Node Similarity

Prof. Ralucca Gera,

Applied Mathematics Dept.
Naval Postgraduate School
Monterey, California
rgera@nps.edu

Learning Outcomes

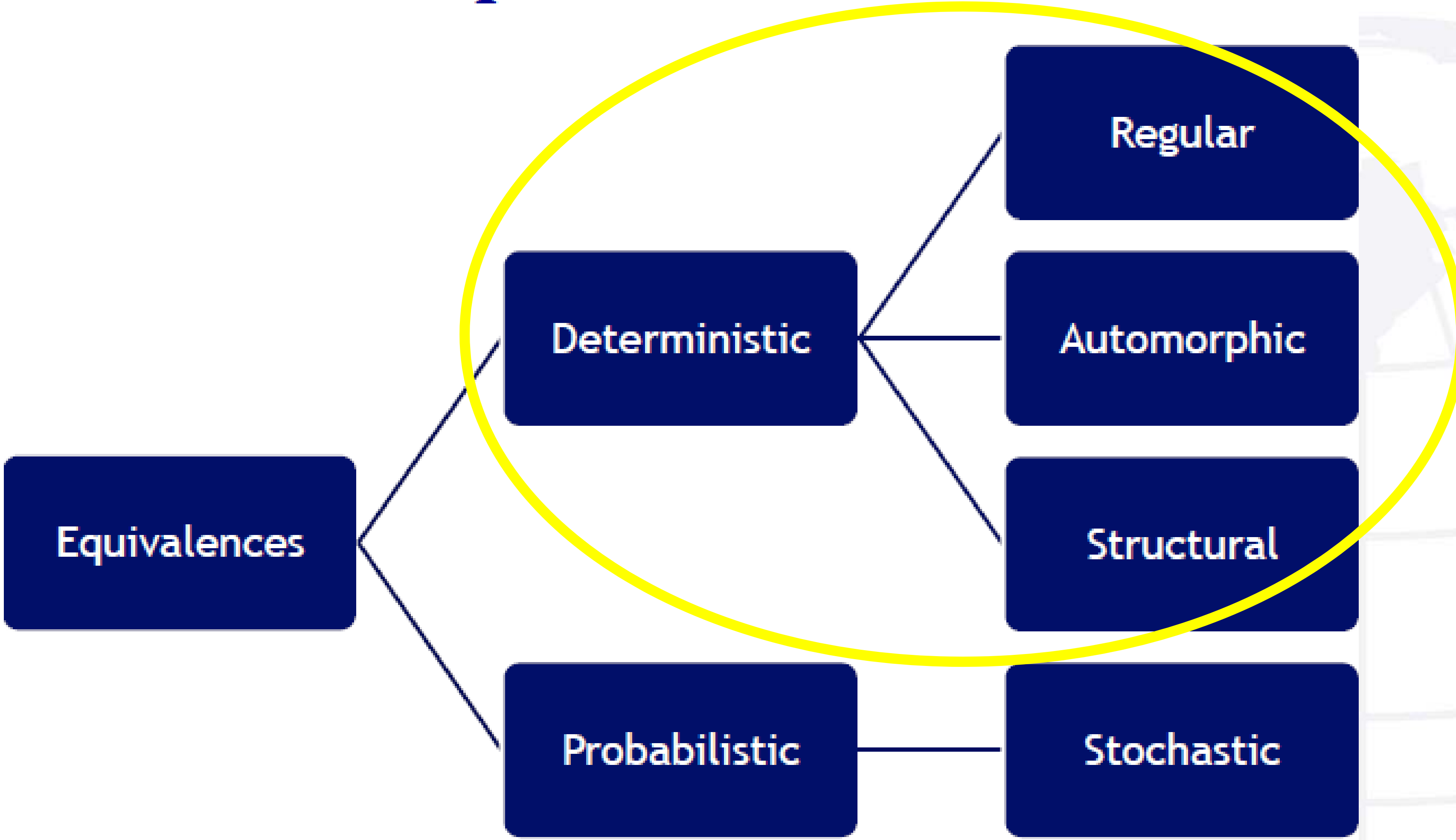
- Building on
  - global properties: average degree, average clustering, ave path length
  - local properties: some node centralities
- Understand the pairwise similarity of nodes based on position, role and structure:
  - Structural equivalence
  - Regular equivalence
  - Automorphic equivalence
- Evaluate the correlation between pairs of nodes using structural equivalence

Similarity/equivalence

- In complex network, one can measure similarity
  - Between vertices
  - Between networks
- We consider the similarity between vertices in the same network.
- Why and how can this similarity be quantified?
- Differentiating vertices helps tease apart the types and relationships of vertices
  - Useful for "click here for pages/movies/books/modules similar to the current one"
- We determine similarities based on the network structure

Types of Similarity/equivalence

Equivalences



https://web.eecs.umich.edu/~dkoutra/tut/sdm14_part1a.pdf

Introduction

- Why care? How do items get suggested to users? Based on the groups they belong to, depending on
  - The behavior of the user
  - The similarity of the users to eachother (similar attitude)

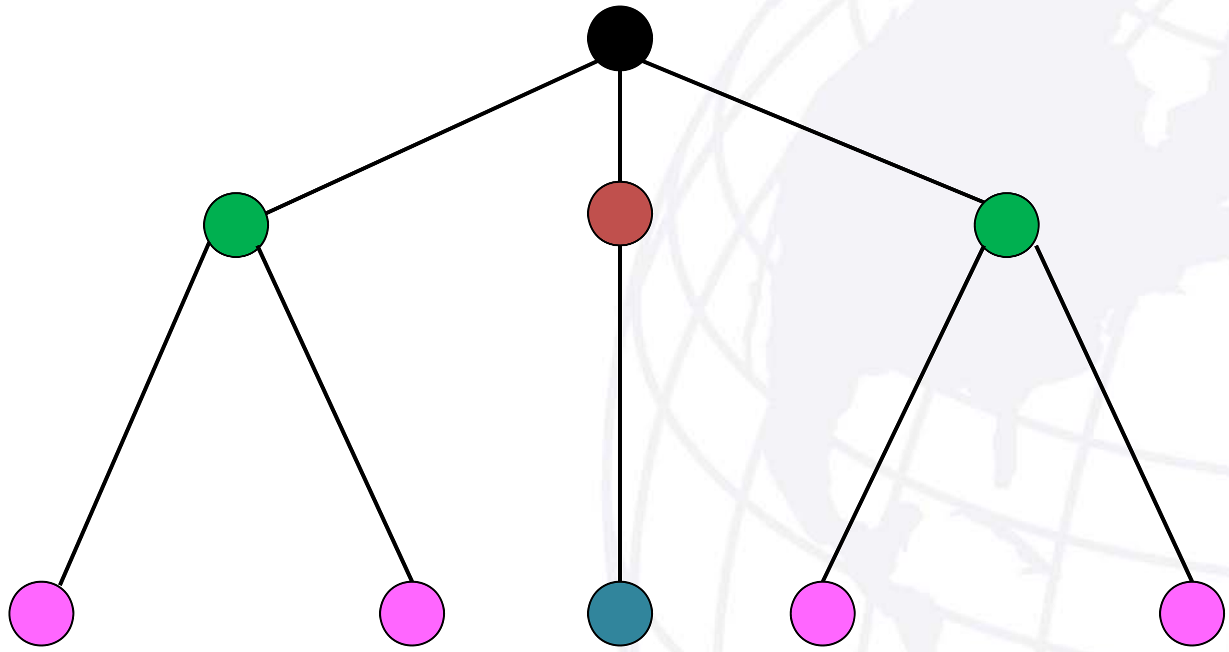
  Hypothesis: similar nodes have similar outcomes.
- There are three types of similarities:
  - Structural equivalence (such as Pearson Corr. Coeff)
  - Regular equivalence
  - Automorphic equivalence (automorphism classes)

  Of these, "automorphic" has rarely been used in substantive work, it is more theoretical than the other ones (mostly studied in graph theory: groups and graphs).

http://faculty.ucr.edu/~hanneman/nettext/C12_Equivalence.html

Automorphic equivalence

Swapping same colored nodes and their neighbors maintains all of the distances among all the nodes in the graph

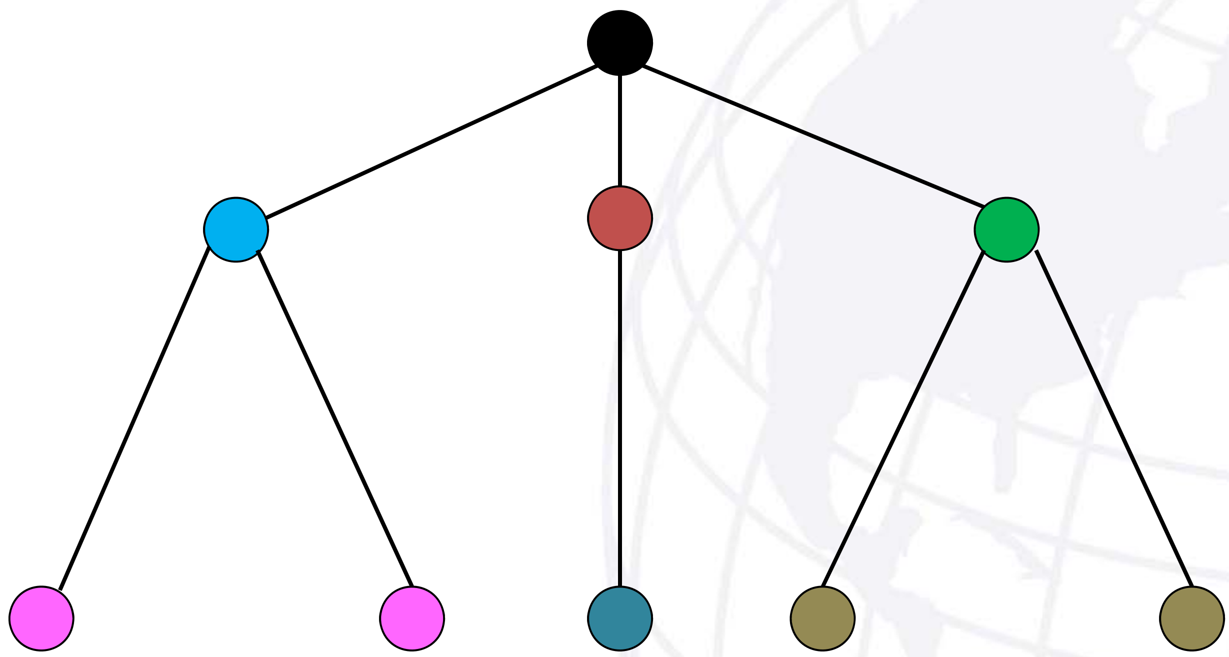


http://faculty.ucr.edu/~hanneman/nettext/C12_Equivalence.html

https://web.eecs.umich.edu/~dkoutra/tut/sdm14_part1a.pdf

Structural equivalence

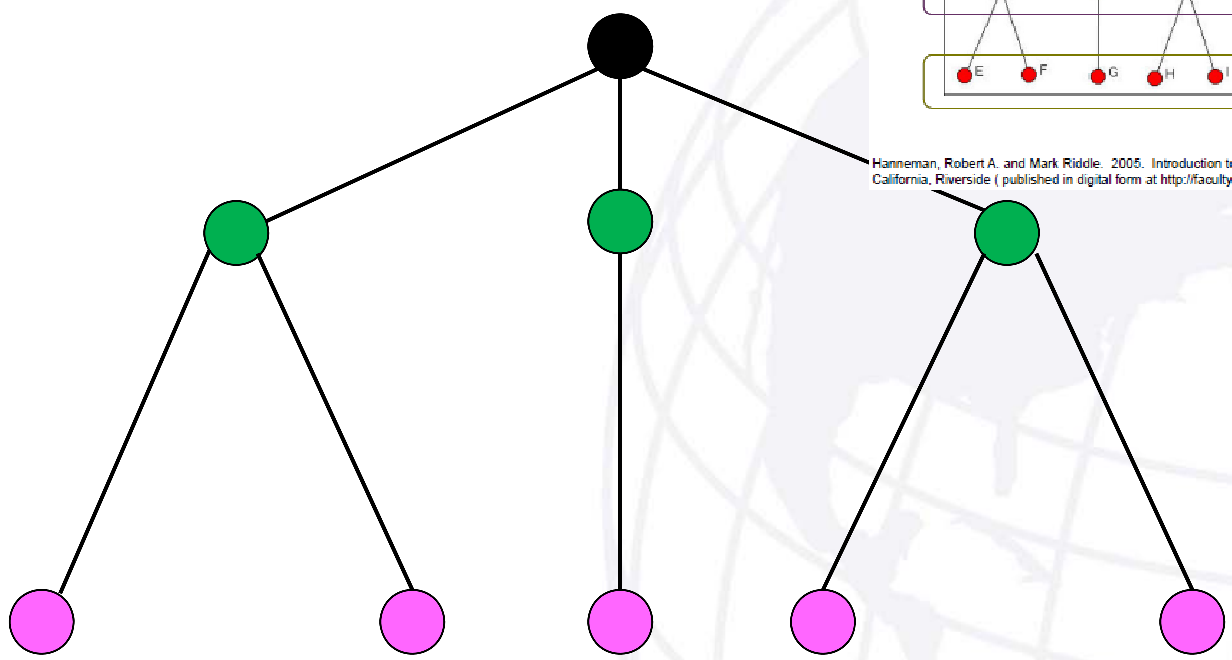
Two nodes of the same color are structurally equivalent if they have the same relationships to the other neighbors, i.e. they are substitutable

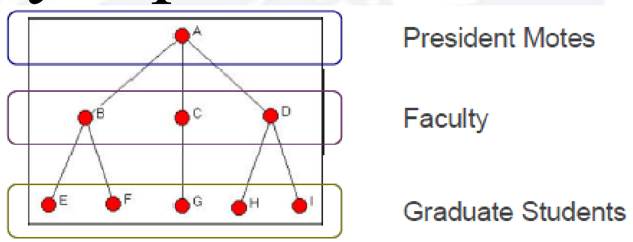


http://faculty.ucr.edu/~hanneman/nettext/C12_Equivalence.html

Regular equivalence

Two nodes are regularly equivalent if they have the same profile of ties with members of other sets of actors that are also regularly equivalent





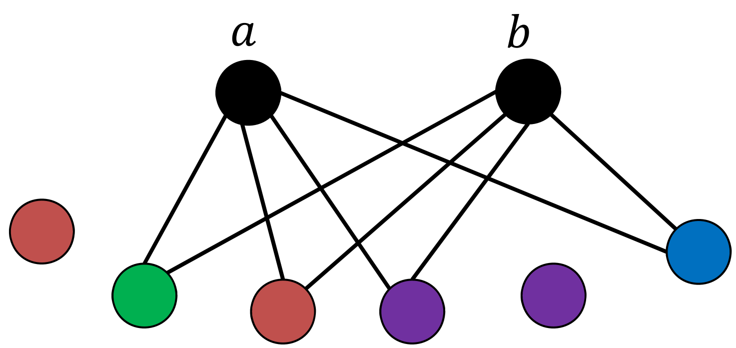
Hanneman, Robert A. and Mark Riddle. 2005. Introduction to social network methods. Riverside, CA: University of California, Riverside (published in digital form at http://faculty.ucr.edu/~hanneman/)

http://faculty.ucr.edu/~hanneman/nettext/C12_Equivalence.html

https://web.eecs.umich.edu/~dkoutra/tut/sdm14_part1a.pdf

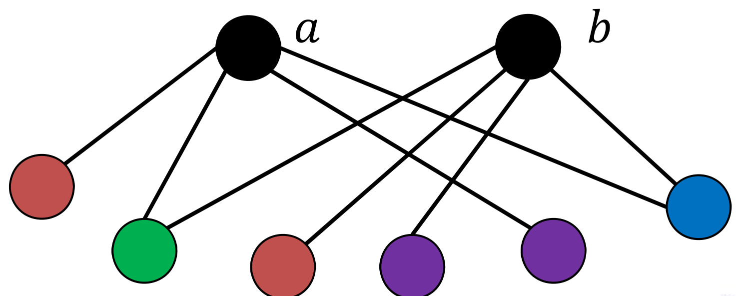
Structural vs. Regular Equivalent nodes

- Defn: Two vertices are structurally equivalent if they share many neighbors.



Such as two math students a and b that know the same professors, same topics, …, , i.e. substitutable in the field

- Defn: Two vertices are regularly equivalent if they have many neighbors that are also equivalent.

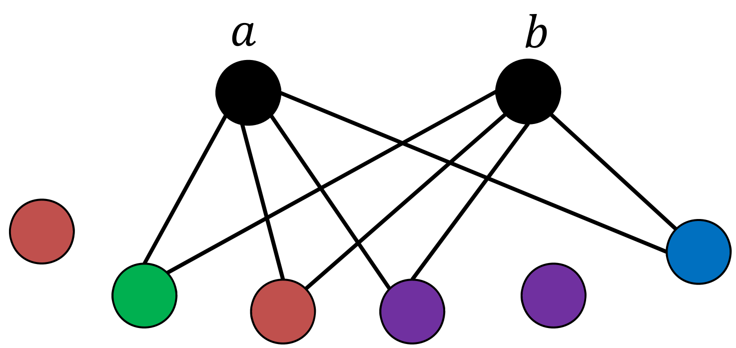


Such as two Deans a and b that have similar ties to provost, president, department chairs, …

Structural Equivalent

Structural Equivalent nodes

- Two vertices are structurally equivalent if they share many neighbors



- How can we measure this similarity?

Measuring
Structural Equivalence

Structural similarity n_{ij}

1. Count of common neighbors n_{ij}:

$$n_{ij} = |N[i] \cap N[j]| = \sum_k a_{ik} a_{kj} = a^2{}_{ij}$$

2. Normalized common neighbors (divide by a constant):

– by the number of vertices: $\frac{n_{ij}}{|V(G)|}$

– by the max number of common neighbors $\frac{n_{ij}}{|V(G)|-2}$

– Jaccard similarity is the quotient of number of common to the union of all neighbors:

$$J_{ij} = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|} = \frac{n_{ij}}{|N(i) \cup N(j)|}$$

Fast to compute (thus popular) as it just counts neighbors

P. Jaccard, Bulletin de la Soci´et´e Vaudoise des Sciences Naturelles 37, 547 (1901).

Structural similarity (2)

3. Cosine similarity (divide by variable):

Let x be the row corresponding to vertex i in A, and

Let y be the column corresponding to vertex j in A:

$$\cos\alpha = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{|\boldsymbol{x}| \cdot |\boldsymbol{y}|} = \frac{\sum_k a_{ik} a_{kj}}{\sqrt{\sum_k a_{ik}^2} \cdot \sqrt{\sum_k a_{jk}^2}} = \frac{?}{?} \text{ simplifies to}$$

$$= \frac{n_{ij}}{\sqrt{\deg i}\,\sqrt{\deg j}} = \frac{number\ of\ common\ neighbors}{\sqrt{\deg i}\,\sqrt{\deg j}}$$

Convention: If *deg i* = 0 (denominator is 0), then *cos α* = 0 (orthogonal vectors). Range: $0 \leq \cos\alpha \leq 1$

G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*(Addison-Wesley, Reading, MA, 1989).

Cosine similarity

- Cosine similarity is an example of a technique used in
  - information retrieval,
  - text analysis, or
  - any comparison of x to y, where each of x and y can be vectorized based on their components
- For example: To find the closest document(s)/website(s) to another document or a query…

Technique used in text analysis

- A collection of n documents $(D_1, D_2, \ldots, D_n)$ can be represented in the vector space model by a term-document matrix.
- A list of t terms of interest: $T_1, T_2, \ldots, T_t$
- An entry in the matrix corresponds to the "weight" of a term in the document; zero means the term has no significance in the document or it simply doesn't exist in the document.

$$
\begin{pmatrix}
 & T_1 & T_2 & \ldots & T_t \\
D_1 & w_{11} & w_{21} & \ldots & w_{t1} \\
D_2 & w_{12} & w_{22} & \ldots & w_{t2} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
D_n & w_{1n} & w_{2n} & \ldots & w_{tn}
\end{pmatrix}
$$

Source: Raymond J. Mooney, University of Texas at Austin

Graphic Representation

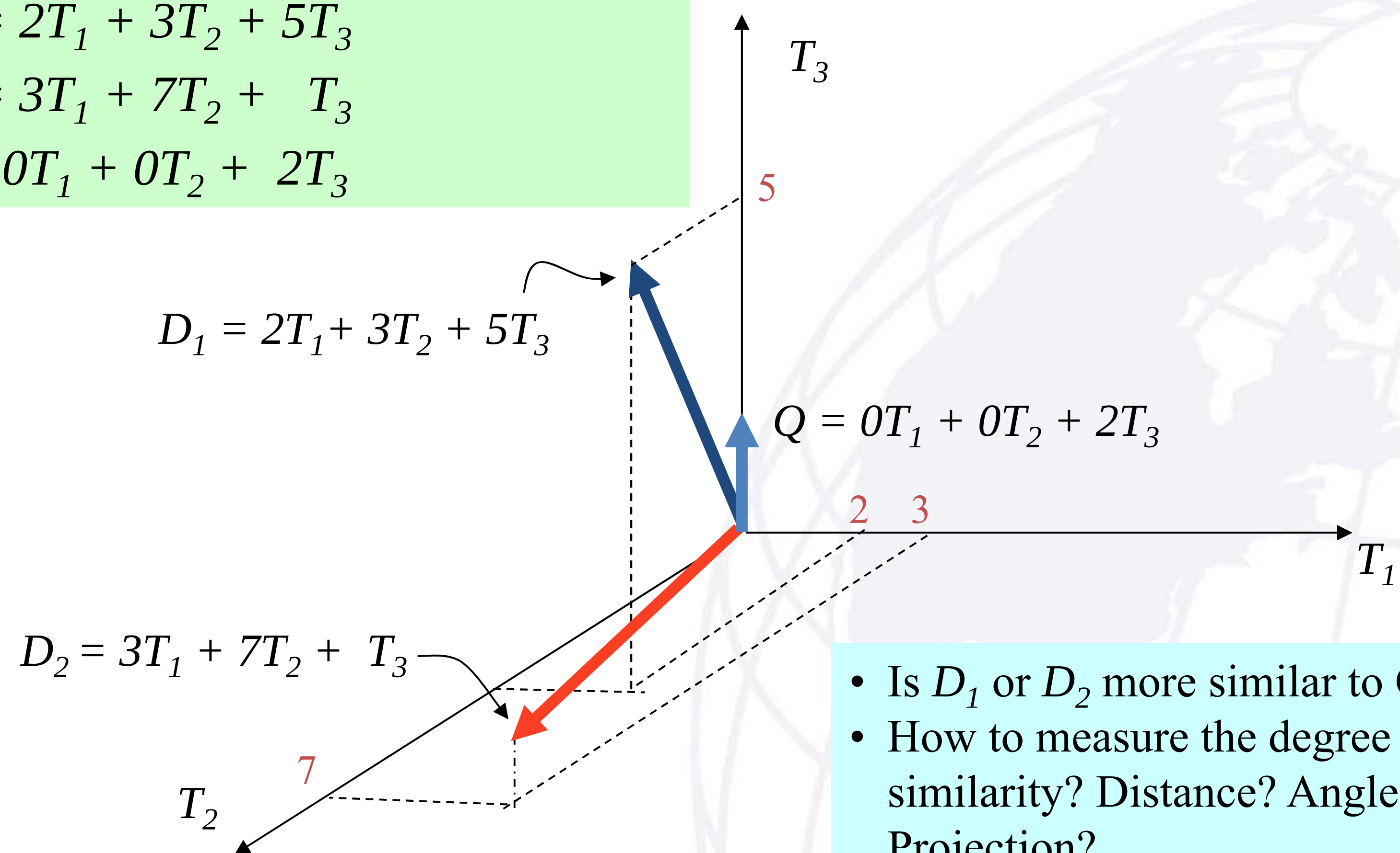
Example: *Given terms $T_{2,}$ $T_{2,}$ T_2*

Documents $D_{1,}$ D_1 and Query Q:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$

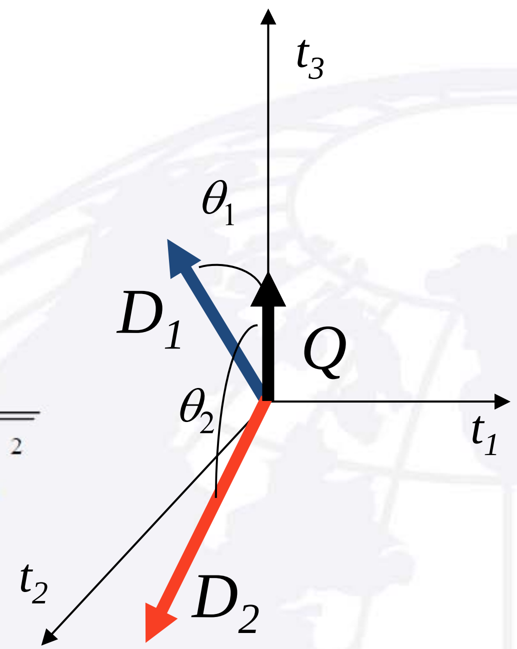


- Is D_1 or D_2 more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?

Cosine Similarity Measure

- Cosine similarity measures the cosine of the angle between two vectors.
- Inner product normalized by the vector lengths.

$$\text{CosSim}(\mathbf{d}_j, \mathbf{q}) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$

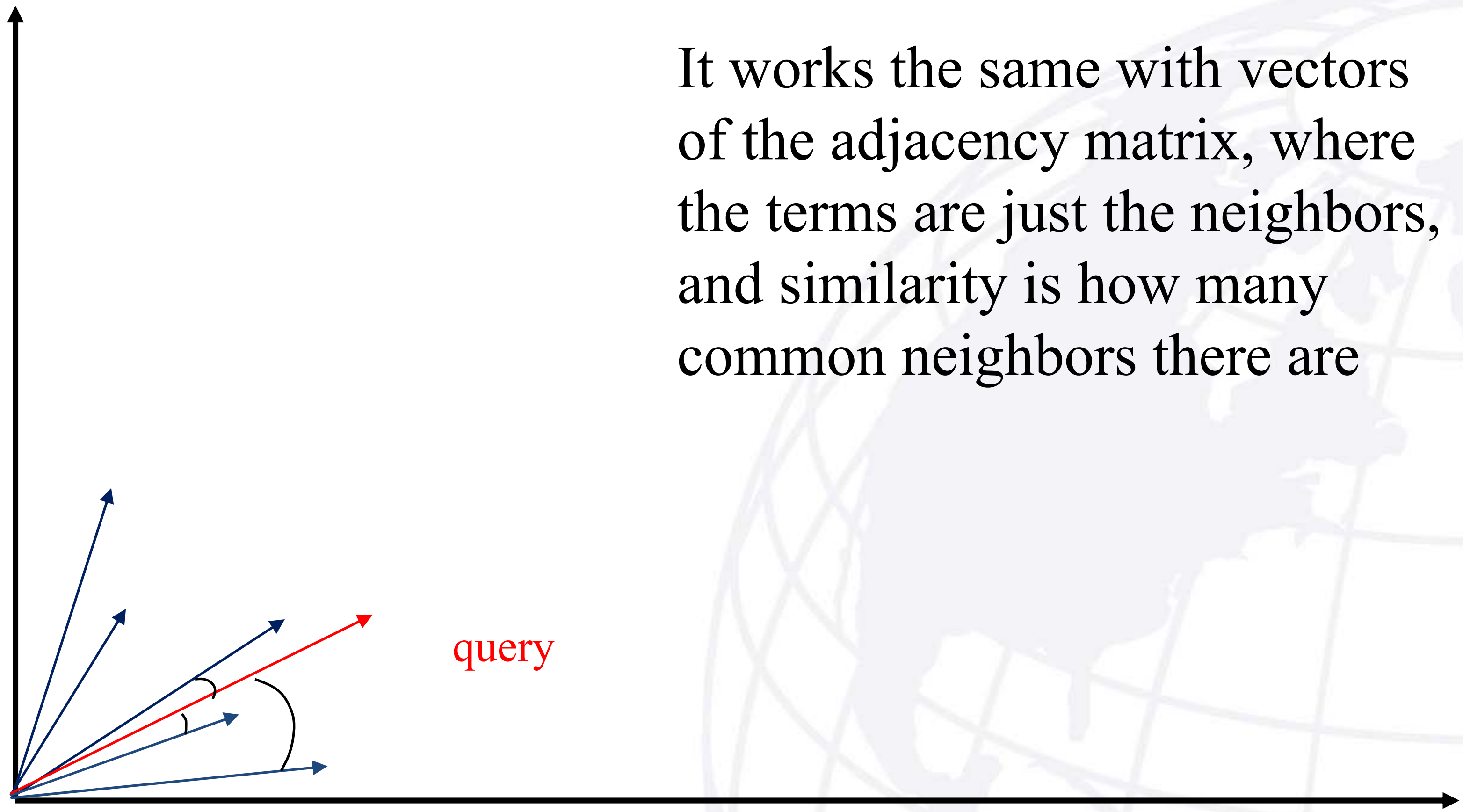


$D_1 = 2T_1 + 3T_2 + 5T_3$ CosSim(D_1 , Q) = 10 / $\sqrt{(4+9+25)(0+0+4)}$ = 0.81
$D_2 = 3T_1 + 7T_2 + 1T_3$ CosSim(D_2 , Q) = 2 / $\sqrt{(9+49+1)(0+0+4)}$ = 0.13
$Q = 0T_1 + 0T_2 + 2T_3$

D_1 is 6 times better than D_2 using cosine similarity

Source: Raymond J. Mooney, University of Texas at Austin

Illustration of 3 Nearest Neighbor for Text



It works the same with vectors of the adjacency matrix, where the terms are just the neighbors, and similarity is how many common neighbors there are

Source: Raymond J. Mooney, University of Texas at Austin

So far: Structural similarity n_{ij}

1. Count of common neighbors:

$$n_{ij} = |N[i] \cap N[j]| = \sum_k a_{ik} a_{kj} = a^2{}_{ij}$$

2. Normalized common neighbors (divide by the):

number of vertices: $\frac{n_{ij}}{|V(G)|} = \frac{a^2{}_{ij}}{|V(G)|}$

max count of common neighbors possible $\frac{n_{ij}}{|V(G)|-2} = \frac{a^2{}_{ij}}{|V(G)|-2}$

exact count of common neighbors (Jaccard similarity)

$$J_{ij} = \frac{n_{ij}}{|N(i) \cup N(j)|} = \frac{a^2{}_{ij}}{|N(i) \cup N(j)|}$$

3. Cosine similarity: $\frac{n_{ij}}{\sqrt{\deg i}\,\sqrt{\deg j}} = \frac{a^2{}_{ij}}{\sqrt{\deg i}\,\sqrt{\deg j}}$

A new one!

(Degree) Pearson Correlation Coeff (1)

4. Degree Pearson Correlation Coefficient: compare to the expected value of common neighbors (i.e. to a network in which vertices choose their neighbors at random). Here is the derivation:

- Let i and j be two vertices of *deg i* and *deg j*
- The probability of *j* to choose at random one of *i*'s neighbors is $\frac{deg\ i}{n-1}$ (or $\frac{deg\ i}{n}$ if loops in *G*)
- Thus the probability that all of *j*'s neighbors are neighbors of *i* is $\frac{deg\ i \cdot \text{deg}\ j}{n-1}$

Pearson Correlation Coeff (2)

4. Degree Pearson Correlation Coefficient is the **number of common** neighbors minus the **expected** common neighbors (that we will normalize):

$$\sum_k a_{ik}a_{jk} - \frac{\deg i \deg j}{n} = \sum_k a_{ik}a_{jk} - \frac{1}{n}\sum_k a_{ik}\sum_l a_{jl}$$

$$= \sum_k a_{ik}a_{jk} - \frac{1}{n} n < a_i > n < a_j >$$

$$= \sum_k a_{ik}a_{jk} - \sum_{k=1}^{n} < a_i >< a_j >$$

$$= \sum_k [a_{ik}a_{jk} - < a_i >< a_j >]$$

$$= \sum_k [a_{ik} - < a_i >][a_{jk} - < a_j >]$$

Since the mean of the i^{th} row is $<a_i> = \frac{\sum_k a_{ik}}{n}$

Since $\sum_k 1 = n$ as $< a_i >< a_j >$ are constants with respect to k

FOIL and simplify using $\sum_k < a_i > a_{jk} = n < a_i >< a_j > = \sum_k < a_j > a_{ik}$

NAVAL POSTGRADUATE SCHOOL

Pearson Correlation Coeff (3)

4. Degree Pearson Correlation Coefficient is the actual number of common neighbors minus the expected common neighbors (from previous page):

$$cov(a_i, a_j) = \sum_k a_{ik}a_{jk} - \frac{\deg i \deg j}{n} = \sum_k [a_{ik} - \langle a_i \rangle][a_{jk} - \langle a_j \rangle]$$

Interpretation:

the mean of the i^{th} row is $\langle a_i \rangle = \frac{\sum_k a_{ik}}{n}$

- $\sum_k a_{ik}a_{jk} - \frac{\deg i \deg j}{n} = 0$ if the # of common neighbors is what is expected by chance
- $\sum_k a_{ik}a_{jk} - \frac{\deg i \deg j}{n} < 0$ if the # of common neighbors is less than what is expected by chance
- $\sum_k a_{ik}a_{jk} - \frac{\deg i \deg j}{n} > 0$ if the # of common neighbors is more than what is expected by chance

Pearson Correlation Coeff (4)

4. Degree Pearson Correlation Coefficient is the actual number of common neighbors minus the expected common neighbors (normalized by row a_i and a_j):

the mean of the i^{th} row is $<a_i> = \frac{\sum_k a_{ik}}{n}$

$$r_{ij} = \frac{cov(a_i, a_j)}{var(a_i)^2} = \frac{cov(a_i, a_j)}{var(a_i)var(a_i)} = \frac{\sum_k [a_{ik} - <a_i>][a_{jk} - <a_j>]}{\sqrt{\sum_k [a_{ik} - <a_i>]^2}\sqrt{\sum_k [a_{jk} - <a_j>]^2}}$$

- $r_{ij} = 0$ if the # of common neighbors is what is expected by chance
- $-1 < r_{ij} < 0$ if the # of common neighbors is less than what is expected by chance
- $0 < r_{ij} < 1$ if the # of common neighbors is more than what is expected by chance
- r_{ij} shows more/less neighbor similarity in a network, compared to vertices expected to be neighbors at random

More measures

- There are more measures for structural equivalences, these are the basic ones.
- The Pearson Correlation Coefficient is commonly used
- Another one is Euclidean distance which is the cardinality of the symmetric difference between the neighbors of *i* and neighbors of *j*

$$d_{ij} = \sum_k (a_{ik} - a_{jk})(a_{jk} - a_{ik}) = \sum_k (a_{ik} - a_{jk})^2$$

All these are measures of structural equivalence of a network: they measure different ways of counting common neighbors

Overview updated: Structural similarity

1. Count of common neighbors:

$$n_{ij} = |N[i] \cap N[j]| = \sum_k a_{ik} a_{kj} = a^2{}_{ij}$$

2. Normalized common neighbors (divide by the):

number of vertices: $\frac{n_{ij}}{|V(G)|} = \frac{a^2{}_{ij}}{|V(G)|}$

max count of common neighbors possible $\frac{n_{ij}}{|V(G)|-2} = \frac{n_{ij}}{|V(G)|-2}$

exact count of common neighbors (Jaccard similarity)

$$J_{ij} = \frac{n_{ij}}{|N(i) \cup N(j)|} = \frac{a^2{}_{ij}}{|N(i) \cup N(j)|}$$

3. Cosine similarity: $\frac{n_{ij}}{\sqrt{\deg i}\sqrt{\deg j}} = \frac{a^2{}_{ij}}{\sqrt{\deg i}\sqrt{\deg j}}$

4. Degree Pearson Correlation Coefficient (coded in NetworkX) :

$$r_{ij} = \frac{cov(a_i, a_j)}{var(a_i)^2} = \frac{cov(a_i, a_j)}{var(a_i) var(a_j)} = \frac{\sum_k [a_{ik} - <a_i>][a_{jk} - <a_j>]}{\sqrt{\sum_k [a_{ik} - <a_i>]^2} \sqrt{\sum_k [a_{jk} - <a_j>]^2}}$$

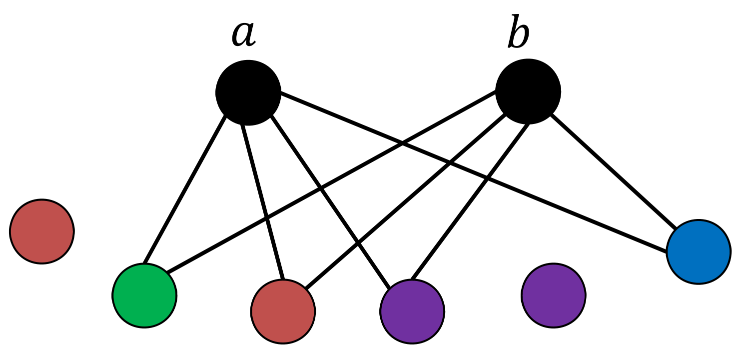
```
>>> G=nx.path_graph(4)
>>> r=nx.degree_pearson_correlation_coefficient(G)
>>> print("%3.1f"%r)
-0.5
```


https://networkx.github.io/documentation/networkx-1.9/reference/generated/networkx.algorithms.assortativity.degree_pearson_correlation_coefficient.html

Regular Equivalent

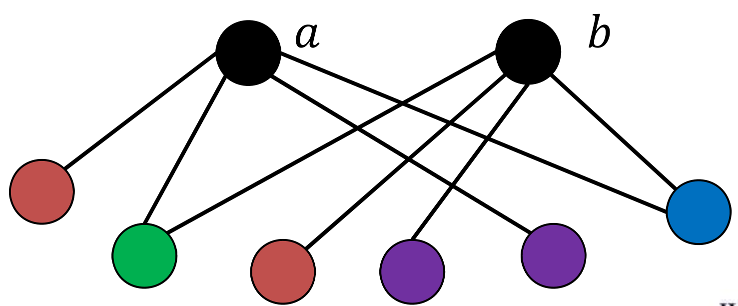
Structural vs. Regular Equivalent nodes

- Defn: Two vertices are structurally equivalent if they share many neighbors.



Such as two math Students a and b that know the same professors

- Defn: Two vertices are regularly equivalent if they have many neighbors that are also equivalent.



Such as two Deans a and b that have similar ties:
provost, president, department chairs (some could actually be common)

Regular Equivalence (cont'd)

- Based solely on the social roles of neighbors
- Interested in
  - Which nodes fall in which social roles?
  - How do social roles relate to each other?
- Hard partitioning of the graph into social roles
- A given graph can have more than one valid regular equivalence set
- Exact regular equivalences can be rare in large graphs

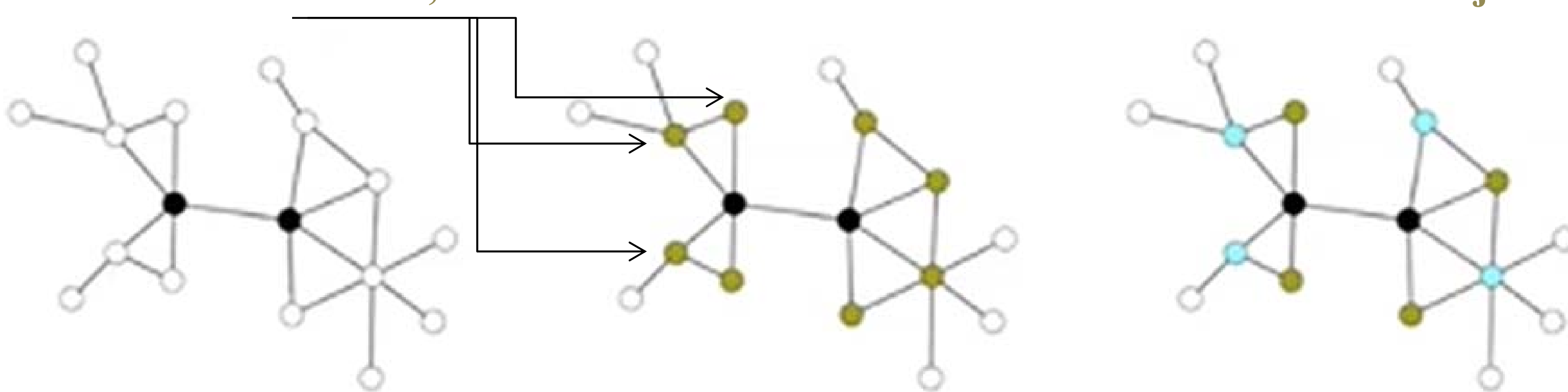
https://web.eecs.umich.edu/~dkoutra/tut/sdm14_part1a.pdf

Defn of regular Equivalence

Defn. Two nodes are regularly equivalent if they are equally related to equivalent nodes (i.e. their neighbors need to be equivalent as well).

We capture equivalence through color pattern adjacencies.

To have similar roles, these nodes need different colors because of their adjacencies

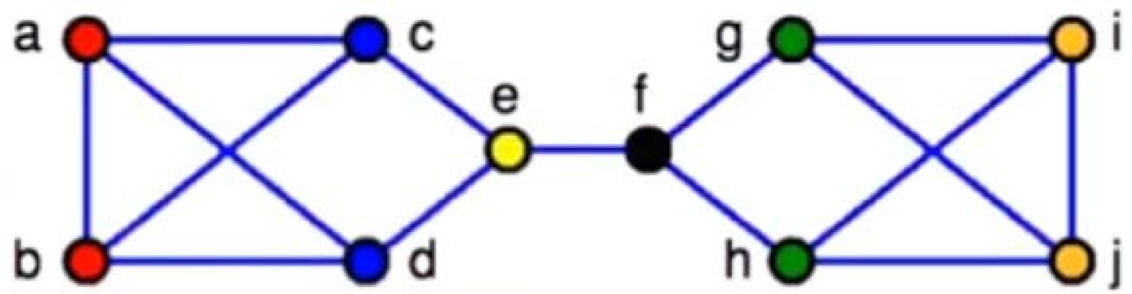


White and Reitz, 1983; Everette and Borgatti, 1991

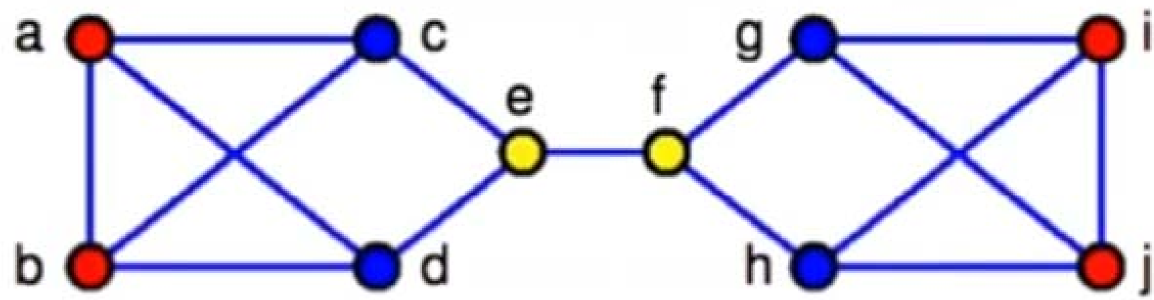
The Borgatti-Everett Network

Structural: Nodes adjacent to a set of nodes (substitutable based on the access to info) vs.
Regular: Nodes adjacent to similar roles in the network (similar positions in the network).

- structural equivalence



- regular equivalence



http://www.stats.ox.ac.uk/~snijders/Equivalences.pdf

Algorithm for Regular Equivalence

NAVAL POSTGRADUATE SCHOOL

Regular Equivalence (1)

- Regular equivalences: nodes don't have to be adjacent to be regular equivalent
- REGE and CUTREGE – original algorithms developed by Borgatti and Everett to discover regular equivalences. Newman says "but the operations are very involved and it is not easy to interpret"
- We'll explore newer methods: develop σ_{ij} that is high if the neighbors k and l of i and j are (regularly) equivalent

$$\sigma_{ij} = \alpha \sum_{kl} a_{ik} a_{jl} \sigma_{kl}$$

σ_{ij} is the product of the counts of common neighbors & their neighbors' regular similarity σ_{kl}

A recurrence relation in terms of σ, much like eigenvector.

S. P. Borgatti and M. G. Everett, Social Networks 15, 361 (1993).

Regular Equivalence (2)

$$\sigma_{ij} = \alpha \sum_{kl} a_{ik} a_{jl} \sigma_{kl}$$

Is equivalent to

$$\sigma = \alpha A \sigma A ,$$

Where, σ_{kl} is the regular similarity of k & l, α is the leading eigenvalue of A, and σ is the eigenvector.

But:

- It may not give high self-similarity σ_{ii} as it heavily depends on the similarity of k & l, the neighbors of i
- It doesn't give high similarity to pairs of vertices with lots of common neighbors (because it looks at values of k & l, including $k \neq l$)

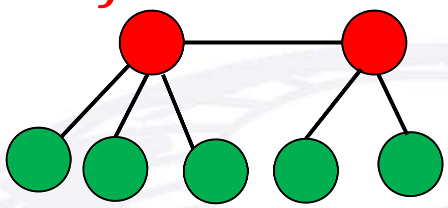
Needs to be altered, and we'll see two methods next.

Regular Equivalence (3)

MEHTOD 1: $\sigma_{ij} = \alpha \sum_{kl} a_{ik} a_{jl} \sigma_{kl} + \delta_{ij}$

Is equivalent to

$$\sigma = \alpha A \sigma A + I,$$



with regular similarity $\sigma^{(t=0)} = \vec{0}$, $\delta_{ij} = \begin{cases} 1, \text{if } i = j \\ 0, \text{if } i \neq j. \end{cases}$

- Calculate by starting with the initial values:

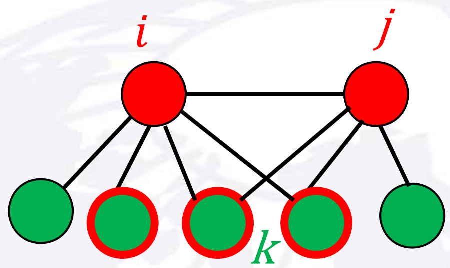
$$\sigma^{(t=1)} = \alpha A \vec{0} A + I = I,$$
$$\sigma^{(t=2)} = \alpha A I A + I = \alpha A^2 + I,$$
$$\sigma^{(t=3)} = \alpha A(\alpha A^2 + I)A + I = \alpha^2 A^4 + \alpha A^2 + I,$$

Note: σ_{ij} is a weighted some of the even powers of A (walks of even length), but we want all the lengths…

Regular Equivalence (4)

METHOD 2: define σ_{ij} to be high if vertex *i* has a neighbor *k* that is similar to *j*

$$\sigma_{ij} = \alpha \sum_k a_{ik}\sigma_{kj} + \delta_{ij}$$



Is equivalent to

$$\sigma = \alpha A\sigma + I, \text{ with } \sigma^{(t=0)} = \vec{0},\ \delta_{ij} = \begin{cases} 1, \text{if } i = j \\ 0, \text{if } i \neq j. \end{cases}$$

➡ $\sigma^{(t=1)} = \alpha A\vec{0} + I = I,$

$\sigma^{(t=2)} = \alpha AI + I = \alpha A + I,$

$\sigma^{(t=3)} = \alpha A(\alpha A + I) + I = \alpha^2 A^2 + \alpha A + I,$

- σ_{ij} is a weighted ($\alpha^?$) count of all walks between *i* and *j*, (with $\alpha < 1$ so longer paths weigh less)

Leicht, Elizabeth A., Petter Holme, and Mark EJ Newman. "Vertex similarity in networks." *Physical Review E* 73.2 (2006): 026120.

Regular Equivalence (5)

METHOD 2: define σ_{ij} to be high if vertex j has a neighbor k that is similar to j

$$\sigma_{ij} = \alpha \sum_k a_{ik}\sigma_{kj} + \delta_{ij}$$

Is equivalent to

$$\sigma = \alpha A\sigma + I, \text{with } \sigma^{(t=0)} = \vec{0}$$

- This looks a lot like Katz centrality (recall $x_i = \alpha \sum_j a_{ij} x_j + \beta$, where β is a constant initial weight given to each vertex so that its out degree matters)

Katz centrality of vertex i is the sum of σ_{ij} , $\forall j \in N(i)$

Regular Equivalence (6)

- The regular equivalence σ_{ij} tends to be high for vertices of high degree (more chances of their neighbors being similar)
- However, some low degree vertices could be similar as well, so the formula can be modified similar to PageRank if desired:

$$\sigma_{ij} = \frac{\alpha}{\deg i} \sum_k a_{ik} \sigma_{kj} + \delta_{ij}$$

which is equivalent to

$$\sigma = \alpha D^{-1} A \sigma + I,$$

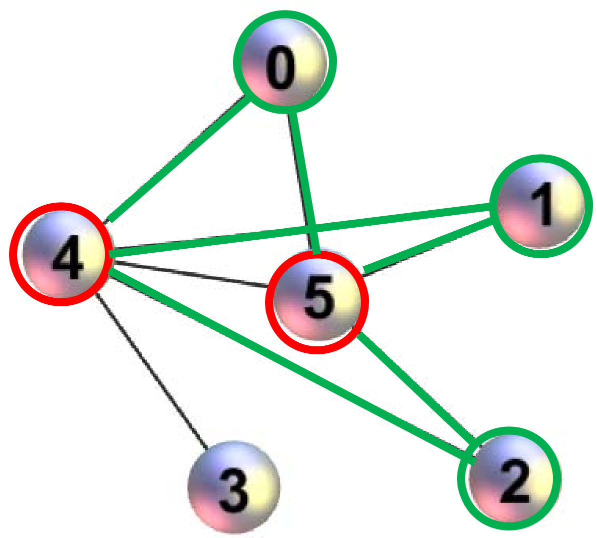
The connection between Regular and Structural Equivalence

Equivalence (1)

- Recall that the regular equivalence σ_{ij} is a weighted count of all walks between *i* and *j*, *i.e*:

$$\sigma^{(t=3)} = \alpha A(\alpha A + I) + I = \alpha^2 A^2 + \alpha A + I,$$

- Recall that the structural equivalence r_{ij} counts the # of common neighbors of *i* and *j* in different ways
- What is the correlation between the number of common neighbors of 4 and 5, and paths of length 2 between 4 and 5?



Equivalence (2)

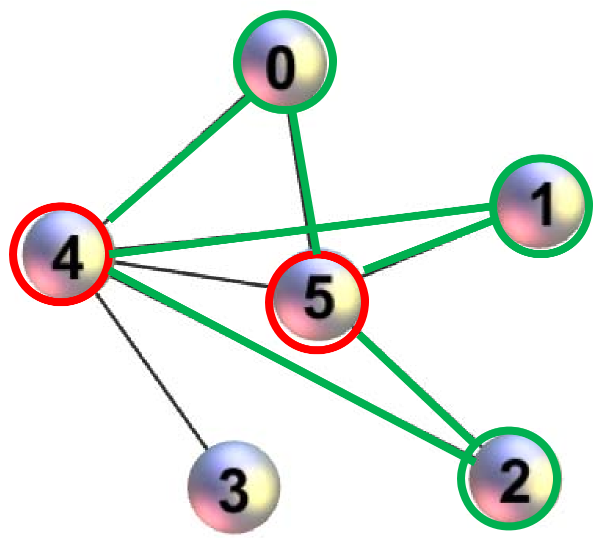
- Recall that the regular equivalence σ_{ij} is a weighted count of all walks between *i* and *j*, *i.e*:

$$\sigma^{(t=3)} = \alpha A(\alpha\, A + I) + I = \alpha^2\, A^2 + \alpha A + I,$$

- Recall that the structural equivalence r_{ij} counts the # of common neighbors of *i* and *j* in different ways

So the regular equivalence σ_{ij} is a generalization of structural equivalence r_{ij}

(r_{ij} counts just paths of length 2)



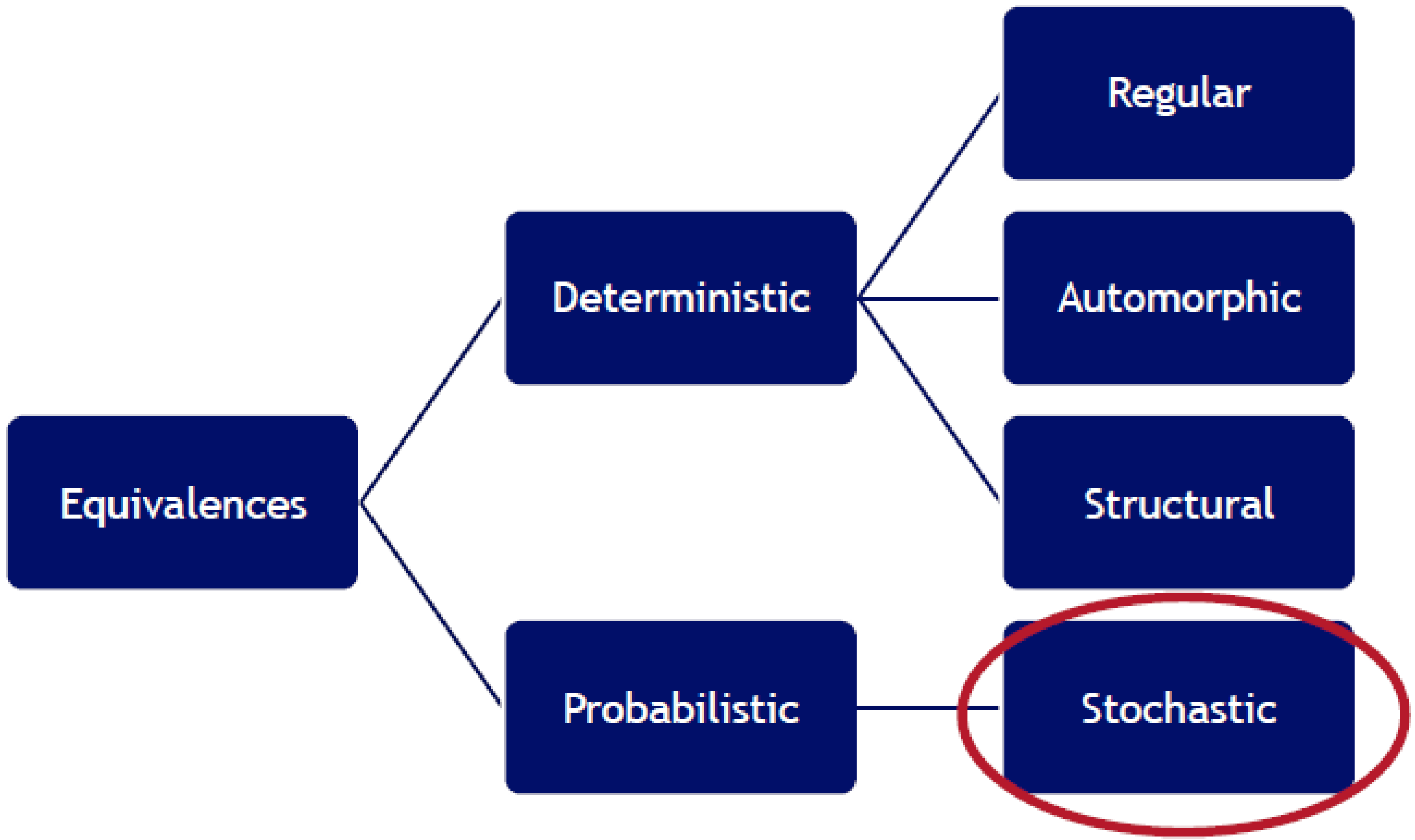
NAVAL POSTGRADUATE SCHOOL

References

- Newman, Mark. *Networks: An introduction*. Oxford university press, 2010.
- Everett, Martin G., and Steve Borgatti. "Role colouring a graph." *Mathematical Social Sciences* 21.2 (1991): 183-188.
- White, Douglas R., and Karl P. Reitz. "Graph and semigroup homomorphisms on networks of relations." *Social Networks* 5.2 (1983): 193-234.
- Leicht, Elizabeth A., Petter Holme, and Mark EJ Newman. "Vertex similarity in networks." *Physical Review E* 73.2 (2006): 026120.
- https://www.cs.utexas.edu/~mooney/
- http://www.stats.ox.ac.uk/~snijders/Equivalences.pdf
- http://faculty.ucr.edu/~hanneman/nettext/C12_Equivalence.html
- https://web.eecs.umich.edu/~dkoutra/tut/sdm14_part1a.pdf

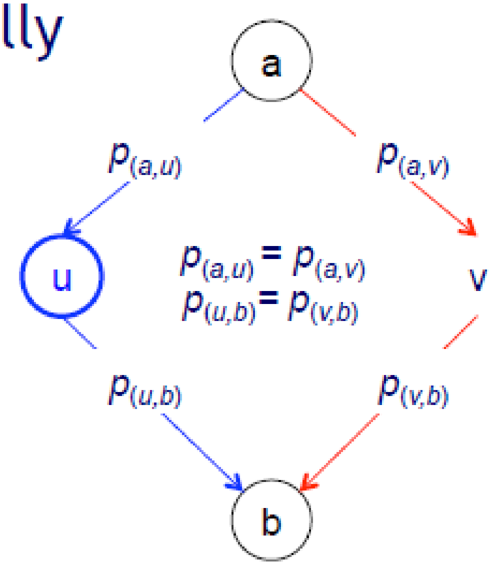
- Backup slides

Equivalences



Stochastic Equivalence

- [Holland, et al. 1983; Wasserman & Anderson, 1987]
- Two nodes are stochastically equivalent if they are "exchangeable" w.r.t. a probability distribution
- Similar to structural equivalence but probabilistic



Stochastic Equivalence: Algorithms

- Many algorithms exist here
- Most recent approaches are generative [Airoldi, et al 2008]
- Some choice points
  - Single [Kemp, et al 2006] vs. mixed-membership [Koutsourelakis & Eliassi-Rad, 2008] equivalences (a.k.a. "positions")
  - Parametric vs. non-parametric models

The stochastic block model is a kind of

Latent Structure Analysis (LSA).

The basic idea of LSA, proposed by Lazarsfeld & Henry (1968), is that there exist latent (i.e. unobserved) variables such that the observations are *conditionally independent* given the latent structure (= latent variables).

The *structural model* then specifies the latent variables and the *measurement model* specifies how the observations depend on the latent variables.