CS 5614: (Big) Data Management Systems

B. Aditya Prakash

Lecture #21: Graph Mining 2
We often think of networks being organized into modules, cluster, communities:
Goal: Find Densely Linked Clusters
Micro-Markets in Sponsored Search

- Find micro-markets by partitioning the query-to-advertiser graph:

[Andersen, Lang: Communities from seed sets, 2006]
Movies and Actors

- Clusters in Movies-to-Actors graph:

[Andersen, Lang: Communities from seed sets, 2006]
Discovering social circles, circles of trust:

- friends under the same advisor
- CS department friends
- college friends
- ‘alters’ $v_i$
- ‘ego’ $u$
- family members
- highschool friends

[McAuley, Leskovec: Discovering social circles in ego networks, 2012]
We will work with **undirected** (unweighted) networks.

How to find communities?

COMMUNITY DETECTION
Method 1: Strength of Weak Ties

- **Edge betweenness**: Number of shortest paths passing over the edge
- **Intuition:**

![Edge strengths (call volume) in a real network](image1)

![Edge betweenness in a real network](image2)

Prakash 2017
Method 1: Girvan-Newman

- Divisive hierarchical clustering based on the notion of edge *betweenness*:
  Number of shortest paths passing through the edge

- **Girvan-Newman Algorithm:**
  - Undirected unweighted networks
  - **Repeat until no edges are left:**
    - Calculate betweenness of edges
    - Remove edges with highest betweenness
  - Connected components are communities
  - Gives a hierarchical decomposition of the network
Girvan-Newman: Example

Need to re-compute betweenness at every step
Girvan-Newman: Example

Step 1:

Step 2:

Step 3:

Hierarchical network decomposition:

Prakash 2017

VT CS 561
Girvan-Newman: Results

Communities in physics collaborations

Prakash 2017
Girvan-Newman: Results

- Zachary’s Karate club: Hierarchical decomposition
WE NEED TO RESOLVE 2 QUESTIONS

1. How to compute betweenness?
2. How to select the number of clusters?
How to Compute Betweenness?

- Want to compute betweenness of paths starting at node A

Breath first search starting from A:
How to Compute Betweenness?

- Count the number of shortest paths from \( A \) to all other nodes of the network:
How to Compute Betweenness?

- **Compute betweenness by working up the tree:** If there are multiple paths count them fractionally.

**The algorithm:**
- **Add edge flows:**
  - node flow = \(1 + \sum \text{child edges}\)
  - split the flow up based on the parent value
- **Repeat the BFS procedure for each starting node** \(U\)

---

Prakash 2017
How to Compute Betweenness?

- **Compute betweenness by working up the tree**: If there are multiple paths count them fractionally

**The algorithm:**
- Add edge flows:
  - node flow = $1 + \sum \text{child edges}$
  - split the flow up based on the parent value
- Repeat the BFS procedure for each starting node $U$
WE NEED TO RESOLVE 2 QUESTIONS

1. How to compute betweenness?
2. How to select the number of clusters?
Network Communities

- **Communities**: sets of tightly connected nodes

- **Define**: Modularity $Q$
  - A measure of how well a network is partitioned into communities
  
  Given a partitioning of the network into groups $s \in S$
  
  $$Q \propto \sum_{s \in S} \left[ \text{(# edges within group } s) - \text{(expected # edges within group } s) \right]$$

Need a null model!
Null Model: Configuration Model

- Given real $G$ on $n$ nodes and $m$ edges, construct rewired network $G'$
  - Same degree distribution but random connections
  - Consider $G'$ as a multigraph

- The expected number of edges between nodes $i$ and $j$ of degrees $k_i$ and $k_j$ equals to: $k_i \cdot \frac{k_j}{2m} = \frac{k_i k_j}{2m}$
  - The expected number of edges in (multigraph) $G'$:
    - $= \frac{1}{2} \sum_{i \in N} \sum_{j \in N} \frac{k_i k_j}{2m} = \frac{1}{2} \cdot \frac{1}{2m} \sum_{i \in N} k_i (\sum_{j \in N} k_j) = $
    - $= \frac{1}{4m} 2m \cdot 2m = m$

Note: $\sum_{u \in N} k_u = 2m$
Modularity of partitioning $S$ of graph $G$:

\[ Q \propto \sum_{s \in S} \left[ (\text{# edges within group } s) - (\text{expected # edges within group } s) \right] \]

\[ Q(G, S) = \frac{1}{2m} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} \left( A_{ij} - \frac{k_i k_j}{2m} \right) \]

Normalizing cost.: $-1 < Q < 1$

Modularity values take range $[-1,1]$

- It is positive if the number of edges within groups exceeds the expected number
- $0.3 - 0.7 < Q$ means significant community structure
Modularity: Number of clusters

- Modularity is useful for selecting the number of clusters:
SPECTRAL CLUSTERING
Graph Partitioning

- Undirected graph

- Bi-partitioning task:
  - Divide vertices into two disjoint groups

- Questions:
  - How can we define a “good” partition of ?
  - How can we efficiently identify such a partition?
Graph Partitioning

- What makes a good partition?
  - Maximize the number of within-group connections
  - Minimize the number of between-group connections
Graph Cuts

- Express partitioning objectives as a function of the “edge cut” of the partition
- **Cut:** Set of edges with only one vertex in a group:

\[
cut(A, B) = \sum_{i \in A, j \in B} w_{ij}
\]

![Diagram showing a graph with two sets A and B, and the calculation of cut(A, B) = 2](image)
Graph Cut Criterion

- **Criterion:** Minimum-cut
  - Minimize weight of connections between groups
    \[ \arg \min_{A,B} \, \text{cut}(A,B) \]

- **Degenerate case:**

- **Problem:**
  - Only considers external cluster connections
  - Does not consider internal cluster connectivity
Graph Cut Criteria

- **Criterion: **Normalized-cut [Shi-Malik, ’97]
  - Connectivity between groups relative to the density of each group
  
  \[ ncut(A, B) = \frac{cut(A, B)}{vol(A)} + \frac{cut(A, B)}{vol(B)} \]
  
  : total weight of the edges with at least one endpoint in :

  **Why use this criterion?**
  - Produces more balanced partitions

- **How do we efficiently find a good partition?**
  - **Problem:** Computing optimal cut is NP-hard
Spectral Graph Partitioning

- **$A$: adjacency matrix of undirected $G$**
  - $A_{ij} = 1$ if $i$ is an edge, else $0$

- **$x$** is a vector in $\mathbb{R}^n$ with components
  - Think of it as a label/value of each node of

- **What is the meaning of $A \cdot x$?**

\[
\begin{bmatrix}
  a_{11} & \cdots & a_{1n} \\
  \vdots   & \ddots & \vdots \\
  a_{n1} & \cdots & a_{nn}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  \vdots \\
  x_n
\end{bmatrix}
= 
\begin{bmatrix}
  y_1 \\
  \vdots \\
  y_n
\end{bmatrix}
\]

\[
y_i = \sum_{j=1}^{n} A_{ij} x_j = \sum_{(i,j) \in E} x_j
\]

- **Entry $y_i$ is a sum of labels $x_j$ of neighbors of $i$**
What is the meaning of $A\mathbf{x}$?

- **$j^{th}$ coordinate of $A\cdot\mathbf{x}$:**
  - Sum of the $\mathbf{x}$-values of neighbors of $j$
  - Make this a new value at node $j$

- **Spectral Graph Theory:**
  - Analyze the “spectrum” of matrix representing
  - **Spectrum:** Eigenvectors of a graph, ordered by the magnitude (strength) of their corresponding eigenvalues:

$$
\mathbf{A}\cdot\mathbf{x} = \lambda\cdot\mathbf{x}
$$

$$
\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}
$$

$$
\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n
$$
Example: d-regular graph

- Suppose all nodes in $G$ have degree $d$ and $G$ is connected.

- What are some eigenvalues/vectors of $G$?
  
  $A \cdot x = \lambda \cdot x$  
  What is $\lambda$?  What $x$?

  - Let’s try: $x = (1, 1, \ldots, 1)$
  - Then: $A \cdot x = (d, d, \ldots, d) = \lambda \cdot x$. So: $\lambda = d$
  - We found eigenpair of $G$: $x = (1, 1, \ldots, 1)$, $\lambda = d$

Remember the meaning of $y = A \cdot x$:

$$y_j = \sum_{i=1}^{n} A_{ij} x_i = \sum_{(j,i) \in E} x_i$$
\( d \) is the largest eigenvalue of \( A \)

- **G** is \( d \)-regular connected, \( A \) is its adjacency matrix
- **Claim:**
  - \( d \) is largest eigenvalue of \( A \),
  - \( d \) has multiplicity of 1 (there is only 1 eigenvector associated with eigenvalue \( d \))
- **Proof:** **Why no eigenvalue \( d' > d \)?**
  - To obtain \( d \) we needed \( x_i = x_j \) for every \( i, j \)
  - This means \( x = c \cdot (1, 1, \ldots, 1) \) for some const. \( c \)
  - **Define:** \( S \) = nodes \( i \) with maximum possible value of \( x_i \)
  - Then consider some vector \( y \) which is not a multiple of vector \( (1, \ldots, 1) \). So not all nodes \( i \) (with labels \( y_i \)) are in \( S \)
  - Consider some node \( j \in S \) and a neighbor \( i \notin S \) then node \( j \) gets a value strictly less than \( d \)
  - So \( y \) is not eigenvector! And so \( d \) is the largest eigenvalue!
Example: Graph on 2 components

- **What if** $G$ **is not connected?**
  - $G$ has 2 components, each $d$-regular
- **What are some eigenvectors?**
  - $x = \text{Put all 1s on } A \text{ and 0s on } B \text{ or vice versa}$
    - $x' = (\underbrace{1, \ldots, 1}_{|A|}, \underbrace{0, \ldots, 0}_{|B|})$ then $A \cdot x' = (d, \ldots, d, 0, \ldots, 0)$
    - $x'' = (0, \ldots, 0, 1, \ldots, 1)$ then $A \cdot x'' = (0, \ldots, 0, d, \ldots, d)$
    - And so in both cases the corresponding $\lambda = d$

- **A bit of intuition:**
  - $\lambda_n = \lambda_{n-1}$
  - $\lambda_n - \lambda_{n-1} \approx 0$
  - $2^{nd}$ largest eigval.
  - $\lambda_{n-1}$ now has value very close to $\lambda_n$
More Intuition

- More intuition:
  \[ \lambda_n = \lambda_{n-1} \]
  \[ \lambda_n - \lambda_{n-1} \approx 0 \]

- If the graph is connected (right example) then we already know that \( x_n = (1, \ldots, 1) \) is an eigenvector.
- Since eigenvectors are orthogonal then the components of \( x_{n-1} \) sum to 0.
  - Why? Because \( x_n \cdot x_{n-1} = \sum_i x_n[i] \cdot x_{n-1}[i] \)
- So we can look at the eigenvector of the 2\(^{nd}\) largest eigenvalue and declare nodes with positive label in \( A \) and negative label in \( B \).
- But there is still lots to sort out.
Matrix Representations

- **Adjacency matrix \((A)\):**
  - \(n \times n\) matrix
  - \(A = [a_{ij}], a_{ij} = 1\) if edge between node \(i\) and \(j\)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Important properties:**
  - Symmetric matrix
  - Eigenvectors are real and orthogonal
Matrix Representations

- **Degree matrix (D):**
  - $n \times n$ diagonal matrix
  - $D = [d_{ii}], \ d_{ii} = \text{degree of node } i$
Matrix Representations

- **Laplacian matrix** (L):
  - $n \times n$ symmetric matrix

- **What is trivial eigenpair?**
  - then and so

- **Important properties:**
  - **Eigenvalues** are non-negative real numbers
  - **Eigenvectors** are real and orthogonal

$L = D - A$
Facts about the Laplacian $L$

(a) All eigenvalues are $\geq 0$

(b) $x^T L x = \sum_{ij} L_{ij} x_i x_j \geq 0$ for every $x$

(c) $L = N^T \cdot N$

- That is, $L$ is positive semi-definite

Proof:

- (c)$\implies$(b): $x^T L x = x^T N^T N x = (xN)^T (N x) \geq 0$
  - As it is just the square of length of $N x$

- (b)$\implies$(a): Let $\lambda$ be an eigenvalue of $L$. Then by (b) $x^T L x \geq 0$ so $x^T L x = x^T \lambda x = \lambda x^T x \implies \lambda \geq 0$

- (a)$\implies$(c): is also easy! Do it yourself.
λ_2 as optimization problem

- **Fact:** For symmetric matrix \( M \):

\[
\lambda_2 = \min_{x} \frac{x^T M x}{x^T x}
\]

- **What is the meaning of \( \min x^T L x \) on \( G \)?**

\[
x^T L x = \sum_{i,j=1}^{n} L_{ij} x_i x_j = \sum_{i,j=1}^{n} (D_{ij} - A_{ij}) x_i x_j
\]

\[
= \sum_i D_{ii} x_i^2 - \sum_{(i,j)\in E} 2x_i x_j
\]

\[
= \sum_{(i,j)\in E} (x_i^2 + x_j^2 - 2x_i x_j) = \sum_{(i,j)\in E} (x_i - x_j)^2
\]

Node \( i \) has degree \( d_i \). So, value \( x_i^2 \) needs to be summed up \( d_i \) times. But each edge \((i,j)\) has two endpoints so we need \( x_i^2 + x_j^2 \)
Proof

- Write \( x \) in axes of eigenvectors \( w_1, w_2, \ldots, w_n \) of \( M \). So, \( x = \sum_i^n \alpha_i w_i \)
- Then we get: \( Mx = \sum_i \alpha_i Mw_i = \sum_i \alpha_i \lambda_i w_i \)
- So, what is \( x^T Mx \)?
  - \( x^T Mx = (\sum_i \alpha_i w_i)(\sum_i \alpha_i \lambda_i w_i) = \sum_{ij} \alpha_i \lambda_j \alpha_j w_i w_j \)
  - \( = \sum_i \alpha_i \lambda_i w_i w_i = \sum_i \lambda_i \alpha_i^2 \)
- To minimize this over all unit vectors \( x \) orthogonal to: \( w = \min \) over choices of \((\alpha_1, \ldots, \alpha_n)\) so that:
  - \( \sum \alpha_i^2 = 1 \) (unit length) \( \sum \alpha_i = 0 \) (orthogonal to \( w_1 \))
- To minimize this, set \( \alpha_2 = 1 \) and so \( \sum_i \lambda_i \alpha_i^2 = \lambda_2 \)
\( \lambda_2 \) as optimization problem

- **What else do we know about \( x \)?**
  - \( x \) is unit vector: \( \sum_i x_i^2 = 1 \)
  - \( x \) is orthogonal to 1\(^{st} \) eigenvector \((1, \ldots, 1)\) thus:
    \( \sum_i x_i \cdot 1 = \sum_i x_i = 0 \)

- **Remember:**
  \[
  \lambda_2 = \min \frac{\sum_{(i,j) \in E} (x_i - x_j)^2}{\sum_i x_i^2}
  \]
  All labelings of nodes \( i \) so that \( \sum x_i = 0 \)

  We want to assign values \( x_i \) to nodes \( i \) such that few edges cross 0.
  (we want \( x_i \) and \( x_j \) to subtract each other)

  Balance to minimize
Find Optimal Cut [Fiedler’73]

- Back to finding the optimal cut
- Express partition (A,B) as a vector
  \[ y_i = \begin{cases} 
  +1 & \text{if } i \in A \\
  -1 & \text{if } i \in B 
  \end{cases} \]
- We can minimize the cut of the partition by finding a non-trivial vector \( x \) that minimizes:
  \[ \arg \min f(y) = \sum_{(i,j)\in E} (y_i - y_j)^2 \]

  \[ y \in [-1, +1]^n \]

Can’t solve exactly. Let’s relax \( y \) and allow it to take any real value.
Rayleigh Theorem

\[
\min_{y \in \mathbb{R}^n} f(y) = \sum_{(i,j) \in E} (y_i - y_j)^2 = y^T Ly
\]

- \( \lambda_2 = \min_y f(y) \): The minimum value of \( f(y) \) is given by the 2\(^{nd}\) smallest eigenvalue \( \lambda_2 \) of the Laplacian matrix \( L \).

- \( x = \arg \min_y f(y) \): The optimal solution for \( y \) is given by the corresponding eigenvector \( x \), referred as the **Fiedler vector**.
Approx. Guarantee of Spectral

- Suppose there is a partition of $G$ into $A$ and $B$ where $|A| \leq |B|$, s.t. $\alpha = \frac{\text{(# edges from } A \text{ to } B)}{|A|}$ then $2\alpha \geq \lambda_2$
  - This is the approximation guarantee of the spectral clustering. It says the cut spectral finds is at most 2 away from the optimal one of score $\alpha$.
- **Proof:**
  - Let: $a=|A|$, $b=|B|$ and $e=\#$ edges from $A$ to $B$
  - Enough to choose some $x_i$ based on $A$ and $B$ such that: $\lambda_2 \leq \frac{\sum (x_i-x_j)^2}{\sum_i x_i^2} \leq 2\alpha$ (while also $\sum_i x_i = 0$)

$\lambda_2$ is only smaller
Approx. Guarantee of Spectral

Proof (continued):

1) Let’s set: \[ x_i = \begin{cases} \frac{-1}{a} & \text{if } i \in A \\ \frac{1}{b} & \text{if } i \in B \end{cases} \]

Let’s quickly verify that \( \sum x_i = 0 \): \( a \left( -\frac{1}{a} \right) + b \left( \frac{1}{b} \right) = 0 \)

2) Then: \( \frac{\sum(x_i - x_j)^2}{\sum x_i^2} = \frac{\sum_{i \in A,j \in B} \left( \frac{1}{b} + \frac{1}{a} \right)^2}{a \left( \frac{1}{a} \right)^2 + b \left( \frac{1}{b} \right)^2} = \frac{e \left( \frac{1}{a} + \frac{1}{b} \right)^2}{\frac{1}{a} + \frac{1}{b}} \)

\( e \left( \frac{1}{a} + \frac{1}{b} \right) \leq e \left( \frac{1}{a} + \frac{1}{a} \right) \leq e \frac{2}{a} = 2a \)

Which proves that the cost achieved by spectral is better than twice the OPT cost

e … number of edges between A and B
Approx. Guarantee of Spectral

- **Putting it all together:**

\[
2\alpha \geq \lambda_2 \geq \frac{\alpha^2}{2k_{\text{max}}}
\]

- where \(k_{\text{max}}\) is the maximum node degree in the graph
  
  - Note we only provide the 1\textsuperscript{st} part: \(2\alpha \geq \lambda_2\)
  
  - We did not prove \(\lambda_2 \geq \frac{\alpha^2}{2k_{\text{max}}}\)

- Overall this always certifies that \(\lambda_2\) always gives a useful bound
So far...

- **How to define a “good” partition of a graph?**
  - Minimize a given graph cut criterion

- **How to efficiently identify such a partition?**
  - Approximate using information provided by the eigenvalues and eigenvectors of a graph

- **Spectral Clustering**
Spectral Clustering Algorithms

- Three basic stages:
  - 1) **Pre-processing**
    - Construct a matrix representation of the graph
  - 2) **Decomposition**
    - Compute eigenvalues and eigenvectors of the matrix
    - Map each point to a lower-dimensional representation based on one or more eigenvectors
  - 3) **Grouping**
    - Assign points to two or more clusters, based on the new representation
Spectral Partitioning Algorithm

1) **Pre-processing:**
   - Build Laplacian matrix $L$ of the graph

2) **Decomposition:**
   - Find eigenvalues $\lambda$ and eigenvectors $x$ of the matrix $L$
   - Map vertices to corresponding components of $\lambda_2$

How do we now find the clusters?
3) **Grouping:**
   - Sort components of reduced 1-dimensional vector
   - Identify clusters by splitting the sorted vector in two

**How to choose a splitting point?**
- Naïve approaches:
  - Split at 0 or median value
- More expensive approaches:
  - Attempt to minimize normalized cut in 1-dimension (sweep over ordering of nodes induced by the eigenvector)

---

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>0.6</td>
<td>0.3</td>
<td>-0.3</td>
<td>-0.3</td>
<td>-0.6</td>
<td></td>
</tr>
</tbody>
</table>

**Split at 0:**
- **Cluster A:** Positive points
- **Cluster B:** Negative points

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>0.6</td>
<td>0.3</td>
<td>-0.3</td>
<td>-0.3</td>
<td>-0.6</td>
<td></td>
</tr>
</tbody>
</table>

Prakash 2017
Example: Spectral Partitioning

Value of $x_2$

Rank in $x_2$
Example: Spectral Partitioning

Components of $x_2$
Example: Spectral partitioning

Components of $x_1$

Components of $x_3$
k-Way Spectral Clustering

- How do we partition a graph into $k$ clusters?

- **Two basic approaches:**
  - **Recursive bi-partitioning** [Hagen et al., ’92]
    - Recursively apply bi-partitioning algorithm in a hierarchical divisive manner
    - Disadvantages: Inefficient, unstable
  - **Cluster multiple eigenvectors** [Shi-Malik, ’00]
    - Build a reduced space from multiple eigenvectors
    - Commonly used in recent papers
    - A preferable approach...
Why use multiple eigenvectors?

- **Approximates the optimal cut** [Shi-Malik, ’00]
  - Can be used to approximate optimal \( k \)-way normalized cut

- **Emphasizes cohesive clusters**
  - Increases the unevenness in the distribution of the data
  - Associations between similar points are amplified, associations between dissimilar points are attenuated
  - The data begins to “approximate a clustering”

- **Well-separated space**
  - Transforms data to a new “embedded space”, consisting of \( k \) orthogonal basis vectors

- **Multiple eigenvectors prevent instability due to information loss**
ANALYSIS OF LARGE GRAPHS: TRAWLING
Trawling

- Searching for small communities in the Web graph
- What is the signature of a community / discussion in a Web graph?

Dense 2-layer graph

Intuition: Many people all talking about the same things

Use this to define “topics”: What the same people on the left talk about on the right
Remember HITS!
A more well-defined problem:
Enumerate complete bipartite subgraphs $K_{s,t}$
- Where $K_{s,t}$: $s$ nodes on the “left” where each links to the same $t$ other nodes on the “right”

$|X| = s = 3$
$|Y| = t = 4$

Fully connected $K_{3,4}$
Frequent Itemset Enumeration

- **Market basket analysis.** Setting:
  - **Market:** Universe $U$ of $n$ items
  - **Baskets:** $m$ subsets of $U$: $S_1, S_2, \ldots, S_m \subseteq U$ ($S_i$ is a set of items one person bought)
  - **Support:** Frequency threshold $f$

- **Goal:**
  - Find all subsets $T$ s.t. $T \subseteq S_i$ of at least $f$ sets $S_i$ (items in $T$ were bought together at least $f$ times)

- **What’s the connection between the itemsets and complete bipartite graphs?**
From Itemsets to Bipartite $K_{s,t}$

Frequent itemsets = complete bipartite graphs!

**How?**

- View each node $i$ as a set $S_i$ of nodes $i$ points to
- $K_{s,t} = \text{a set } Y \text{ of size } t$ that occurs in $s$ sets $S_i$
- Looking for $K_{s,t} \to \text{set of frequency threshold to } s$ and look at layer $t$ – all frequent sets of size $t$

$s \ldots \text{minimum support (}|X|=s)$
$t \ldots \text{itemset size (}|Y|=t)$
From Itemsets to Bipartite $K_{s,t}$

View each node $i$ as a set $S_i$ of nodes $i$ points to

$S_i=\{a,b,c,d\}$

Find frequent itemsets:
- $s$ ... minimum support
- $t$ ... itemset size

Say we find a frequent itemset $Y=\{a,b,c\}$ of supp $s$

So, there are $s$ nodes that link to all of $\{a,b,c\}$:

We found $K_{s,t}$!

$K_{s,t}$ = a set $Y$ of size $t$ that occurs in $s$ sets $S_i$
Example (1)

- **Support threshold** \( s=2 \)
  - \( \{b,d\} \): support 3
  - \( \{e,f\} \): support 2

- **And we just found 2 bipartite subgraphs:**

  Itemsets:
  - \( a = \{b,c,d\} \)
  - \( b = \{d\} \)
  - \( c = \{b,d,e,f\} \)
  - \( d = \{e,f\} \)
  - \( e = \{b,d\} \)
  - \( f = \{\} \)
Example (2)

- **Example of a community from a web graph**

<table>
<thead>
<tr>
<th>Nodes on the right</th>
<th>Nodes on the left</th>
</tr>
</thead>
<tbody>
<tr>
<td>A community of Australian fire brigades</td>
<td></td>
</tr>
<tr>
<td><strong>Authorities</strong></td>
<td><strong>Hubs</strong></td>
</tr>
<tr>
<td>NSW Rural Fire Service Internet Site</td>
<td>New South Wales Fir...ial Australian Links</td>
</tr>
<tr>
<td>NSW Fire Brigades</td>
<td>Feuerwehrlinks Australien</td>
</tr>
<tr>
<td>Sutherland Rural Fire Service</td>
<td>FireNet Information Network</td>
</tr>
<tr>
<td>CFA: County Fire Authority</td>
<td>The Cherrybrook Rur...re Brigade Home Page</td>
</tr>
<tr>
<td>“The National Cente...ted Children’s Ho...</td>
<td>New South Wales Fir...ial Australian Links</td>
</tr>
<tr>
<td>CRAFTI Internet Connexions-INFO</td>
<td>Fire Departments, F... Information Network</td>
</tr>
<tr>
<td>Welcome to Blackwoo... Fire Safety Serv...</td>
<td>The Australian Firefighter Page</td>
</tr>
<tr>
<td>The World Famous Guestbook Server</td>
<td>Kristiansand brannv...dens brannvesener...</td>
</tr>
<tr>
<td>Wilberforce County Fire Brigade</td>
<td>Australian Fire Services Links</td>
</tr>
<tr>
<td>NEW SOUTH WALES FIR...ES 377 STATION</td>
<td>The 911 F,P,M., Fir...mp; Canada A Section</td>
</tr>
<tr>
<td>Woronora Bushfire Brigade</td>
<td>Feuerwehrlinks Australien</td>
</tr>
<tr>
<td>Mongarlowe Bush Fire – Home Page</td>
<td>Sanctuary Point Rural Fire Brigade</td>
</tr>
<tr>
<td>Golden Square Fire Brigade</td>
<td>Fire Trails “I...ghters around the...</td>
</tr>
<tr>
<td>FIREBREAK Home Page</td>
<td>FireSafe – Fire and Safety Directory</td>
</tr>
<tr>
<td>Guises Creek Volunt...ficial Home Page...</td>
<td>Kristiansand Firede...departments of th...</td>
</tr>
</tbody>
</table>

[Kumar, Raghavan, Rajagopalan, Tomkins: Trawling the Web for emerging cyber-communities 1999]  
Prakash 2017