Università di Pisa

Data Mining and Machine Learning

Bioinspired computational methods

Biological data mining

Clustering Graphs and Network Data

Francesco Marcelloni

Department of Information Engineering
University of Pisa
ITALY

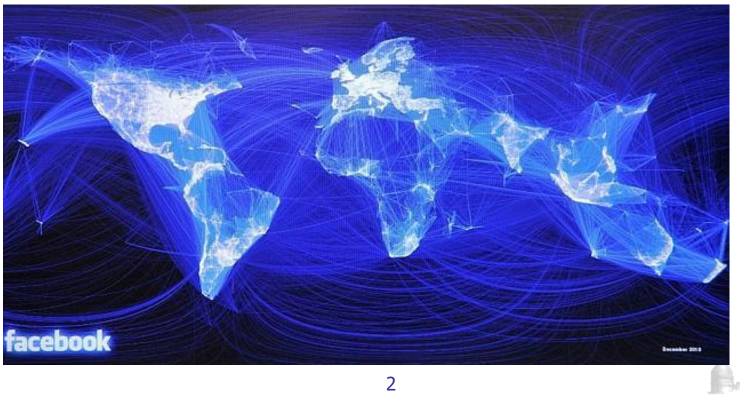
Some slides belong to the collection

Jiawei Han, Micheline Kamber, and Jian Pei
University of Illinois at Urbana-Champaign
Simon Fraser University

1

What is a network?

- Network: a collection of entities that are interconnected with links
  - Social networks
    - Entities: People
    - Links: Friendships

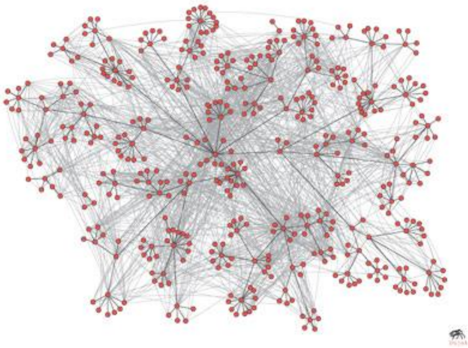


2

Università di Pisa Francesco Marcelloni

What is a network?

- Communication networks
  - Entities: People
  - Links: e-mail exchange



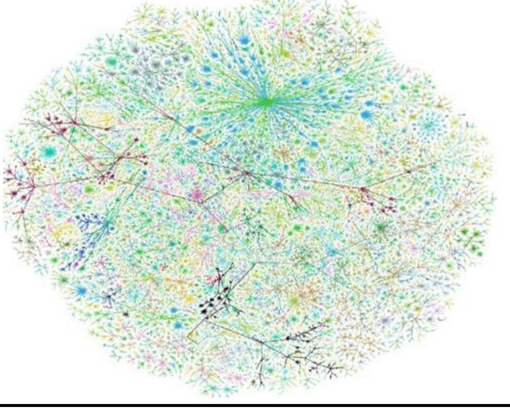
3

3

Università di Pisa Francesco Marcelloni

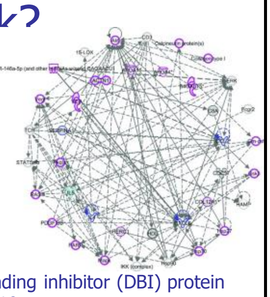
What is a network?

- Communication networks
  - Entities: Internet nodes
  - Links: Communication between nodes



4

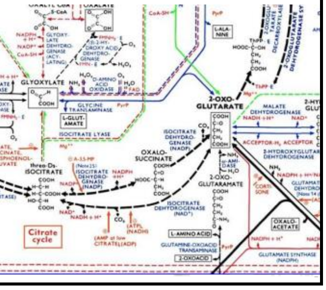
What is a network?



- Biological networks
  - Entities: Proteins
  - Links: Interactions

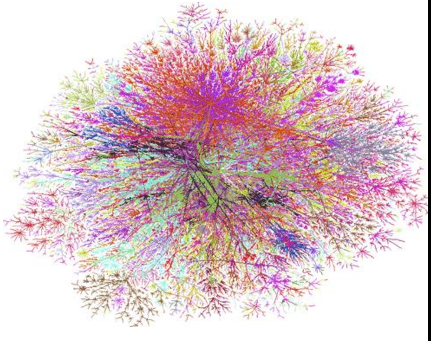
Phenotypic subgrouping and multi-omics analyses reveal reduced diazepam-binding inhibitor (DBI) protein levels in autism spectrum disorder with severe language impairment, March 2019
PLoS ONE 14(3):e0214198

- Entities: metabolites, enzymes
- Links: chemical reactions



What is a network?

- Information/Media networks
  - Entities: Web pages
  - Links: Links



- Entities: Twitter users
- Links: Follows/conversations

- Many other examples

Why networks are imporant?

- We cannot truly understand a complex system unless we understand the underlying network.
  - Everything is connected, studying individual entities gives only a partial view of a system
- Two main themes:
  - What are the structural properties of the network?
  - How do processes happen in the network?

Graphs and Networks

- In mathematics, networks are called graphs, the entities are nodes, and the links are edges
- Graph theory starts in the 18th century, with Leonhard Euler
- Graphs have been used in the past to model existing networks (e.g., networks of highways, social networks)
  - usually these networks were small
  - visual inspection can reveal a lot of information

Networks now

- More and larger networks appear
  - Products of technology
    - e.g., Internet, Web, Facebook, Twitter
  - Result of our ability to collect more, better, and more complex data
    - e.g., gene regulatory networks
  - Result of the willingness of users to contribute data
    - e.g., users making their relationships public online
- Networks of thousands, millions, or billions of nodes
  - Impossible to process visually
  - Problems become harder
  - Processes are more complex

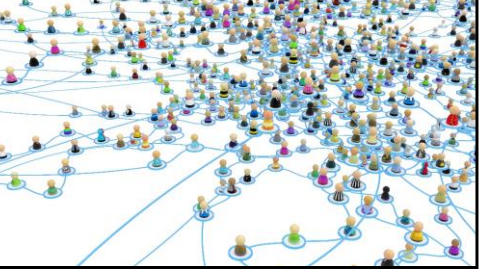
Current problems

- **Ranking of nodes on the web?**
  - Is my home page as important as the Google page?
  - We need algorithms to compute the importance of nodes in a graph
  - For instance, the PageRank algorithm in Google

  - Theoretically, it is impossible to develop a web search engine without understanding the web graph

Current problems

- **Information/Virus Cascade?**
  - How do viruses spread between individuals? How can we stop them?
  - How does information propagate in social and information networks? What items become viral? Who are the influencers and trend-setters?
  - We need models and algorithms to answer these questions



Current problems

- **Link Prediction**
  - Given a snapshot of a social network at time t, we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time t'.
  - Applications
    - Accelerate the growth of a social network (e.g., Facebook, LinkedIn, Twitter) that would otherwise take longer to form.
    - Identify suspect relationships

Current problems

- **Network content**
  - Users on online social networks generate content.
  - Mining the content in conjunction with the network can be useful
    - Do friends post similar content on Facebook?
    - Can we understand a user's interests by looking at those of their friends?
    - Social recommendations: Can we predict a movie rating using the social network?

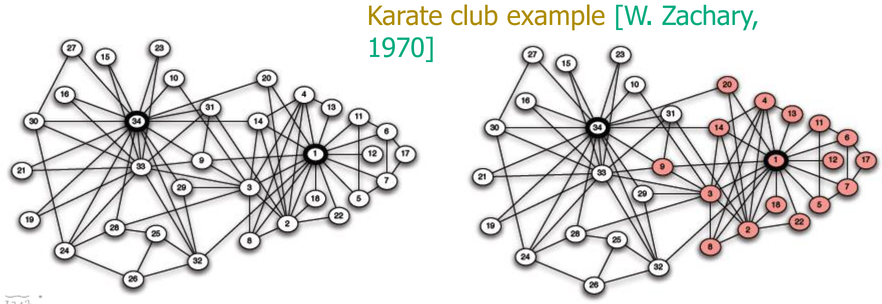
Current problems

- **Social Media**
  - Today Social Media (Twitter, Facebook, Instagram) have supplanted the traditional media sources
    - Information is generated and disseminated mostly online by users
      - E.g., the assassination of Bin Laden appeared first on Twitter
    - Twitter has become a global "sensor" detecting and reporting everything
  - Interesting problems:
    - Automatically detect events using Twitter
      - Earthquake news propagation
      - Crisis detection and management
    - Sentiment mining
    - Track the evolution of events: socially, geographically, over time.

Current problems

- **Clustering and Finding Communities?**
  - A community: "Cohesive subgroups are subsets of actors among whom there are relatively strong, direct, intense, frequent, or positive ties." [Wasserman & Faust '97]

Karate club example [W. Zachary, 1970]

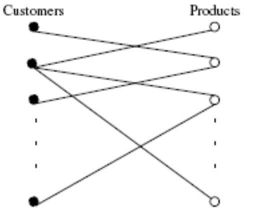


Current problems

- **Community Evolution**
  - Homophily:"Birds of a feather flock together"
  - Caused by two related social forces [Friedkin98, Lazarsfeld54]
    - Social influence: People become similar to those they interact with
    - Selection: People seek out similar people to interact with
  - Both processes contribute to homophily, but
    - Social influence leads to community-wide homogeneity
    - Selection leads to fragmentation of the community
  - Applications in online marketing
    - viral marketing relies upon social influence affecting behavior
    - recommender systems predict behavior based on similarity

Clustering Graphs and Network Data

- Applications
  - Bi-partite graphs, e.g., customers and products, authors and conferences
    - Clustering customers buying similar products
    - Identify customers out of the clusters



Clustering Graphs and Network Data

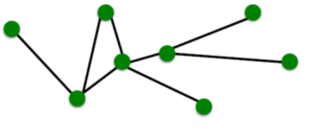
- Applications
  - Web search engines, e.g., click through graphs and Web graphs
    - Click-through information
      - An edge links a query to a web page if a user clicks the web page when asking the query.
      - Valuable information can be obtained by cluster analyses on the query–web page bipartite graph.
    - web graph: each web page is a vertex, and each hyperlink is an edge pointing from a source page to a destination page.

Clustering Graphs and Network Data

- Applications
  - Social networks, friendship/coauthor graphs
    - the vertices are individuals or organizations, and the links are interdependencies between the vertices, representing friendship, common interests, or collaborative activities
    - For instance, customers of a company form a social network, where each customer is a vertex, and an edge links two customers if they know each other.
    - Customers within a cluster may influence one another regarding purchase decision making.
    - As another example, the authors of scientific publications form a social network.
    - The **network** is, in general, a **weighted graph** because an edge between two authors can carry a weight representing the strength of the collaboration such as how many publications the two authors (as the end vertices) coauthored.

Basics on a Network

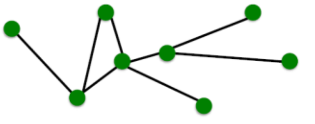
- Objects: nodes, vertices **N**
- Interactions: links, edges **E**
- System: network, graph **G(N,E)**



- **Network often refers to real systems**
  - Web, Social network, Metabolic network
  - **Language: Network, node, link**
- **Graph is a mathematical representation of a network**
  - Web graph, Social graph, Knowledge Graph
  - **Language: Graph, vertex, edge**

Basics on a Network

- How to build a graph:
  - What are nodes?
  - What are edges?



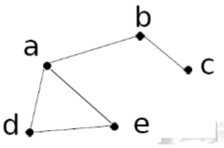
- **Choice of the proper network representation of a given domain/problem determines our ability to use networks successfully:**
  - In some cases there is a unique, unambiguous representation
  - In other cases, the representation is by no means unique
  - The way you assign links will determine the nature of the question you can study

Clustering Graphs and Network Data

- We can apply standard clustering algorithms by introducing a specific definition of similarity measures
  - Geodesic distances
  - Distance based on random walk (SimRank)
- Graph clustering methods
  - Minimum cuts: FastModularity (Clauset, Newman & Moore, 2004)
  - Density-based clustering: SCAN (Xu et al., KDD'2007)

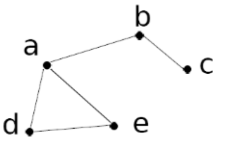
Similarity Measure (I) Geodesic Distance

- Distance between two vertices in a graph: the shortest path between the vertices:
  - Geodesic distance (A, B): length (i.e., # of edges) of the shortest path between A and B (if not connected, defined as infinite)
- Eccentricity of v, eccen(v): The largest geodesic distance between v and any other vertex $u \in V - \{v\}$.
  E.g., eccen(a) = eccen(b) = 2; eccen(c) = eccen(d) = eccen(e) = 3
- Radius of graph G: The minimum eccentricity of all vertices, i.e., the distance between the "most central point" and the "farthest border"
  $r = \min_{v \in V} \text{eccen}(v)$
  E.g., radius (g) = 2



Similarity Measure (II) Geodesic Distance

- Diameter of graph G: The maximum eccentricity of all vertices, i.e., the largest distance between any pair of vertices in G
  $d = \max_{v \in V} \text{eccen}(v)$
  E.g., diameter (g) = 3
- A peripheral vertex is a vertex that achieves the diameter.
  E.g., Vertices c, d, and e are peripheral vertices



Similarity Measure (III) Geodesic Distance

- Let us consider the similarity between two vertices in a customer social network.
- How well can geodesic distance measure similarity and closeness in a network?
  - Suppose that Ada and Bob are two customers in the network
  - The geodesic distance (i.e., the length of the shortest path between Ada and Bob) is the shortest path that a message can be passed from Ada to Bob and vice versa.
  - Is this information useful?
    - Typically, the company is not interested in how a message is passed from Ada to Bob.
  - We need to define what does similarity mean in a social network

Similarity Measure (IV) Similarity in a social network

Two different meanings

- Structural context-based similarity
  - Two customers are considered similar to one another if they have similar neighbors in the social network.
  - two people receiving recommendations from a good number of common friends often make similar decisions: intuitive!
- Similarity based on random walk
  - the company sends promotional information to both Ada and Bob in the social network.
  - Ada and Bob may randomly forward such information to their friends (or neighbors) in the network.
  - The closeness between Ada and Bob can then be measured by the likelihood that other customers simultaneously receive the promotional information that was originally sent to Ada and Bob.

SimRank: Similarity Based on Random Walk and Structural Context

- SimRank: structural-context similarity, i.e., based on the similarity of its neighbors

- In a directed graph G = (V,E),
  - individual in-neighborhood of v: $I(v) = \{u \mid (u, v) \in E\}$
  - individual out-neighborhood of v: $O(v) = \{w \mid (v, w) \in E\}$

- Similarity in SimRank:

$$s(u, v) = \frac{C}{|I(u)||I(v)|} \sum_{x \in I(u)} \sum_{y \in I(v)} s(x, y)$$

  where C is a constant between 0 and 1.
- If a vertex does not have any neighbor, we define s(u,v) = 0

SimRank: Similarity Based on Random Walk and Structural Context

- How can compute SimRank?
  - Iteratively compute the previous equation until a fixed point is reached.

  - Let n be the number of nodes in graph G.
  - For each iteration i we can keep n^2 entries $s_i(*,*)$, where $s_i(u,v)$ gives the score between u and v on iteration i.
  - We start with $s_0(*,*)$ where each $s_0(u,v)$ is a lower bound on the actual SimRank score $s(u,v)$:

$$s_0(u, v) = \begin{cases} 0 & \text{if } u \neq v \\ 1 & \text{if } u = v. \end{cases}$$

SimRank: Similarity Based on Random Walk and Structural Context

- To compute $s_{i+1}(u,v)$ from $s_i(*,*)$ we use

$$s_{i+1}(u,v) = \frac{C}{|I(u)||I(v)|} \sum_{x \in I(u)} \sum_{y \in I(v)} s_i(x,y) \quad \text{if } u \neq v$$

$$s_{i+1}(u,v) = 1 \quad \text{if } u = v.$$

The values $s_i(*,*)$ are non-decreasing as i increases.

Complexity: $O(Kn^2d_2)$ where d_2 is the average of $|I(u)||I(v)|$

K is the number of iterations and typically is equal to 5

SimRank: Similarity Based on Random Walk and Structural Context

- Similarity based on random walk: in a strongly connected graph a path exists between every two nodes).
- Expected distance from u to v:

$$d(u,v) = \sum_{t: u \rightsquigarrow v} P[t]\, l(t)$$

- The sum is computed over all tours t which start at u and end at v, and do not touch v except at the end.
- For a tour $t = \langle w_1, \ldots, w_k \rangle$ the length $l(t)$ of t is k-1.
- The probability $P(t)$ of travelling t is

$$P[t] = \begin{cases} \prod_{i=1}^{k-1} \frac{1}{\boxed{|O(w_i)|}} & \text{if } l(t) > 0 \\ 0 & \text{if } l(t) = 0 \end{cases}$$

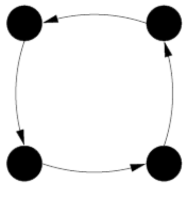
Out-neighbors of w_i

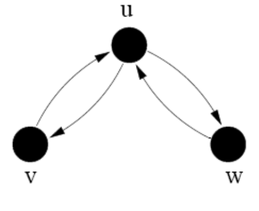
SimRank: Similarity Based on Random Walk and Structural Context

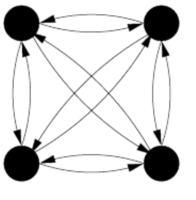
- Note that the case where $u = v$, for which $d(u,v) = 0$ is a special case of the formula of the distance: only one tour is in the summation and it has length 0.
- The expected distance from u to v is exactly the expected number of steps a random surfer, who at each step follows a random out-edge, would take before he first reaches v, starting from u.
- Expected meeting distance (EMD): the expected meeting distance $m(u,v)$ between u and v is the expected number of steps required before two surfers, one starting at u and the other at v, would meet if they walked (randomly) in lock-step.
- The EMD is symmetric by definition

SimRank: Similarity Based on Random Walk and Structural Context

- Some examples of EMD







EMD between two distinct nodes is infinite

$m(u,v) = m(u,w) = \infty$
$m(v,w) = 1$

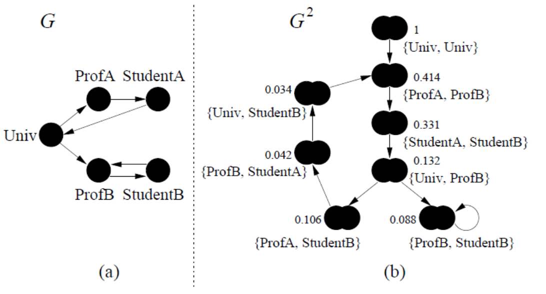
$EMD = 3$

SimRank: Similarity Based on Random Walk and Structural Context

- To define EMD formally in G, we use the derived graph G^2 of node-pairs.
  - Each node (u, v) of V^2 can be thought of as the present state of a pair of surfers in V, where an edge from (u, v) to (c, d) in G^2 says that in the original graph G, one surfer can move from u to c while the other moves from v to d.
  - A tour in G^2 of length n represents a pair of tours in G also having length n.

SimRank: Similarity Based on Random Walk and Structural Context

- G^2 represents an ordered pair of nodes of G. A node (a,b) of G^2 points to a node (c,d) if, in G, a points to c and b points to d. The example represents the Web pages of two professors ProfA and ProfB, their students StudentA and StudentB, and the home page of their university Univ



SimRank: Similarity Based on Random Walk and Structural Context

- Formally, the EMD $m(u, v)$ is simply the expected distance in G^2 from (u, v) to any singleton node $(x, x) \in V^2$, since singleton nodes in G^2 represent states where both surfers are at the same node. More precisely,

$$m(u,v) = \sum_{t:(u,v) \rightsquigarrow (x,x)} P[t]\, l(t)$$

- The sum is taken over all tours t starting from (u,v) which touch a singleton node at the end and only at the end.
- Unfortunately, G^2 may not always be strongly connected (even if G is), and in such cases there may be no tours t for (u,v) in the summation. In this case, $m(u,v) = \infty$.
  - this definition would cause problems in defining distances for nodes from which some tours lead to singleton nodes while others lead to (u, v).

SimRank: Similarity Based on Random Walk and Structural Context

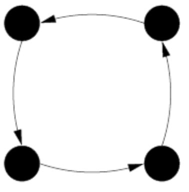
- Solution: Expected-f Meeting distance
  - Map all distances to a finite interval: instead of computing expected length l(t) of a tour, we can compute the expected f(l(t)), for a nonnegative, monotonic function which is bounded on the domain $[0,\infty)$.

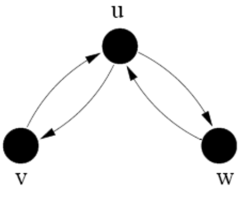
$$s'(u,v) = \sum_{t:(u,v) \rightsquigarrow (x,x)} P[t]\, C^{l(t)} \qquad C \in (0,1)$$

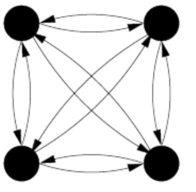
  - Close nodes have a lower score (meeting distances of 0 go to 1 and distances of ∞ go to 0), matching our intuition of similarity.
    - s′(a,b)=0 -> No tour from (a,b) to any singleton nodes
    - s′(a,b)=1 -> a=b
    - s′(a,b)$\in$[0,1] -> for all a,b

SimRank: Similarity Based on Random Walk and Structural Context

- Some examples of expected-f meeting distance with C=0.8.







s′(a,b) = 0

$s'(u,v) = s'(u,w) = 0$
$s'(v,w) = 0.8$

$s'(a,b) = 0.47$

SimRank: Similarity Based on Random Walk and Structural Context

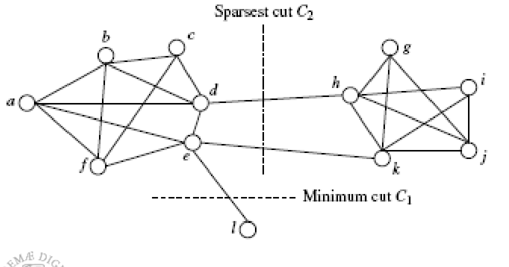
- It has been proved that the SimRank score, with parameter C, between two nodes is their expected-f meeting distance traveling back-edges, for $f(z) = C^z$
- In other words, s(u,v) = s′(u,v) for any two vertices u and v. That is, SimRank is based on both structural context and random walk.

Graph Clustering: Sparsest Cut

- How should we conduct clustering in a graph?
  - Intuitively, we should cut the graph into pieces, each piece being a cluster, such that the vertices within a cluster are well connected and the vertices in different clusters are connected in a much weaker way.
- Let G = (V,E) be a direct graph.
  - A cut C(S,T) is a partitioning of the set of vertices V in G, that is, V= $S \cup T$ and $S \cap T = \emptyset$.
  - The cut set of a cut is the set of edges {(u, v) ∈ E | u ∈ S, v ∈ T}
  - Size of the cut: number of edges in the cut set. If the edges are weighted, the value of the cut is the sum of weights.

Graph Clustering: Sparsest Cut

- What kinds of cuts are good for deriving clusters in graphs?
  - Minimum cut: cut's size is not greater than any other cut's size.
    - Polynomial time algorithms to compute minimum cuts of graphs (Edmonds-Karp algorithm)



Cut C_2 = ({*a*, *b*, *c*, *d*, *e*, *f* , *l* }, {*g*, *h*, *i*, *j*, *k*}) leads to a much better clustering than C_1. The edges in the cut set of C_2 are those connecting the two "natural clusters" in the graph. Specifically, for edges (*d*,*h*) and (*e*,*k*) that are in the cut set, most of the edges connecting *d*, *h*, *e*, and *k* belong to one cluster.

Graph Clustering: Sparsest Cut

- A better measure: Sparsity
- Intuition: choose a cut where, for each vertex *u* that is involved in an edge in the cut set, most of the edges connecting to *u* belong to one cluster.
- The sparsity of a cut C = (S,T) is defined as:

$$\Phi = \frac{\text{cut size}}{\min\{|S|, \boxed{|T|}\}}$$

Number of vertices

- A cut is sparsest if its sparsity is not greater than that of any other cut.
  - Favors solutions that are both sparse (few edges crossing the cut) and balanced (close to a bisection).
  - The problem is known to be NP-Hard, and the best known algorithm is an $O(\sqrt{\log n})$ approximation due to Arora, Rao & Vazirani (2009)
- Ex. Cut C_2 = ({a, b, c, d, e, f, l}, {g, h, i, j, k}) is the sparsest cut

Graph Clustering: Sparsest Cut

- For k clusters, the modularity of a clustering assesses the quality of the clustering:

$$Q = \sum_{i=1}^{k}\left(\boxed{\frac{l_i}{|E|}} - \boxed{\left(\frac{d_i}{2|E|}\right)^2}\right)$$

probability edge is in cluster i

probability a random edge would fall into cluster i

l_i: number of edges between vertices in the i-th cluster

d_i: the sum of the degrees of the vertices in the i-th cluster

where degree of a vertex u: number of edges connecting to u

- The modularity of a clustering of a graph is the difference between the fraction of all edges that fall into individual clusters and the fraction that would do so if the graph vertices were randomly connected

- The optimal clustering of graphs maximizes the modularity

Graph Clustering: Challenges of Finding Good Cuts

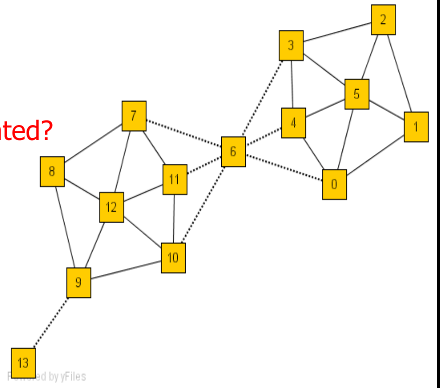
- High computational cost
  - Many graph cut problems are computationally expensive
  - The sparsest cut problem is NP-hard
  - Need to tradeoff between efficiency/scalability and quality
- Sophisticated graphs
  - May involve weights and/or cycles.
- High dimensionality
  - A graph can have many vertices. In a similarity matrix, a vertex is represented as a vector (a row in the matrix) whose dimensionality is the number of vertices in the graph
- Sparsity
  - A large graph is often sparse, meaning each vertex on average connects to only a small number of other vertices
  - A similarity matrix from a large sparse graph can also be sparse

Graph Clustering: Methods

- There exist two kinds of methods
  - Clustering methods for high-dimensional data
  - Clustering methods designed specifically for clustering graphs
- Clustering methods for high-dimensional data
  - Extract a similarity matrix from a graph using a similarity measure
  - A clustering algorithm for high-dimensional data is therefore applied
- Clustering methods designed specifically for clustering graphs
  - Exploit the peculiarities of the graph for performing the clustering process

SCAN: Density-based clustering of Networks

- How many clusters?
- What size should they be?
- What is the best partitioning?
- Should some points be segregated?



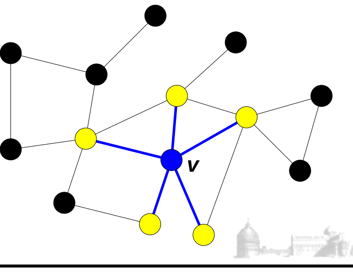
- Application: Given simply information of who associates with whom, could one identify clusters of individuals with common interests or special relationships (families, cliques, terrorist cells)?

Università di Pisa
Francesco Marcelloni

SCAN: Density-based clustering of Networks

- Cliques, hubs and outliers
- Individuals in a tight social group, or clique, know many of the same people, regardless of the size of the group
- Individuals who are hubs know many people in different groups but belong to no single group. Politicians, for example bridge multiple groups
- Individuals who are outliers reside at the margins of society. Hermits, for example, know few people and belong to no group
- The Neighborhood of a Vertex

Define $\Gamma(v)$ as the immediate neighborhood of a vertex (i.e. the set of people that an individual knows)

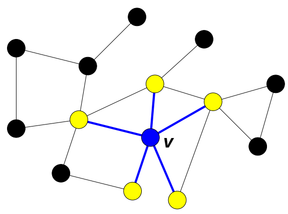


Structure Similarity

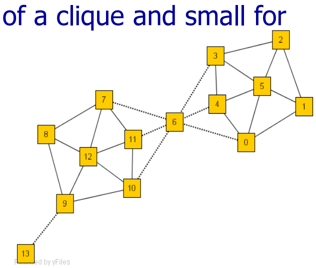
- The desired features tend to be captured by a measure we call Structural Similarity

$$\sigma(v,w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{|\Gamma(v)||\Gamma(w)|}}$$

$$\Gamma(u) = \{v | (u,v) \in E\} \cup \{u\}$$



- Structural similarity is large for members of a clique and small for hubs and outliers



Structural Connectivity

- SCAN uses a similarity threshold ε to define the cluster membership
- For a vertex $v \in V$, the ε-Neighborhood of v is defined as:

$$N_\varepsilon(v) = \{w \in \Gamma(v) | \sigma(v,w) \geq \varepsilon\}$$

- A core vertex is a vertex inside of a cluster. v is a core vertex if and only if:

$$CORE_{\varepsilon,\mu}(v) \Leftrightarrow |N_\varepsilon(v)| \geq \mu$$

where μ is a popularity threshold.

- SCAN grows cluster from core vertices (similar to DBSCAN)
  - If a vertex v is in the ε-Neighborhood of *a core u*, then v is assigned to the same cluster as u
  - The growing process continues until no cluster can be further grown.

Structural Connectivity

- Formally, a vertex *w* can be directly reached from a core *v* if

$$DirRECH_{\varepsilon,\mu}(v,w) \Leftrightarrow CORE_{\varepsilon,\mu}(v) \wedge w \in N_{\varepsilon}(v)$$

- Structure reachable: transitive closure of direct structure reachability. A vertex *v* can be reached from a core vertex *u* if there exist vertices w_1, …, w_n such that w_1 can be reached from *u*, w_i can be reached from w_{i-1}, for $1<i<=n$, and *v* can be reached from w_n.

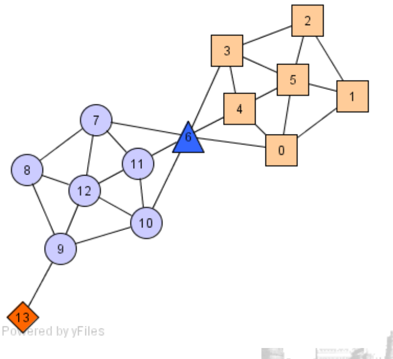
- Structure connected: two vertices v and w, which may or may not be cores, are said connected there exists a core *u such that v and w can be reached from u.*

$$CONNECT_{\varepsilon,\mu}(v,w) \Leftrightarrow \exists u \in V : RECH_{\varepsilon,\mu}(u,v) \wedge RECH_{\varepsilon,\mu}(u,w)$$

M. Ester, H. P. Kriegel, J. Sander, & X. Xu (KDD'96) "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases

Structure-connected clusters

- Structure-connected cluster C
  - Connectivity: $\forall v,w \in C : CONNECT_{\varepsilon,\mu}(v,w)$
  - Maximality: $\forall v,w \in V : v \in C \wedge REACH_{\varepsilon,\mu}(v,w) \Rightarrow w \in C$
- Hubs:
  - Not belong to any cluster
  - Bridge to many clusters
- Outliers:
  - Not belong to any cluster
  - Connect to less clusters



SCAN Algorithm

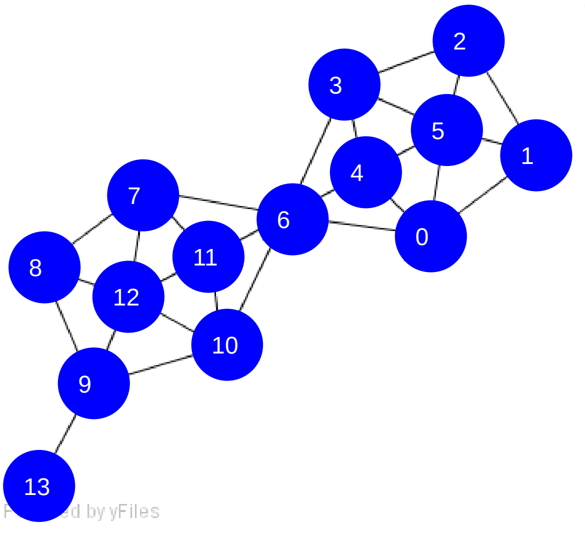
Algorithm: SCAN for clusters on graph data.
Input: a graph $G = (V, E)$, a similarity threshold ε, and a
population threshold μ
Output: a set of clusters
Method: set all vertices in V unlabeled
for all unlabeled vertex u **do**
if u is a core **then**
generate a new cluster-id c
insert all $v \in N_\varepsilon(u)$ into a queue Q
while $Q \neq$ **do**
$w \leftarrow$ the first vertex in Q
$R \leftarrow$ the set of vertices that can be directly reached from w
for all $s \in R$ **do**
if s is not unlabeled or labeled as nonmember **then**
assign the current cluster-id c to s
endif
if s is unlabeled **then**
insert s into queue Q
endif
endfor

SCAN Algorithm

remove w from Q
end while
else
label u as nonmember
endif
endfor
for all vertex u labeled nonmember **do**
if $\exists x, y \in \Gamma(u)$: x and y have different cluster-ids **then**
label u as hub
else
label u as outlier
endif
endfor

Algorithm

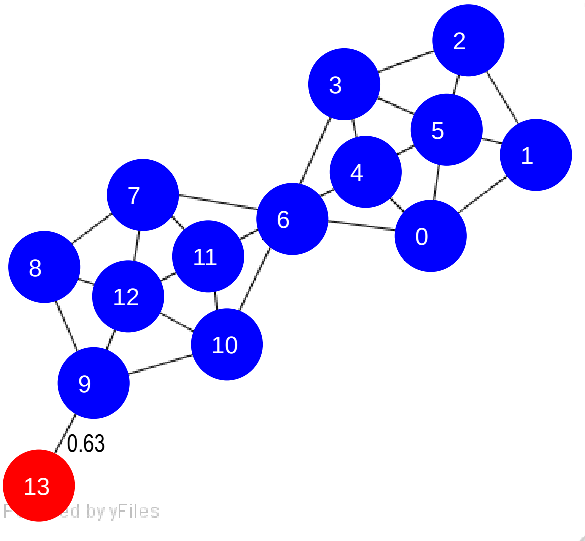
$\mu = 2$
$\varepsilon = 0.7$



53

Algorithm

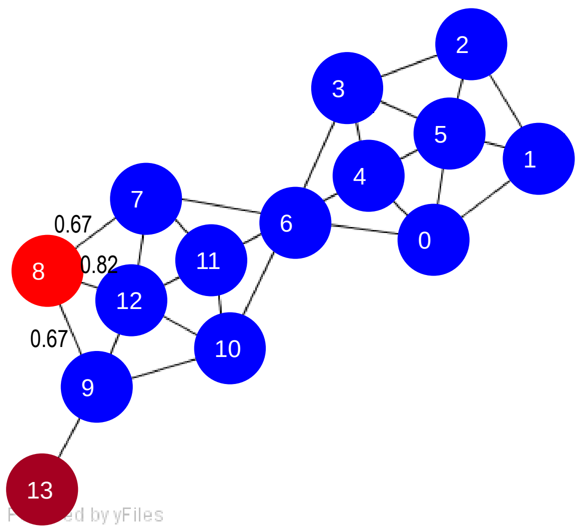
$\mu = 2$
$\varepsilon = 0.7$



54

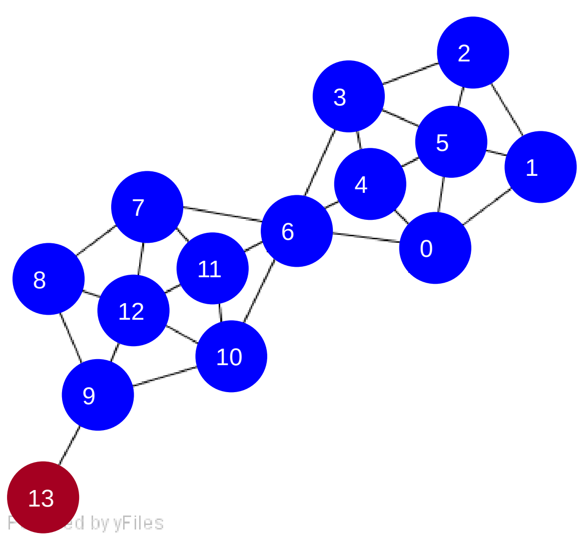
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



Algorithm

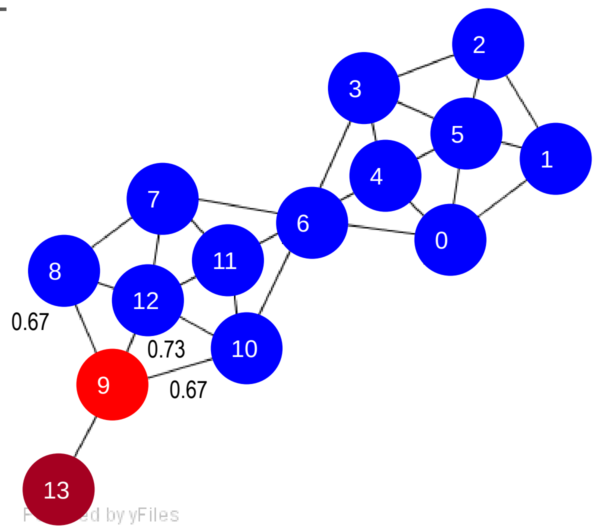
$\mu = 2$
$\varepsilon = 0.7$



Università di Pisa

Algorithm

$\mu = 2$
$\varepsilon = 0.7$

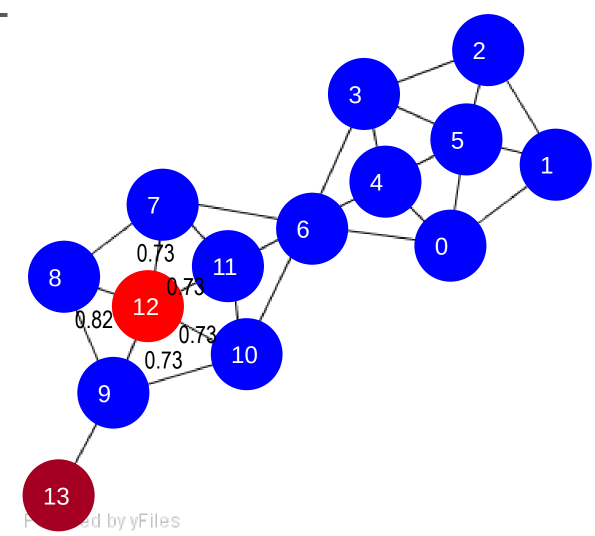


57

Università di Pisa

Algorithm

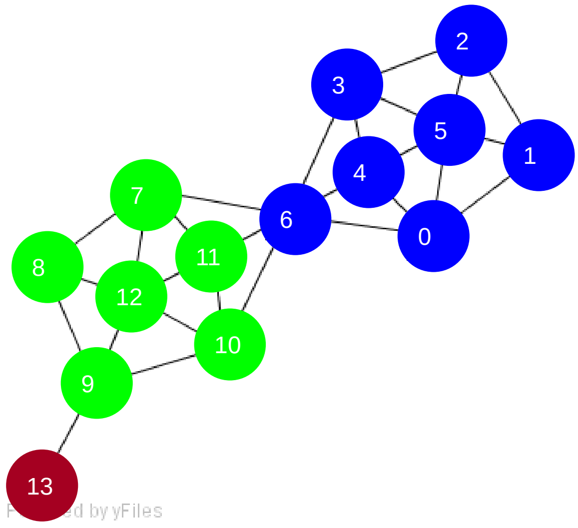
$\mu = 2$
$\varepsilon = 0.7$



58

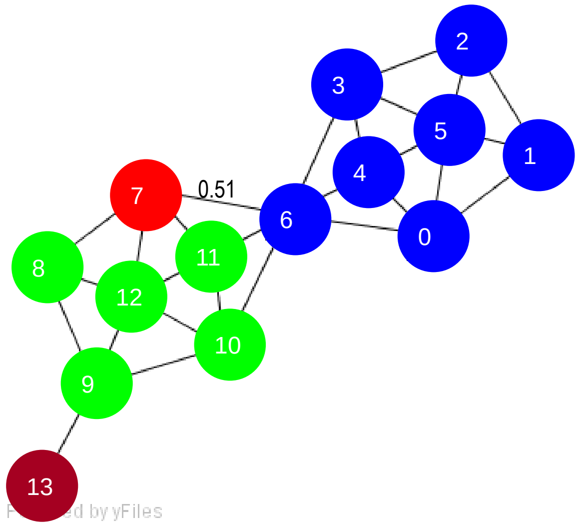
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



Algorithm

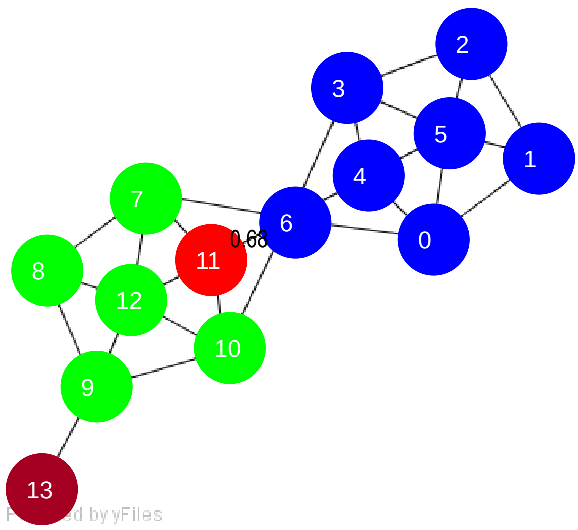
$\mu = 2$
$\varepsilon = 0.7$



Università di Pisa

Algorithm

$\mu = 2$
$\varepsilon = 0.7$

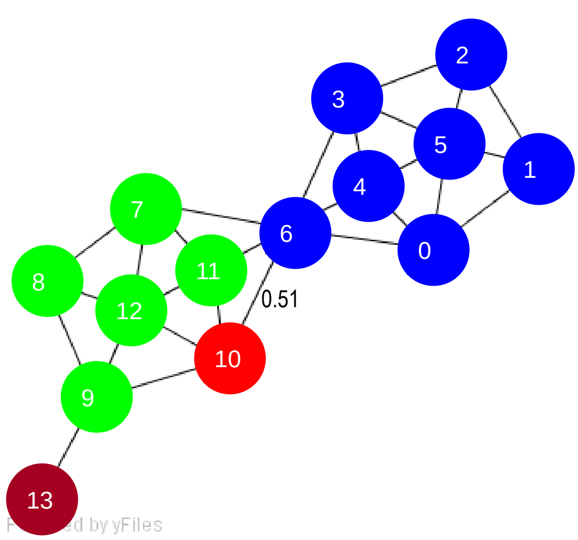


61

Università di Pisa

Algorithm

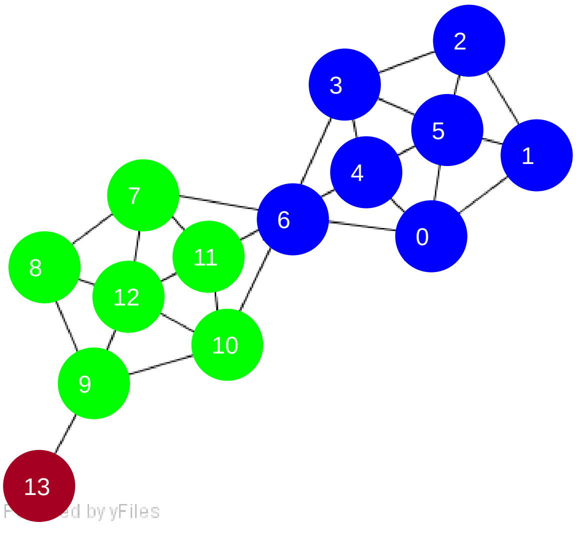
$\mu = 2$
$\varepsilon = 0.7$



62

Algorithm

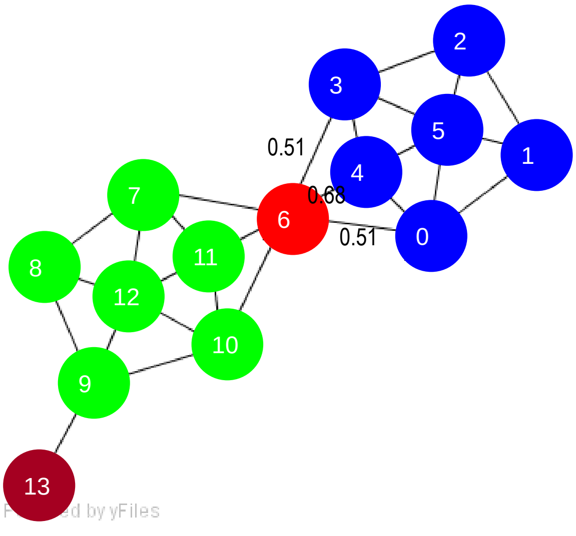
$\mu = 2$
$\varepsilon = 0.7$



63

Algorithm

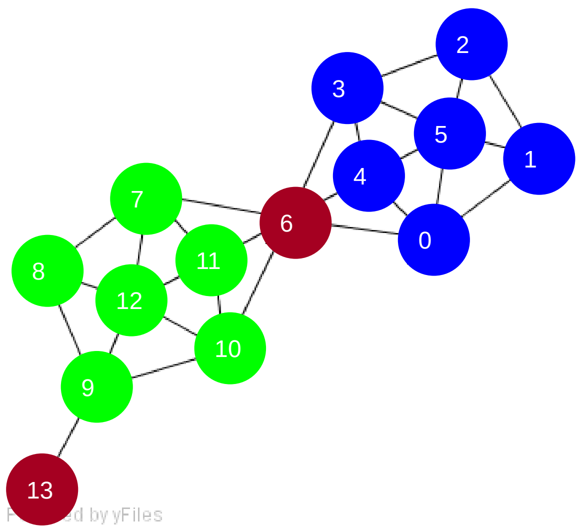
$\mu = 2$
$\varepsilon = 0.7$



64

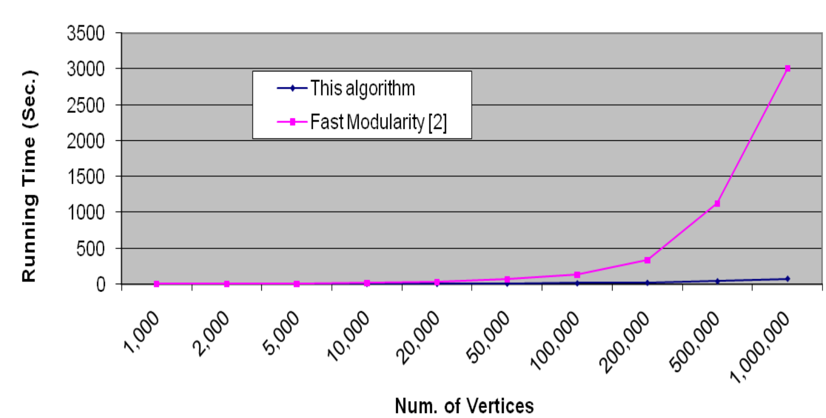
Algorithm

$\mu = 2$
$\varepsilon = 0.7$



Running time

- Running time = O(|E|)
- For sparse networks = O(|V|)



A. Clauset, M. E. J. Newman, & C. Moore, *Phys. Rev. E* **70**, 066111 (2004).