v_1. It turns out that this roughy discovers the "length" of the point cloud! The next singular vector v_3 likewise discovers the "breadth", albeit in a heuristic sense.

So for the purpose of drawing, we consider the dimensions v_2, v_3. So we need coordinates of all vertices along these dimensions. The coordinate of a vector B_i along direction v_2 is simply $b_i \cdot v_2$. Thus $B \cdot v_2$ gives the coordinates of all vertices in a single vector. Note further that $B \cdot v_2 = \sigma_2 u_2$. Thus u_2 gives the coordinates of each vertex in the direction v_2 (within scaling) and likewise u_3 the coordinate along direction v_3.

Next, note that u_2 is the second left eigenvector of BB^T. But $BB^T = A+D$. For a k-regular graph, $D = kI$. Hence BB^T has the same eigenvectors as A, or for that matter $D^{-1}A$. So for regular graphs, this discussion indicates that we may draw a graph using the eigenvectors of A or those of $W = D^{-1}A = A/k$.

2.1 General graphs

It appears that using the eigenvectors of the walk matrix also works reasonably for general graphs[3]. Note however that W does not have the same eigenvectors as BB^T for general graphs. Nor are the eigenvectors of W the same as those of CC^T, although they are related, as will be seen in the next section.

Let x_2, x_3 be the second and third eigenvectors, and α_2, α_3 be the second and third eigenvalues of W. Then $Wx_2 = \alpha_2 x_2$, and hence $x_2(i) = (Wx_2)_i/\alpha_2 = (\sum_{j|(i,j)\in E} x_2(j)/d_i)/\alpha_2$. Now $\sum_{j|(i,j)\in E} x_2(j)/d_i$ is simply the mean of the coordinates of the neighbours of i. Thus this embedding places vertex i at $(x_2(i), x_3(i))$ which is nearly the center of mass of where its neighbours are placed, if α_2, α_3 are reasonably close to 1. For the binary hypercube Q_n, the second and third eigenvalues are $1 - 2/n$, and are thus close to 1 for large n. A similar situation also holds for the grid.

3 Graph Partitioning

We have looked at the mincut problem, where we wanted to split a graph by removing as few edges as possible. However, in that problem, there was no restriction on how large the parts had to be.

In many applications, it is required that a none of the parts be too small. Say you have a circuit that you want to place on multiple printed circuit boards. Typically, within a board there can be many wires, but each board has only a few "pins" to connect to the external world. So the partitioning must have the property that relatively large subgraphs are created, but with as few wires connecting the subgraphs as possible. In divide and conquer graph algorithms, it is again useful to split graphs into equal parts; the amount of work that needs to be done while putting together the solutions for the parts depends upon the number of edges cut, and this needs to be minimized.

We might ask for graph bisection, and ask for algorithms that find the *bisection width*, i.e. the minimum number of edges that need to be removed so that the resulting two subgraphs have equal number of vertices. In many cases, a more general definition might be more useful.

Let $G = (V, E)$ be the graph and $S \subset V$. Let $E(S, V - S)$ denote the number of edges with one endpoint in S and the other in $\bar{S} = V - S$.

The sparsity of a cut $(S, \bar{S})$ is defined as

$$\text{sp}(S, V - S) = \frac{|E(S, V - S)|}{\min(|S|, |V - S|)}$$

The sparsity of a graph is defined as $\text{sp}(G) = \min_S \text{sp}(S, V - S)$.

Another measure is the ratio of a cut:

$$r(S, V - S) = \frac{|E(S, V - S)|}{|S||V - S|}$$

It is easily seen that $r(S, V - S)|V|/2 \leq \text{sp}(S, V - S) \leq r(S, V - S)|V|$, so these measures are not really too different. The latter might be preferred only because it seems to avoid the non-linear operator min. The cut ratio for a graph is defined as $r(G) = \min_S r(S, V - S)$.

We would like to find the (cuts corresponding to) $r(G)$ or $\text{sp}(G)$.

3.1 Formulation as a numerical optimization

This formulation is from Spielman[3]. Represent a set of vertices S by its characteristic vector x, i.e. $x_i = 1$ if $i \in S$ and 0 otherwise. We first consider the ratio cut $r(S) = \frac{|E(S,V-S)|}{|S||V-S|}$ because it is algebraically easier:

The numerator may be written as $\sum_{(i,j)\in E}(x_i - x_j)^2$; for each edge that crosses from S to $V - S$ we get a 1, so that the total is $|E(S, V - S)|$.

The denominator requires a contribution of 1 for every i, j where $i \in S, j \in \bar{S}$. Thus we may write this as $\sum_{i>j}(x_i - x_j)^2$. So the problem of finding a set of minimum ratio is equivalent to solving

$$r_G = \min_S r(S) = \min_{x\in\{0,1\}^n} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i>j}(x_i - x_j)^2}$$

Solving for integer values is harder than solving for reals, typically. So we let x range over real vectors. This will allow the right hand side expression to possibly take even smaller values than what might be possible with integers, so we only get a lower bound. Note further that since only differences $x_i - x_j$ appear in the expression, we can uniformly increment all numbers without changing the value. So we may assert $\sum_i x_i = 0$, i.e. $x \cdot \mathbf{1} = 0$, or $x \perp \mathbf{1}$. Thus we get

$$r_G \geq \min_{x\perp\mathbf{1}} \frac{\sum_{(i,j)\in E}(x_i - x_j)^2}{\sum_{i>j}(x_i - x_j)^2}$$

The numerator and the denominator are quadratic in x_i, hence they can be written in the form $x^T H x$ where H is a suitably chosen matrix. For the numerator we use $H = L = D - A$ which is called the Laplacian of the graph. It maybe seen that $x^T L x = \sum_{i,j} l_{ij} x_i x_j = \sum_i d_i x_i^2 - \sum_{i\neq j} a_{ij} x_i x_j = \sum_i d_i x_i^2 - \sum_{(i,j)\in E} 2x_i x_j = \sum_{(i,j)\in E}(x_i - x_j)^2$. Further note that the denominator

$$\sum_{i>j}(x_i - x_j)^2 = (n-1)\sum_i x_i^2 - 2\sum_{i>j} x_i x_j = n\sum_i x_i^2 - (\sum_i x_i)^2 = n\sum_i x_i^2$$

since $\sum_i x_i = 0$ by assumption. Thus we get

$$r_G \geq \min_{x\perp\mathbf{1}} \frac{x^T L x}{n x^T x}$$

Notice that the denominator $x^T x$ in this expression is simply the Euclidean length of x. Asserting that this be 1 does not change the minimum. Thus we may write:

$$r_G \geq \min_{x\perp\mathbf{1}, ||x||_2=1} \frac{x^T L x}{n}$$

Since L is symmetric, it has real eigenvalues $\mu_1, \ldots, \mu_n$ from smallest to largest (our ordering was reverse of this for eigenvalues and singular values so far), and orthogonal eigenvectors $z_1, \ldots, z_n$. It is also easily seen that $\mu_1 = 0$ and $z_1 = \mathbf{1}$.

We next resolve x along the eigenvectors z_i, i.e. let c_i be such that $x = \sum_i c_i z_i$. Then $Lx = \sum_i c_i L z_i = \sum_i c_i \mu_i z_i$. Noting that $z_i \cdot z_j = 0$ for $i \neq j$:

$$x^T L x = (\sum_i c_i z_i) \cdot (\sum_i c_i \mu_i z_i) = \sum_i c_i^2 \mu_i$$

Since we want to minimize $x^T L x$, it would be desirable to make c_i large for smaller indices, since μ_i is non-increasing. Note now that $x \perp \mathbf{1} = z_1$. Thus $c_1 = 0$. Thus to minimize we should set $c_2 = 1$ whereupon $x^T L x = \mu_2 x^T x$. Thus we have shown

$$r_G \geq \frac{\mu_2}{n}$$

3.2 Algorithm

Does the previous formulation help in designing an algorithm?

It is instructive to consider a similar but far simpler problem. Suppose we want to find an integer x such that $x^2 + 3.1x$ is minimized. We would naturally do this by relaxing the condition that x be real, and taking the derivative, assert that $2x + 3.1 = 0$. This would yield $x = -1.55$. We would then venture that a *nearby* integer value, -2 or -1 would have to be the answer. We would probably just check both and pick the smaller. Note however that the integer minimum -2.2 in this case, is not the same as the real minimum, -2.4025.

We can try something similar for our problem as well. We compute $x = z_2$, the second smallest eigenvector of the Laplacian matrix L. Then we somehow determine an S from x. Remember that we started off by assuming x to be a bit-vector, which we then shifted by an amount say α so that $x \cdot \mathbf{1} = 0$. In the new x vector there were only 2 values: α, wherever there was a 0 before, and $1 + \alpha$, wherever there was a 1 before. If z_2 has exactly this form, we can right away decide which vertex to put in S.

But in general z_2 will not have this form. So what do we do? Perhaps those vertices i whose coordinate $z_2(i)$ is large should be put in S. But how do we decide what "large" means? The simple answer is we try out all possibilities! After all, there are only n possible thresholds $z_2(j)$ for the

different values of j. So we define $S_j = \{i | z_2(i) > z_2(j)\}$. We calculate $r(S_j, V - S_j)$ and pick the minimum.

So the next question is: how much worse than μ_2/n can the cut ratio of the cut selected above be?

3.3 Cheeger's Theorem

Theorem 1 (Cheeger) *Let L be the Laplacian matrix of a graph G, and $\mu_1, \ldots, \mu_n$ its eigenvalues ordered from the smallest to the largest. Then*

$$\frac{r(G)^2 n}{8 d_{max}} \leq \frac{\mu_2}{n} \leq r(G)$$

where d_{max} is the maximum degree. Further there is a simple algorithm to find a cut matching the first inequality, i.e. S such that

$$r(S, V - S) \leq \frac{\sqrt{8 d_{max} \mu_2}}{n}$$

The second inequality we have proved. The first inequality is substantially harder, and we will not prove. But we note that the algorithm referred to in the theorem is indeed the algorithm we discussed, and it may not find a cut of ratio μ_2/n, but will certainly find a cut of ratio at most $\frac{\sqrt{8 d_{max} \mu_2}}{n}$. Note that the cut found need not be the best, but the best cut ratio must be somewhere between $\frac{\mu_2}{n}$ and $\frac{\sqrt{8 d_{max} \mu_2}}{n}$.

Sometimes the bounds are stated in terms of sparsity:

$$\frac{\mathrm{sp}(G)^2}{2 d_{max}} \leq \mu_2 \leq 2\,\mathrm{sp}(G)$$

Also, we can find a cut $(S, V - S)$ such that $\mathrm{sp}(S, V - S) \leq \sqrt{2 d_{max} \mu_2}$.

Cheeger's theorem may be used to ascertain how good spectral partitioning (the algorithm discussed above) works; we can also use it to bound the second eigenvalue of graphs whose sparsity (or cut ratio) is known. We discuss both these possibilities in the examples below.

Our first example is the cycle C_n. For this, it can be shown that $\mu_2 = O(1/n^2)$. Thus we can find a cut of sparsity $O(\sqrt{2 \cdot 2 \cdot 1/n^2}) = O(1/n)$. We know this to be true: this cut is obtained by bisecting the cycle, and indeed has the smallest sparsity. Thus spectral partitioning has worked well.

Knowing the sparsity to be $\theta(1/n)$, Cheeger's theorem would have established that μ_2 is between $\theta(1/n^2)$ and $\theta(1/n)$.

Our second example is the binary hypercube. We will shortly establish that for $Q_{\log n}$, $\mu_2 = 2$. This gives us that the sparsity of the cut we get using the algorithm would be $\sqrt{4 \log n}$, whereas we know that cuts of sparsity 1 can be obtained, say by removing the edges along any dimension.

Our final example is the class of planar graphs, for which it can be shown[4] that $\mu_2 = O(1/n)$. This establishes that spectral partitioning would give us a cut of sparsity $O(\sqrt{d_{max}/n})$. For graphs of bounded degree, this becomes $O(1/\sqrt{n})$. While for specific planar graphs we can get cuts of smaller sparsity (e.g. for C_n), in general improving this is not possible. For example, for $P_{\sqrt{n}} \Box P_{\sqrt{n}}$, sparsity is $1/\sqrt{n}$. Spectral partitioning does not guarantee balance, i.e. the two sides of the cut may have very different number of vertices. However, by repeatedly applying spectral partitioning on a bounded degree planar graph, it may be seen that we can get a 1/3–2/3 separation if we wish, while keeping the number of edges removed $O(\sqrt{n})$. Thus this may be considered to be another proof of the planar separator theorem (though it requires μ_2 to be computed by independent means) for bounded degree graphs.

3.4 Relationship to incidence matrix embedding

For k-regular graphs, $L = D - A = kI - A$. Thus L has the same eigenvectors as A. However, the order is reversed. The eigenvector with the ith smallest eigenvalue in L has the ith largest eigenvalue in A. Note further that $BB^T = A + D = kI + A$ also has same eigenvectors as A, and hence L. Thus the left singular vector u_i of B of singular value σ_i is the same as the eigenvector z_i of L of eigenvalue μ_i.

So the algorithm above was effectively: order vertices according to their coordinate in $u_2 = z_2$, and construct set S_j by picking those vertices i whose coordinate $u_2(i) = z_2(i)$ is greater than $u_2(j)$. Of all sets S_j return the one which has the least cut ratio.

If we believe that the point cloud obtained from the vertices has its "length" along direction v_2, then it would seem natural to believe that the "cross section" perpendicular to v_2 is the "narrowest". So if we want to cut very few edges, we should perhaps be partitioning the graph by slicing perpendicular to v_2. In other words we project each vertex onto the direc-

tion v_2, and then take a set of contiguous vertices in that order. But note that the projection of the vertex i is simply the ith component of Bv_2. But $Bv_2 = \sigma_2 u_2$. Since we do not care about the scale, it is enough to consider the entries in u_2 as giving the projections of the vertices. So to get a "narrow" cross section, we should take all vertices having their u_2 value above some threshold. This is exactly what the algorithm described above does!

3.5 General graphs

The algorithm actually also works for general graphs; however, there is no clear correspondence with the edge incidence embedding. This is because for general graphs the vectors u_2 and z_2 will be different.

However, for general graphs, using the normalized incidence matrix does give an algorithm that finds a cut of the so-called minimum conductance (ratio of number of edges cut to the mnimum of the sums of the degrees on either side of the cut). The second singular vector of the normalized incidence matrix is in fact the same as the second eigenvector of the so called normalized Laplacian that is encountered. So even in this algorithm can be viewed as slicing the point cloud perpendicular to the second singular vector.

But the details get very complicated[1].

4 Eigenvalues of Expander graphs

A (non-bipartite) (α, β, n, d) expander is a graph d regular $G = (V, E)$ with $|V| = n$ in which any subset S of V has at least $\beta|S|$ neighbours outside of S. We can use randomized constructions to construct such expanders.

Given such an expander, we note that by Cheeger's theorem, $\mu_2 \leq r(G)n$. If $\Gamma(S)$ denotes the neighbourhood of S, then we have $|\Gamma(S)| \geq |E(S, V - S)/d|$. Thus $\frac{|\Gamma(S)|}{|S|d} \geq \frac{r(S,V-S)|V-S|}{d} \geq r(G)\frac{(1-\alpha)n}{d} \geq \frac{\mu_2(1-\alpha)}{d}$. Thus, if $\frac{\mu_2(1-\alpha)}{d} \geq \beta$, we can rest assured that G is indeed an (α, β, n, d) expander.

Something similar works also for the bipartite case.

5 Eigenvalues of the hypercube

It is quite easy to find the eigenvalues of the adjacency matrix of the hypercube by exploiting its recursive structure. First consider the adjacency

matrix.

By simple algebra, it can easily be seen that the $-1, 1$ are the eigenvalues of Q_2. Now observe that if A_i is the adjacency matrix of Q_i, then we have:

$$A_{i+1} = \begin{pmatrix} A_i & I \\ I & A_i \end{pmatrix}$$

From this it is easily verified that if λ is any eigenvalue of A_i and x the corresponding eigenvector, then $\lambda \pm 1$ are eigenvalues of A_{i+1}, and $\begin{pmatrix} x \\ \pm x \end{pmatrix}$ are the corresponding eigenvectors.

Thus it follows that the adjacency matrix of Q_n has eigenvalues $n - 2i$ with multiplicity $\binom{n}{i}$, for $i = 0$ to n. Since Q_n is a regular graph, we can show that the Laplacian will have eigenvalues $2i$ with multiplicity $\binom{n}{i}$, $i = 0$ to n. Thus the second smallest eigenvalue is 2 as claimed earlier.

6 Concluding Remarks

Singular value decomposition is a standard data analysis technique. It has been used in analysis of web pages for searching, and also various aspects of image processing.

Many other algorithms are known for graph partitioning. Some of these (implicitly) build up on the algorithm we discussed.

In general, the idea of viewing a graph (or indeed other objects too) in a high dimensional space is very powerful. Dont beat up yourself if you cannot *imagine* more than 3 dimensions. It is not necessary to do so! Often, we only look at 2-3 dimensions out of the many, and restricted to these, our notions of perpendicularity, projections, etc. work nicely. If we need to deal with many dimensions at the same time our intuition about volumes, areas etc. will not work; so we make guesses by analogy, prove them correct by algebra, and then proceed.

Exercises

1. Suppose B is the node-edge adjacency matrix of a graph G, for a certain numbering of the vertices and edges (remember: the ith vertex corresponds to the i th row, and jth edge to the jth column). Suppose

u_i, σ_i, v_i are the left singular vector, singular value and right singular vector of B. Suppose now that the vertices and edges of G are numbered differently, giving a new matrix B'. Suppose we find its singular vectors and values u'_i, σ'_i, v'_i. How will these compare to u_i, σ_i, v_i?

2. From first principles (i.e. by using elementary calculus) find the singular value decompositions for the matrices

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$$

You may check correctness by using a program such as scilab.

3. Find the largest singular value and the range of singular values for the edge incidence matrix of K_4, the complete graph on 4 vertices.

4. Show that for a regular graph, the first singular vector passes through the center of mass of the vertices.

5. Draw a picture of the edge incidence embedding of all connected graphs having 3 edges.

6. Show that the first singular value of a d-regular graph is $\sqrt{2d}$. What are the first singular vectors? Hint: Thanks to Anvit Singh: the following corollary of the Cauchy Schwarz inequality could be useful: $(\sum_{i=1}^{k} x_i)^2 \leq k(\sum_{i=1}^{k} x_i^2)$.

7. Let B denote the node-edge incidence matrix for a possibly non regular graph G in which vertex i has degree d_i. Let d be the maximum degree. Suppose for each i we add $d - d_i$ columns having a 1 in row i and zeros elsewhere, giving us a matrix B'. What is $B'B'^T$? What can you say about its eigenvectors and eigenvalues?

8. Show that the normalized node-edge incidence matrix C as defined in the text has the same eigenvectors and eigenvalues as the matrix $D^{-1/2}LD^{-1/2}$, the normalized Laplacian. Note that $L = D - A$ is the Laplacian. What is the relationship between the eigenvalues of the ordinary and normalized Laplacians?

9. Show that if σ_i, u_i, v_i are singular values and singular vectors of B, then σ_i^2 are eigenvalues of BB^T and B^TB, and u_i, v_i are eigenvectors of BB^T, B^TB respectively.

10. Suppose you have a solid body made up of unit masses (discretized) at points $p_1, \ldots, p_n$. Suppose the center of mass of the body is at the origin. The first principal axis of rotation of the solid body is that line through the origin which minimizes $\sum_j r_j^2$, where r_j is the perpendicular distance to the line from p_j. Show that the first principal axis of rotation can be found using singular value decomposition.

11. "Spectral partitioning" essentially partitions a graph as follows: (a) Embed it in some Euclidean space, say as per the rows of the node edge incidence matrix. (b) Find a suitable hyperplane H and determine L, R = sets of vertices on either side of H in the space. (c) Return the subgraphs induced by L, R. An interesting question is, does there always exist a Hyperplane H such that it will return a given partition.

    Consider C_4, the cycle on 4 vertices. Let B be its edge incidence matrix. Let the cycle be embedded as per the edge incidence embedding. (a) Find a hyperplane that has vertices 2,3 on one side and 4,1 on the other. (b) Find a hyperplane that has vertex 1 on one side and 2,3,4 on the other side. (c) Prove that there is no hyperplane that has vertices 1,3 on one side and vertices 2,4 on the other side. (d) Let G', G'' be partitions of a graph G. Show that if G', G'' are connected, then there exists a hyperplane which will return G', G'' given the node-edge incidence embedding of G.

12. Show that the expression $x_1^2 - 3x_1x_2 - 4x_3^2 + 2x_1x_3$ can be expressed in the form $x^T P x$ where $x^T = (x_1, x_2, x_3)$, where P is a symmetric matrix.

13. Show that all eigenvalues of any Laplacian matrix L are non-negative, and observe that $\mathbf{1}$ is an eigenvector of eigenvalue 0.

14. What is $\mathrm{sp}(G)$ where G is (a) Q_n, (b) $K_{3n/4}$ connected by a single edge to $K_{n/4}$, (c) $P_{\sqrt{n}} \Box P_{\sqrt{n}}$.

15. The public domain program `scilab` is very useful to experiment with singular values and eigenvalues. It contains simple commands to manipulate matrices, e.g. get their singular/eigenvalues and vectors etc. (a) Use scilab to draw Q_3 using the second and third singular value. Also try other graphs. (b) Use scilab to find a cut of small ratio of different graphs using the second smallest eigenvector of the Laplacian,

say $P_n \Box P_n$. (c) Use scilab to find μ_2 for a random d-regular graph for various values of d. Comment on how much expansion you get.

16. *Research Question:* The node-edge adjacency embedding given by the rows of B has the properties that for d-regular graphs, adjacent vertices are embedded at a distance $\sqrt{2d-1}$, and non-adjacent vertices at a distance $\sqrt{2d}$. Suppose we are given an arbitrary embedding (in an arbitrary number of dimensions) having these two properties. Will SVD be able to get reasonable partition from that embedding? You might think that it might suffice to ask for non-neighbours to be embedded *at least* a certain distance rather than exactly a certain distance. Show that this relaxation does not work: There exists an embedding of a graph such that each vertex is at a distance $\sqrt{2d-1}$ from its neighbour, and such that non-neighbours are at a distance *at least* (rather than exactly) $\sqrt{2d}$, but does not have any hyperplane separator of low ratio. Hint: it suffices to choose the graph to be a cycle.

References

[1] F. Chung. *Spectral Graph Theory.* Americal Mathematical Society, 1997.

[2] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 1996.

[3] D. Spielman. Graphs and networks: Random walks and spectral graph drawing, September 2007. www.cs.yale.edu/homes/spielman/462/lect4-07.pdf.

[4] D. Spielman and S. Teng. Spectral partitioning works: Planar graphs and finite element meshes. In *Proceedings of the IEEE Annual Symposium on The Foundations of Computer Science*, pages 1105–1115, 1996.