

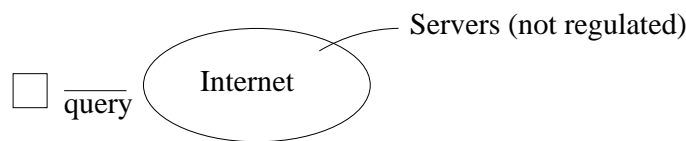
Lecture 1 — September 15, 2002

Lecturer: Shang-Hua Teng

Scribe: Ben Hescott

1.1 Topics covered in course

1. Information Infrastructure/ Internet queries are more complex than structured queries of relational databases.



Considering the Internet from a database perspective it is not just a database, but a distributed database. Given this perspective we can address many algorithmic questions

- Where is the content? (distributed DB)
- How to answer a query (search engine)
- Information gathering (distributed and parallel crawler)
- Information organization (data mining, clustering, information hierarchy and taxonomy)
- Structure of Web

2. Networking

- Efficient Communication
- Network Infrastructures
 - (a) Traffic Control
 - (b) Load Balance
 - (c) Networking Addressing
 - (d) Data Compression - need to find efficient ways to compress
 - (e) Error correcting codes - fault tolerance

3. Parallel Processing

- Complex Design
- Task Partitioning

- Load Balancing

4. Modern Algorithm Analysis

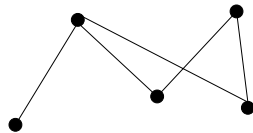
- Heuristics - Algorithms used by web, circuit design, etc. (many times algorithms used today do not "look good" for traditional algorithmic analysis) Today careful testing needed for cost trade offs, in this course will look to smooth analysis. Idea is to try to analyze the heuristics that are working in practice. It is a "physics type" approach, observe it then find a mathematical model.
- Study behavior of special class of inputs (some inputs have a great deal of parallelism)

Problem is where to begin, first study concrete to try understand web design, and load balancing issues.

1.2 Spectral Techniques for Optimization

Spectral generally refers to eigenvalues and eigenvectors. An example is how does Google rate the relevance of particular search results. To find out first define a graph.

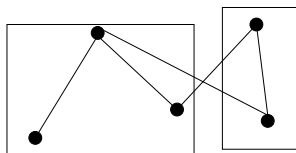
Definition 1.1 (Graph). A graph, G , is a set (V, E) consisting of a set of nodes or vertices V , and set edges, E . Note: $E \subset V \times V$



May be directed or undirected

Example: References in Documents - Let each paper be a node and each citation an edge to another paper(node). Edges can be directed, paper A cites paper B. Some more examples are friends, web-graph (similar to reference documents), maps, circuits, nearest neighbor in geometry.

Graph Partition Problem: Consider as input an unweighted, undirected graph $G(V,E)$ and an integer $k > 1$. Want to partition into k pieces. $V = V_1 \cup V_2 \cup V_3 \cup \dots \cup V_k$ Normally want to minimize or maximize a partition. The cost of partition is defined as $(V_1, V_2, V_3, \dots, V_k) = \# \text{edges whose endpoints are in different set}$.



Cost of partition is 2

Idea is want to minimize cost, note that this is trivial if you put everything in one set, no partitions, no cost. You generally want to partition subject to other conditions. The balanced condition is subject to exact k-way partition $|v_i| \leq \lceil \frac{n}{k} \rceil$, i.e. we want balanced partition into k pieces. So given a 5 node graph and want to partition into 2 pieces, want 2 and 3, not 1 and 4 element subsets. Note when k=2 this is bisection problem, this problem is believed to be NP-hard.

Definition 1.2 (NP-Hard). A problem A is said to be NP-Hard if $\forall B \in NP, B \leq_T A$. Specifically every problem in NP can be solved using an oracle to A . Note that it is not necessarily true that A be in NP.

Given that, this is an optimization problem we can deal with it one of two ways, approximation, or heuristics. An algorithm is an α -approximation ($\alpha \geq 1$) for exact k-way partition $\Leftrightarrow \text{cost}(A) \leq \alpha \times \text{cost-optimal}(\ast)$. Bisection problem does not have a known constant approximation. Today we will consider heuristic approaches, the technique used by Google.

Mathematical programming $V = (1, 2, \dots, n)$ $(x_1, x_2, x_3, \dots, x_n) \in \{\pm 1\}$ where -1 denotes left side, +1 denotes right side. We will conquer the problem without eigenvalues, but rather with graph theory. Consider n even and see how far we can go.

We know

1. $\sum x_i = 0$
2. $\sum x_i^2 = n$

Now cost is

$$E(i, j) \in E$$

$$(x_i - x_j)^2 = \begin{cases} 1 \\ 4 \end{cases}$$

And the cost of partition $4 \times \text{Cost} = \sum_{(i,j) \in E} (x_i - x_j)^2$

So we want to minimize this cost subject to

$$\sum x_i = 0$$

$$\sum x_i^2 = n$$

But the problem we have is that each $x_i \in \mathbb{Z}$. Now consider matrix representation of graphs, idea is need to be able to describe eigenvalue/eigenvectors in graph.

Definition 1.3 (Incidence Matrix). The incidence matrix is the N by E matrix representing graph G , where N is the number of nodes and E is the number of edges in G , whose entries $x_{(i,j)} = 1$ if n_i is the positive end of edge $e_{i,j}$, -1 if it is the negative edge, 0 otherwise.

So given graph G as follows

1. L is symmetric $L^T = L$
2. Sum of any row is zero.
3. $L = M \cdot M^T$

Definition 1.7 (Eigenvector). A vector \vec{x} is called the eigenvector of L if \exists a scalar γ s.t. $L \cdot \vec{x} = \gamma \cdot \vec{x}$ Furthermore γ is called the eigenvalue.

So for the Laplacian we have an eigenvector of $\begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$ and an eigenvalue of 0. Note that

this comes easily as we have a row sum of zero for the Laplacian. The eigenvector with its corresponding eigenvalue is called the spectral of the matrix.

Now a heuristic argument for property 3, $L = M \cdot M^T$. First consider

$$\vec{x}^T \cdot L \cdot \vec{x} = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix} \cdot L \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \sum_{(i,j) \in E} (x_i - x_j)^2$$

Now look at property 3

$$\vec{x}^T L \vec{x} = \vec{x}^T M M^T \vec{x} = \langle M^T \vec{x}, M^T \vec{x} \rangle$$

Which is the dot product, notice the degree of $M^T \cdot \vec{x}$ is $|E| |\vec{x}|$ Also notice that each entry of $M^T \vec{x}$ is $(x_i - x_j)_{(i,j) \in E}$ so

$$\langle M^T \vec{x}, M^T \vec{x} \rangle = \sum_{(i,j) \in E} (x_i - x_j)^2$$

So the problem of minimize cost of a partition reduces to minimizing $\vec{x}^T \cdot L \cdot \vec{x}$ subject to $\sum x_i = 0$ and $\sum x_i^2 = 1$ Now consider translation (normalization) that is $y_i = \frac{x_i}{\sqrt{n}}$ $\sum y_i = 0$ and $\sum y_i^2 = \sum \frac{x_i^2}{n} = 1$ So we want to minimize $\vec{y}^T \cdot \vec{y}$ subject to $\sum y_i = 0$ and $\sum y_i^2 = 1$

Consider L, it has n eigenvalues, say $\gamma_1, \gamma_2, \dots, \gamma_n$ Since L is symmetric then γ 's are \mathbb{R} So we can order them say $\gamma_1 \leq \gamma_2 \leq \gamma_3 \leq \dots \leq \gamma_n$ Recall that since $L = M \cdot M^T$ that L is positive definite, and then $\forall \vec{x}, \vec{x}^T \cdot L \cdot \vec{x} \geq 0$

Notice here that $\gamma_1 = 0$, so $u_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$ Also recall that for a symmetric matrix all

eigenvectors are orthogonal, ie $u_1 \perp u_2 \Leftrightarrow \langle u_1, u_2 \rangle = 0$

Claim second eigenvector is solution to minimizing $\vec{y}^T L \vec{y}$ with $\sum y_i = 0$ and $\sum y_i^2 = 1$

Consider the power up matrix of L.

$$Lx, L^2x, L^3x, \dots$$

$$x = \sum \alpha_i u_i$$

$$L^k x = \sum \alpha_i \gamma_i^k u_i$$

Now consider the Rayleigh Quotient $\gamma_{\vec{x}}(L) = \frac{\vec{x}^T L \vec{x}}{\vec{x}^T \vec{x}}$

Smallest eigenvalue of $L = \gamma_1 = \min_x \gamma_x(L)$ So $\gamma_2 = \min_{x \perp \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}} \frac{\vec{x}^T L \vec{x}}{\vec{x}^T \vec{x}}$

Also note that we can ignore $x^T x$ if vectors are normalized.

So it is efficient to find eigenvector - why is this not a solution? When u_2 comes back it is possible it is not equal to ± 1 , it may not even be an integer. You find a solution that is guaranteed \leq optimal, but when you use it you may do it wrong.

1.3 Spectral Bisection Algorithm

1. Form L from $D - A$
2. Compute u_2 , second eigenvector of L
3. $u = u_2$, called the Fiedler vector $u = (w_1, w_2, \dots, w_n)$
4. $L = \{i, w_i \leq \text{median}\}$, $R = \{i, w_i \geq \text{median}\}$

Spectral Embedding of Graph Question: Why just on eigenvector?

Consider 2 eigenvectors u_2, u_3 with $u_2 = (w_1, w_2, \dots, w_n)$ and $u_3 = (z_1, z_2, \dots, z_n)$ We have embedding of graph, but we have more freedom in finding median, division need not be perpendicular to u_2, u_3 . It is unknown how to increase these eigenvectors and use them in algorithm.