Graph Theory: Summary of Selected Topics

This is a succinct summary of selected graph-related topics, intended to map the main materials in the field and shortly explain what they are.

While the chapters are mostly independent (in particular the *basic graph algorithms* chapter can be easily read first), former familiarity with the very basic graph definitions is recommended for any of the chapters.

The various references used for this summary are stated in the beginning of each chapter.

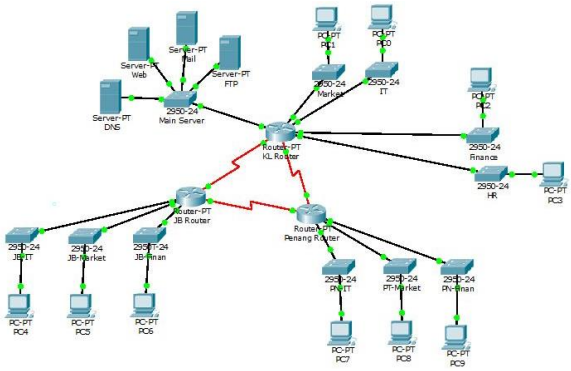
Summarized by Ido Greenberg, 2019.

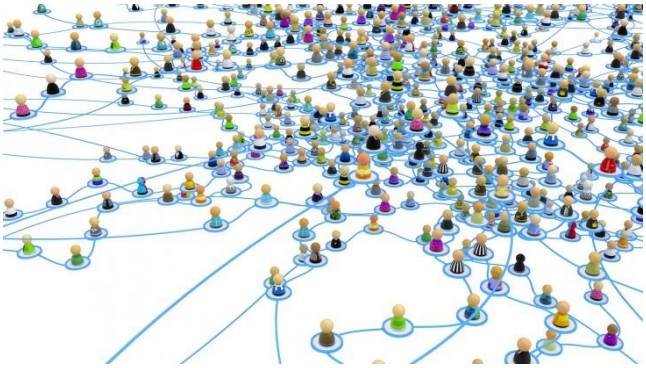
Contents

Applications

Some more and less intuitive problems and fields that can be researched using graph representation:

- Traffic & navigation
- Telecommunication
- Social networks
- Deceases spreading
- Pairs-match (e.g. matchmaking)
- Strategic and military alliances
- Water pipes systems
- Clustering of elements by similarity or closeness
- Estimation of viewing directions $\{R_i\}_i$ of different cameras watching the same object, through a known subset of the relative directions $\{Ri\_vs\_Rj\}_{ij}$





Graph Theory

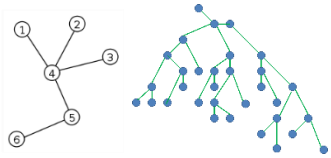
Main source: Arazim project notes of Graph Theory course in TAU.

Basic definitions

- ***Graph***: $G = (V, E)$ (*nodes/vertices* $\{v\}$ & *edges* $\{(v,u)\}$); $n \coloneqq |V|$, $m \coloneqq |E|$.
  - Unless otherwise specified, all graphs in this summary are assumed to be ***simple*** (neither self-edges nor double edges) and ***undirected***.
- ***Complete graph***: $E = \{(u,v) | u \neq v \in V\}$ = all possible edges.
- ***Adjacency matrix***: $A_{ij} \coloneqq 1$ if $(i,j) \in E$ else 0.
- ***Line graph*** (*edge-to-vertex dual*, *conjugate*, *edge graph*) $L(G)$: graph of edges of G as nodes.
- ***Clique***: a set of nodes which is fully connected (i.e. with edges between all nodes).
- ***Independent set***: a set of nodes with no edges between them.
- ***Walk*** (repetitions allowed), ***trail*** (no edge repetitions), ***path*** (no edge/node repetitions).
- ***Circuit*** (no edges repetitions), ***cycle*** (no edge/node repetitions).
- ***Connected graph***: there's a path between any two nodes.
- ***Degree*** of node (d_v): number of neighbors (or edges). Min & max degrees in G are denoted δ, Δ.
- ***Regular graph***: all nodes have the same degree.
- ***Bipartite graph***: $E \subseteq A \times B$ where $V = A \biguplus B$.

Trees

- ***Tree*** – equivalent definitions:
  - Connected graph $G = (V, E)$ with no cycles. (not connected → ***forest***)
  - Connected G with $|V| - 1$ edges.
  - G with $|V| - 1$ edges and no cycles.
  - G with exactly 1 path between each 2 nodes.
- ***Cayley's formula***: the number of different possible trees on n labeled nodes (or equivalently, number of spanning trees in a complete graph) is n^{n-2}. That's a private case of:
- ***Kirchhoff's matrix tree theorem***: the **number of spanning trees in a connected graph is** $\frac{1}{n}\prod_{i=1}^{n-1}\lambda_i$, where λ_i are the sorted eigenvalues of the Laplacian matrix $L_{ii} \coloneqq deg_i$, $L_{ij} \coloneqq -\chi_{(i,j)\in E}$.



Trees

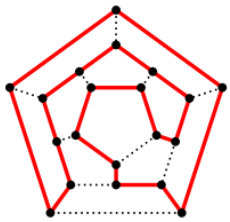
Connectivity

- ***Cut***: all edges between A and B for $V = A \biguplus B$.
- ***Cut vertex***: $v \in V$ such that (V, E) is connected but $(V \backslash \{v\}, E)$ is not.
- ***Bridge***: $e \in E$ such that (V, E) is connected but $(V, E \backslash \{e\})$ is not.
  - E.g. any edge in a tree is a bridge.
- ***Vertex connectivity*** ($\kappa(G)$): minimum number of nodes whose deletion makes G disconnected.
  - ***k-connected*** graph: $\kappa(G) \geq k$.
- ***Edges connectivity*** ($\kappa'(G)$): minimum number of edges whose deletion makes G disconnected.
  - $\boldsymbol{\kappa(G) \leq \kappa'(G) \leq \delta(G)}$.

- ***Block***: 2-connected subgraph which is maximal wrt containment (i.e. doesn't have cut-nodes).
  - Intersection of 2 blocks cannot contain more than 1 node.

Euler and Hamilton cycles

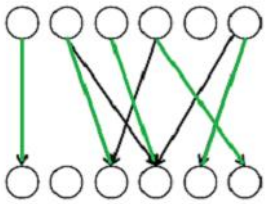
- ***Euler circuit*** (originated in Konigsberg's bridges): pass through all edges exactly once.
  - Exists in a connected graph iff all degrees are even.
- ***Hamilton path/cycle***: pass through all nodes exactly once.
  - Hamilton cycle in the line-graph is (up to pathological examples) Euler circuit in the graph.
  - Decide its existence in a graph is NP-complete.
  - Necessary condition: $\forall \phi \neq S \subset V$: $G(V \backslash S)$'s connectivity components number $\leq |S|$.
  - Sufficient condition (Dirac, 1952): $n \geq 3$ and $\forall v \in V: d_v \geq n/2$.
  - Sufficient condition (Ore, 1960): $n \geq 3$ and $\forall (i,j) \notin E: d_i + d_j \geq n$.



Hamilton cycle

Matchings and covers

- ***Matching*** (*independent edge set*): $M \subset E$ with no shared nodes (i.e. each node appears at most once in M).
  - ***Matching number***: $\mu(G) \coloneqq \max_{\text{M is a matching}} |M|$.
  - M ***saturates*** $A \subset V$ if each $v \in A$ has edge in M.
  - ***Perfect matching***: M saturates all the nodes (i.e. $|M| = |V|/2$).
    - ***Near-perfect***: exactly 1 node is not saturated.
- ***Vertex cover***: $T \subset V$ that covers all edges.
  - $\boldsymbol{T}$ **is a cover** $\Leftrightarrow$ $\boldsymbol{V \backslash T}$ **is an independent set**.
  - ***Cover number***: $\tau(G) \coloneqq \min_{\text{T is a cover}} |T|$.
    - $\forall G$: $\boldsymbol{\mu \leq \tau \leq 2\mu}$.
- In bipartite graphs:
  - $\boldsymbol{\mu(G) = \tau(G)}$.
  - ***Hall's theorem***: a bipartite graph $(A \dot{\cup} B, E)$ has A-saturating matching $\Leftrightarrow$ any $X \subset A$ has more neighbors than elements.
  - Any bipartite regular graph has a perfect matching.

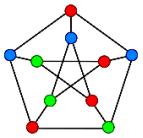


Maximum matching in a bipartite graph

Coloring

Vertex coloring

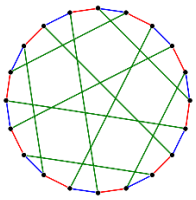
- ***k*-coloring** of a graph: assignment of colors $f: V \to \{1 \dots k\}$, s.t. $\forall (v,u) \in E: f(v) \neq f(u)$.
  - Equivalently, k-coloring is a partition of V to k independent sets.
- ***Chromatic number*** ($\chi(G)$): min k s.t. G is k-colorable.
  - $\chi(G) \geq \frac{|V|}{|\text{max independent set}|}$ (since coloring is partition to ind. sets)
  - $V = A \cup B \;\Rightarrow\; \chi(G) \leq \chi\big(G(A)\big) + \chi\big(G(B)\big)$ (proved by union of disjoint colorings)
  - $\chi(G) + \chi(\bar{G}) \leq |V| + 1$ (where $\bar{G}$ – the ***complement graph*** – includes all edges but E).
- ***Degeneracy*** of graph: min degree in the [subgraph that maximizes it] ($degen(G) \coloneqq \max_{\tilde{G} \subset G} \delta(\tilde{G})$).
  - $\delta(G) \leq degen(G) \leq \Delta(G)$ ($\leq$ by $\tilde{G} \coloneqq G$, $\geq$ trivially)
  - $\boldsymbol{\chi(G) \leq 1 + degen(G)} \leq 1 + \Delta(G)$
  - ***Brooks theorem***: actually $\boldsymbol{\chi(G) \leq \Delta(G)}$ – unless G is complete or an odd cycle.
- ***k*-critical** graph: smallest that's still k-colorable (i.e. $\chi(G) = k \;\wedge\; \forall \tilde{G} \subsetneq G: \chi(\tilde{G}) < k$).
  - G is k-critical $\Rightarrow \delta(G) \geq k - 1$ (otherwise remove v with $d_v \leq k-2$, do (k-1)-coloring, and return v with a color different from its (k-2) neighbors $\Rightarrow G$ is (k-1)-colorable).
    - In particular, $|E| \geq \frac{k-1}{2}|V|$.
  - G is k-critical $\Rightarrow \kappa'(G) \geq k - 1$.
- ***Erdos theorem***: $\forall k, l \in N, \exists G$: [all cycles of G are longer than l] $\wedge$ [$\chi(G) \geq k$].
  - I.e. lack of "local" (short) cycles doesn't guarantee that we can do with just few colors.



Vertex 3-coloring

Edges coloring

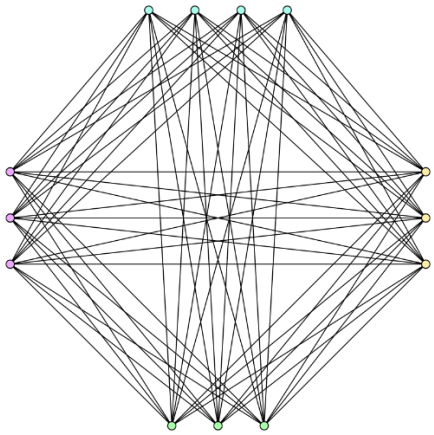
- *Edge* ***k*-coloring**: assignment of colors $f: E \to \{1 \dots k\}$, s.t. $\forall (e, \tilde{e}) \in E: f(e) \neq f(\tilde{e})$.
- ***Chromatic index*** ($\chi'(G)$): min k s.t. G is k-edge-colorable.
- $\chi'(G) = \chi\big(L(G)\big)$ (from definition of the Line-graph)
- $\chi'(G) \geq \Delta(G)$ (the edges of v with $d_v = \Delta$ must have Δ different colors)
- ***Vizing theorem***: $\boldsymbol{\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1}$.
  - G is bipartite $\Rightarrow \chi'(G) = \Delta(G)$.
- ***Ramzi theorem***: $\forall k, l \in N, \exists r_{k,l}, \forall G$: $|\boldsymbol{V}| \geq \boldsymbol{r_{k,l}} \Rightarrow$ **$\exists$either a *k*-size clique or a *l*-size ind. set**.
  - Equivalently, any edges-assignment of 2 colors in a large ($n \geq r_{k,l}$) complete graph, forms a clique of either of the colors.
  - The minimal $r_{k,l}$ is ***Ramzi number*** of k & l, and satisfies $r_{k,l} \leq \binom{k+l-2}{k-1}$.



Edge 3-coloring

Extremal graph theory

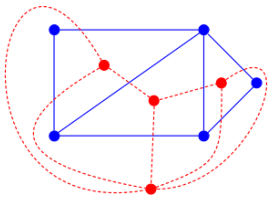
- Extremal graph theory deals with finding the maximal or minimal graph wrt certain constraints. One of the foundations of the theory is Turan's theorem (1941), which claims how many edges a graph can have without containing a clique of certain size.
- ***Complete multipartite graph***: $V = \biguplus_{i=1}^{m} A_i$ and $E = \{(a_i, a_j) \mid a_i \in A_i, a_j \in A_j, i \neq j\}$.
- ***Turan graph*** ($T_{n,m}$): balanced complete multipartite graph (i.e. $|A_i| \in \left\{\left\lfloor \frac{n}{m} \right\rfloor, \left\lceil \frac{n}{m} \right\rceil\right\}$).
  - Note: largest clique in Turan graph is of size m.
- ***Turan's theorem*** (1941): Turan graph has the maximum number of edges among [n-nodes graph with no $(m+1)$-clique].



Turan graph

Planar graphs

- ***Plane graph***: embedding (i.e. drawing) of a graph in a plane.
  - ***Face***: connectivity component of the plane after removing the plane graph.
  - ***Dual graph*** (G^*): graph of faces (faces are connected iff they share an edge in the plane graph).
- ***Planar graph***: can be embedded in a plane with no intersection of edges (up to in the vertices).
  - Planar ⇔ can be embedded in a sphere.
  - ***Euler's formula***: $\boldsymbol{\phi} \coloneqq |\{\mathbf{faces}\}| = |\boldsymbol{E}| - |\boldsymbol{V}| + \mathbf{2}$. (also $e^{i\pi} + 1 = 0$)
- A clique of 5 nodes can be embedded in a torus with no intersections.



A dual graph (in red)

Basic Graph Algorithms

Main source: Prof. Reuven Bar-Yehuda lectures (Technion, 2013) and lecture notes (Technion, 2015).

Search

Most basic graph ***search*** algorithms (AKA ***traversal***, ***exploration***) try to find a path (often the shortest one) from s to t by beginning from s and explore neighbor-nodes. This process goes on recursively, where at each point there're 3 groups of nodes: those fully scanned, those not reached yet, and those reached but not fully scanned yet (the ***frontier*** of the search). The algorithms differ in the order of the exploration, which is determined by the sorting of the frontier.

Algorithm	Weights	Frontier	Running time
BFS (breadth-first)	Ignoring weights	Queue	$O(\|V\|+\|E\|)$
DFS (depth-first)		Stack	
Dijkstra (close-first)	$w \in [0,\infty)$	Heap: distance from s	$O(\|V\|\log\|V\|+\|E\|)$
A* (close-first)		Heap: distance from s + heuristic distance to t	
Bellman-Ford	$w \in R$	All reached nodes (nodes can never be removed from frontier due to possibly negative edges)	$O(\|V\|\cdot\|E\|)$

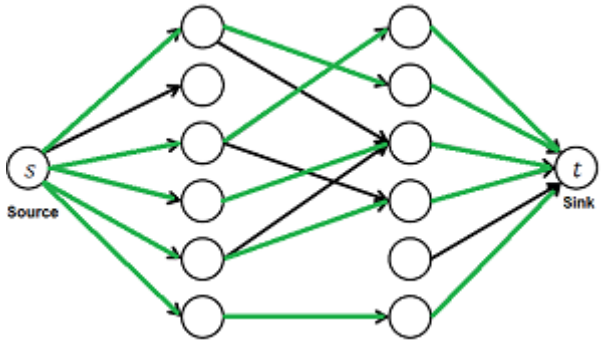
- DFS: the stack frontier means that the next node to explore is a neighbor of the one that was just reached, which intuitively keeps the search "continuous" (no immediate jumps to other nodes in the frontier), which is useful for:
  - Mapping ***connectivity components*** and ***separating nodes***.
  - Physical search (like an actual maze).
  - ***Topological sort*** in directed acyclic graphs (***DAG***): $a < b$ iff there's a path $a \to b$.
- ***Floyd-Warshall algorithm***: use **dynamic programming** to find shortest paths between all pairs of nodes (each step find all shortest paths of length $\leq n$ based on all those of length $\leq n-1$).
  - Can handle negative weights (but not negative cycles).
  - Running time: $O(|V|^3)$.

Minimum spanning tree (MST)

- *Tree*: a connected graph with no cycles.
  - *Spanning tree* of a graph (V,E): a tree (V,E') with $E' \subset E$.
    - *Minimum spanning tree*: spanning tree with minimal sum of weights.
- Fundamental lemma: if $V = X \biguplus Y$, then there exists MST with the shortest edge from X to Y.
- ***Prim's algorithm***: iteratively go over all edges from X to $V \backslash X$, choose the shortest (x,y), and add y to X.
  - Running time: $O(|V|\cdot|E|)$; there's a variant with $O(|V|^2+|E|)$).
- ***Kruskal algorithm***: iteratively add to E' the shortest edge $(x,y) \in E \backslash E'$, and shrink x,y into a single node.
  - Running time: $O(|E|\log|E|)$ if the edges are sorted in advance.

Flow and matching

- ***Max flow problem***: find the strongest flow from s to t within a graph, subject to edges capacities.
- ***Min cut***: find the partition $V = X \sqcup Y$ for which the edges $\{(x, y)\}$ have minimal sum of capacities.
- Equivalence: **[max flow from s to t] $\equiv$ [min cut subject to $s \in X$ and $t \in Y$]**.
- ***Ford-Fulkerson algorithm***: while there's a path $s \to t$, add its maximal flow and update capacities.
  - In the end, the subgraph which is still connected to s defines a min cut.
  - Note: the greediness of the algorithm is not a limitation, since each added path can be practically canceled later by paths which use the same edges in the opposite direction.
  - Note: time complexity strongly depends on implementation.
    - ***Edmonds-Karp*** **($O(|V|^3 \cdot |E|)$)**: choose next path using BFS.
    - ***Dinic*** **($O(|V| \log |V| \cdot |E|)$)**: don't update capacities every single added path.
- Generalizations:
  - Flow with lower bound (edges have both min & max flow constraints):
    - Simply generalizable given a valid flow (where all edges constraints are satisfies).
    - Initial valid flow can be found using tricky construction of fictive new nodes along with shift of the valid-flow-interval of the edges.
  - Minimum flow: just look for max inverted flow $t \to s$.
  - Multiple sources $\{s_i\}$ and targets $\{t_i\}$: add fictive s_0 & t_0 with $c(e_{s_0 s_i}), c(e_{t_i t_0}) \equiv \infty$.
- Applications:
  - ***Maximum matching problem***: given a ***bipartite graph*** (i.e. $E \subseteq X \times Y$ for $V = X \sqcup Y$) of possible matches (possibly weighted) between X and Y, find a maximum match.
    - Can be solved through max flow using the construction $s \to X \to Y \to t$.
    - Hall's condition: an equivalent condition for existence of a full match (i.e. match that fully covers either X or Y).
  - Max ***independent set***: find a maximal set $U \subseteq V$ such that there are no edges in U.
  - Min ***vertex cover***: find a minimal set $\tilde{U} \subseteq V$ such that every edge has a node in $\tilde{U}$.
    - Equivalence: U is a maximal independent set iff U^C is a minimal vertex cover.
    - For general graphs this problem is NP-hard.
    - For bipartite graphs it's solvable through the resulted connectivity components of a max-flow algorithm, using the same construction as in the matching problem.



Matching using maximum flow in bipartite graph

Spectral Graph Theory

Main sources: Stanford's presentation, Yale's presentation and course, MIT notes and Chicago's notes.

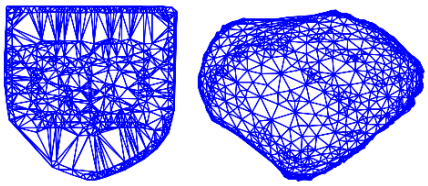
Basic definitions and properties

- ***Spectral graph theory***: use algebraic properties of matrix representation to research graphs.
- Matrix representation of an undirected graph (directed analogs are available):
  - ***Adjacency matrix***: $A_{ij} \coloneqq w_{ij}$ $\quad (= \chi_{(i,j)\in E}$ in the case of unweighted graph)
    - Eigenvalues of the matrix are associated with combinatorial properties.
  - ***Laplacian matrix***: $L \coloneqq D - A \quad (L_{ii} = d_i, L_{ij} = L_{ji} = -\chi_{(i,j)\in E})$
    - Terminology comes from being the discrete Laplacian operator – the 2nd (discrete) derivative of any function of the nodes.
    - ***Symmetric normalized Laplacian***: $L^{sym} \coloneqq D^{-1/2}LD^{-1/2} = I - D^{-1/2}AD^{-1/2}$.
- Adjacency matrix – basic properties (assuming unweighted graph):
  - $\sum_k \boldsymbol{\lambda_k^l}$ = **number of closed walks of length $\boldsymbol{l}$** (in particular $\sum\lambda_k = 0,\ \sum\lambda_k^2 = 2|E|$).
    - Similarly, the number of walks of length l between v and u can be calculated through the eigenvalues & eigenvectors of A.
  - **The largest eigenvalue $\boldsymbol{\lambda_1}$** corresponds to a (weakly-)**constant-sign eigenvector**.
    - It is also largest in absolute value (equivalently $|\lambda_n| \le \lambda_1$ for decreasing eigs).
    - $\lambda_n = -\lambda_1$ iff G is bipartite, in which case the spectrum is symmetric around 0.
    - $\bar{d} \le \lambda_1 \wedge \sqrt{\Delta(G)} \le \lambda_1 \le \Delta(G)$ (where $\Delta, \bar{d}$ are the max & average degrees of G).
  - If G is connected, then λ_1 has **multiplicity $\mathbf{1}$ and strictly-positive eigenvector**, which is the **only eigenvector** with homogeneous entries' sign.
  - If G is d-regular, then:
    - $\lambda_1 = d$ with multiplicity as the number of components, and eigenvectors which are constant over each component.
    - $\forall S,T \subset V\colon \quad (d-\lambda_2)\frac{|S||T|}{|V|} \le e(S,T) \le (d-\lambda_n)\frac{|S||T|}{|V|}$ $\quad (S,T$ may intersect).
      - Intuitively, $\boldsymbol{\lambda_2, \lambda_n}$ **determine the randomness of the graph**. In particular, $\lambda_2, \lambda_n \approx 0 \Rightarrow \left[\forall S,T \subset V\colon\ e(S,T) \approx d\frac{|S||T|}{|V|}\right]$.
  - If H is a subgraph of G, then $\lambda_1(H) \le \lambda_1(G)$.
- Laplacian matrix – basic properties (**henceforth $\boldsymbol{\lambda}$ will refer by default to $\boldsymbol{L}$'s eigenvalues**):
  - Eigenvalues are real, invariant to nodes' order, and satisfy $\mathbf{0} = \boldsymbol{\lambda_1} \le \cdots \le \boldsymbol{\lambda_n} \le \mathbf{2\Delta(G)}$.
  - $\Delta(G) + 1 \le \boldsymbol{\lambda_n} \le 2\Delta(G), \qquad \lambda_n = 2\Delta(G) \Leftrightarrow G$ is a regular bipartite graph.
  - $c \cdot L_G - L_H$ is positive semidefinite $\Rightarrow c \cdot \lambda_2(G) \ge \lambda_2(H)$.
- Eigenvectors as optimization:
  - For general symmetric matrix: $\lambda_k = \min_{U \text{ is } k-\dim \text{ subspace of } \mathrm{R^n}} \max_{x\in U} \frac{x^T M x}{x^T x}$.
  - For $M \coloneqq L$, x is actually a function $x\colon V \to R$.
  - $\frac{\boldsymbol{x^T L x}}{\boldsymbol{x^T x}} = \frac{\sum_v d_v x_v^2 - \sum_{(u,v)\in E} x_u x_v}{\sum_v x_v^2} = \frac{\sum_{(u,v)\in E}|\boldsymbol{X_u - x_v}|^2}{\sum_v \boldsymbol{x_v^2}}$.
    - In particular, $\frac{x^T L x}{x^T x} = 0 \Leftrightarrow [x_v \equiv const$ over any connectivity component$]$.
    - Hence, **always $\boldsymbol{\lambda_1} = \mathbf{0}$** and $x_1 = (1, \dots, 1)$.
    - Also, the **number of 0-eigs $\left(\max_{\lambda_k = 0} k\right)$ is the number of connectivity components**.

- Examples – spectra of standard graphs:
  - **Clique** (K_n):
    - A_{K_n}: $\begin{pmatrix} n-1 & -1 \\ 1 & n-1 \end{pmatrix}$ (i.e. $(n-1, -1, \dots, -1)$)
    - $L_{K_n} = (n-1)I - A_{K_n}$: $\begin{pmatrix} 0 & n \\ 1 & n-1 \end{pmatrix}$ (i.e. $(\ 0\ ,\ n\ , \dots ,\ n\ )$)
    - $L_{L(K_n)}$ (line-graph): $\begin{pmatrix} 2n-4 & n-4 & -2 \\ 1 & n-1 & \binom{n}{2}-n \end{pmatrix}$
  - **Cycle/path**'s Laplacian: $\lambda_k = 2 - 2\cos\frac{2\pi k}{n}$ (with multiplicity 2), $x_{k,i} \sim \cos(aik + b)$.
- ***Kirchhoff's matrix tree theorem***: the number of spanning trees in a connected graph is $\frac{1}{n}\prod_{i=1}^{n-1}\lambda_i$.
- ***Random walk*** on a graph is a Markov chain with **transitions** $\boldsymbol{T \coloneqq AD^{-1}}$ (not symmetric), where the stationary state $p_0 = Tp_0$ corresponds to $\lambda = 1$.
  - Convergence rate: $|p_t(v) - \pi(v)| \le \sqrt{d_v/\delta}(\mathbf{1} - \boldsymbol{\lambda_{n-1}})^t \qquad \left(\delta \coloneqq \min_u d_u\right)$
  - Note: $L^{sym} = I - D^{-1/2}TD^{1/2} \quad \Rightarrow \quad \mathbf{1} - \boldsymbol{\lambda_{n-1}}(\boldsymbol{T}) = \boldsymbol{\lambda_2}(\boldsymbol{L^{sym}})$.
- **Planar graphs**: $\lambda_2 < \lambda_5$ (strict inequality); $\lambda_2 \le \frac{8\Delta(G)}{|V|}$.

Spectral embedding

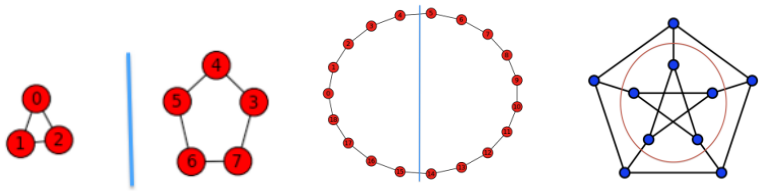
- Min squared-distance embedding: map $V \to R^k$ with minimized $x^T L X = \sum_E |x_u - x_v|^2$.
  - Goal is drawing the graph in low dim with short (or more balanced) edges, as derived from the squared penalty. The perpendicularity constraints are less intuitive to me.
  - $k = 1$: trivial is $x \equiv 1 = x(\lambda_1)$, but enforcing $x \perp \mathbf{1}$ yields $x = x(\lambda_2)$.
  - $k = 2$: trivial is $x_v = (1,1)$ or $x_v = \left(x_v(\lambda_2), x_v(\lambda_2)\right)$; enforcing certain perpendicularity yields $x_v = \left(x_v(\lambda_2), x_v(\lambda_3)\right)$.
- ***Tutte planar embedding***: fix several nodes and put every other node in the mean of its neighbors.
  - Under certain conditions, it's indeed a crossing-free planar embedding with convex faces.
- **Isomorphism testing**: graphs are isomorphic ⇔ [the spectrum is identical and eigenvectors are identical up to sign (i.e. embedding is identical up to rotations and reflections)].
  - Graphs with strong regularity or large degeneracy may require many eigenvectors to distinguish between the nodes embeddings, making the isomorphism test inefficient.
    - Polynomial running time is possible if degeneracy is small [Babai, 82].



The right graph is the min-squared-distance embedding of the left one

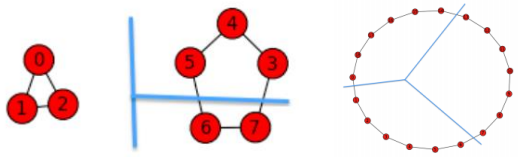
Conductance

- ***Conductance***:
  - Of a cut $S \subset V$: $\phi(S) \coloneqq \frac{E(S,\bar{S})}{vol(S)}$ $(vol(S) \coloneqq \sum_{v \in S} d_v)$
  - Of a graph G: $\phi(G) \coloneqq \min_{S \subset V \mid \mathrm{vol}(S) \leq \frac{1}{2} vol(V)} \phi(S)$
  - **Cuts of small conductance correspond to more *homogeneous* or *closed* groups S and $\bar{S}$.**
  - Finding $\phi(G)$ (***sparsest cut problem***) is NP-complete, though approximation is available in $O\left(\sqrt{\log n}\right)$ [Arora, 2009].



The conductance and corresponding min cut in 3 graphs (left to right): 0, 1/9, 1/3

- ***Cheeger inequality***: $\frac{\lambda_2}{2} \leq \boldsymbol{\phi(G)} \leq \sqrt{\boldsymbol{2\lambda_2}}$
  - Private case: $\phi(G) = 0 \Leftrightarrow G$ is disconnected $\Leftrightarrow \lambda_2 = 0$.
  - Note: convergence rate of random walk (see above) $\propto \frac{1}{\lambda_2} \propto \frac{1}{\phi^2(G)}$.
- In analog to connectivity: number of small eigs = number of disjoint sets of small conductance.
  - ***Order-k conductance*** – between k disjoint sets: $\phi_k(G) \coloneqq \min_{\text{disjoint } S_1 \dots S_k} \max_{1 \leq i \leq k} \phi(S_i)$.
  - $\boldsymbol{\phi_k(G) = 0 \Leftrightarrow G}$ **has** $\boldsymbol{\geq k}$ **connected components** $\boldsymbol{\Leftrightarrow \lambda_k = 0}$.
  - $\frac{\lambda_k}{2} \leq \phi_k(G) \leq O(k^2) \cdot \sqrt{\lambda_k}$.



The order-3 conductance and corresponding min cuts in 2 graphs (left to right): 1/2, 1/6

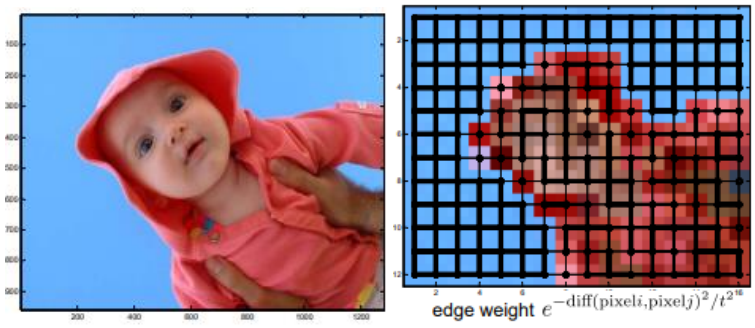
- ***Eigenvalue gap***: $\lambda_{k+1} > 5^k \sqrt{\lambda_k} \Rightarrow \exists k$ sets of small(?) conductance [Tanaka, 2012].
  - Note: analog to k connectivity components causing $\lambda_k = 0 \wedge \lambda_{k+1} > 0$.

Spectral clustering

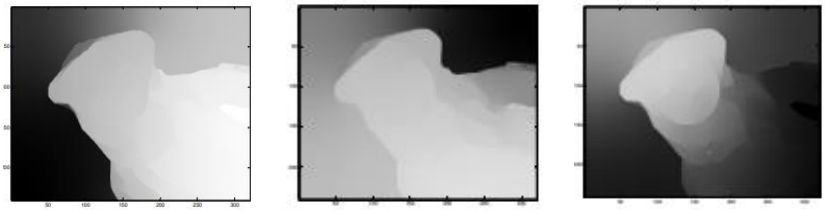
- Problem: apply clustering to data points V whose relations are expressed by a graph $G = (V, E)$.
  - Note: straight-forward generalizations of the spectral theory to weighted graph allow to express non-binary relations as well.
- ***Graph bisection*** (simple private case of ***stochastic block models*** or ***planted partition model***):
  - The graph is modeled by the adjacency matrix A whose entries are random Bernoulli variables with parameter p (inside V_1 or inside V_2) or $q < p$ (between V_1 and V_2).
  - A can be seen as perturbation of the underlying probabilities matrix (up to permutations):

$$\boldsymbol{M} = \begin{bmatrix} p & \cdots & p & q & \cdots & q \\ \vdots & & & & \vdots & \\ p & \cdots & p & q & \cdots & q \\ q & \cdots & q & p & \cdots & p \\ \vdots & & & & \vdots & \\ q & \cdots & q & p & \cdots & p \end{bmatrix}$$

  - The partition $V = V_1 \biguplus V_2$ can be recovered from the 2nd(-largest) eigenvector of A.
  - In general, perturbation theory can be very powerful for analysis of random objects. "For example, it inspired Shkolnisky and Singer to design an exciting algorithm for the image processing problems that occur in cryo-electron microscopy" [Spielman, Yale].
- Eigenvectors-based clustering:
  - Compute k smallest eigenvectors of L $(x_1 \dots x_k)$.
  - Define the k-dimensional spectral embedding $F: V \to R^k$, $F(v) \coloneqq (x_1^v \dots x_k^v)$.
  - Apply k-means.
- Conductance-based clustering (LRTV?): find k small-conductance sets (not always trivial to do).
- Image segmentation: detect objects as "closed" (i.e. small-conductance) sets of pixels.
  - Current spectral segmentation methods mostly don't use any former image processing tools. Future incorporation of such essentially different methods may be beneficial.



Representation of an image as a graph of pixels



2nd, 3rd and 4th eigenvectors assign large (white) values to different small-conductance sets

Social Network Analysis (SNA)

Main sources: SNA introduction notes by UVA, MIT , Analytic Technologies and Orgnet.

Basic definitions

- ***Social network analysis*** is the science of connections between human entities.
- Main research approaches:
  - ***Socio-centered***: focused on global properties and structure of the graph of connections.
    - Example – ***network centralization***: measure of the imbalance (i.e. variance) in centrality of nodes – how much the whole network is centralized in few nodes.
  - ***Ego-centered***: focused on local properties and structures, small-range influences, etc.
    - Example – ***network reach***: it is claimed that in certain senses, no significant influence can reach farther than 2 steps in a network, leading to measuring local centrality through the number of connections in distance ≤ 2.
- **SNA model $\boldsymbol{N = (V, L, F_V, F_L)}$** is a **generalization of a graph**: **$\boldsymbol{V}$=nodes**; **L=links** (may include both directed & undirected, and more than one link between the same 2 nodes); **$\boldsymbol{F_V, F_L}$ are functions specifying the properties** of the nodes & links respectively.
  - Terminology: ***actors*** = nodes = vertices; ***links*** = ties = edges; ***geodesics*** = shortest paths.
- ***Sociogram***: visual graph representation of a social network. May visualize F_V & F_L as well (e.g. through size of nodes and width of edges).
  - ***Sociomatrix***: adjacency matrix of (either all or part of the) social links.

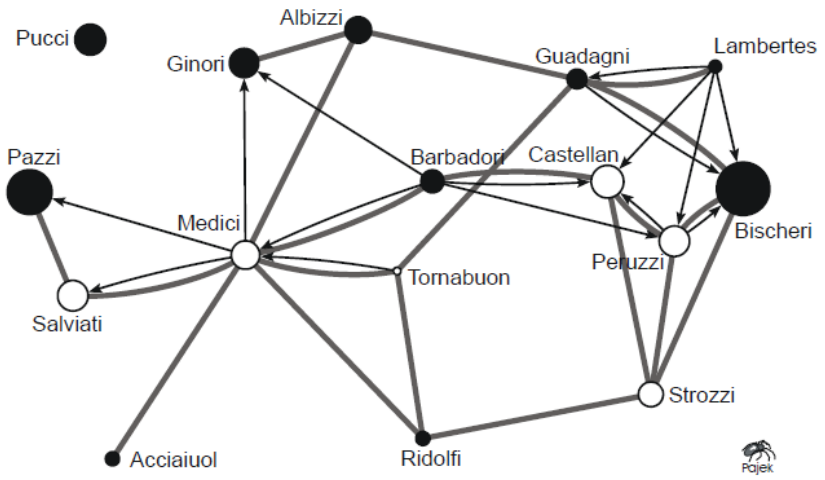
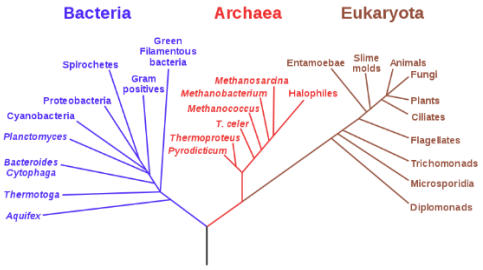


Figure 1 - Sociogram of marriage (grey edges) and business (black arcs) relations between 16 Florentine families (circa 1400 AD).

- Mid-cohesive types of subgraphs (between connectivity-components and cliques):
  - ***n-clique***: maximal subgraph in which $\forall u, v: d(u, v) \leq n$ ($n = 1 \Rightarrow$ standard clique).
  - ***n-clan***: an n-clique in which all $d(u, v)$'s paths use only nodes from within the subgraph.
  - ***n-club***: maximal subgraph with diameter n.
    - n-clan $\Rightarrow$ n-club $\wedge$ n-clique.
    - n-club doesn't have to satisfy the maximality of n-clique, hence $\not\Rightarrow$ n-clan.
  - ***k-plex***: subgraph in which $\forall v: d_v \geq n - k$ (degree within the subgraph).
  - ***k-core***: subgraph in which $\forall v: d_v \geq k$ (degree within the subgraph).
  - ***Level-c clique***: any u, v share at least c neighbors (i.e. have ***affiliation*** $\geq c$).

Popular metrics

- Size metrics: $n = |V|,\ m = |E|$.
- Connectivity (***cohesion***) metrics:
  - ***Density***: $|E|/\binom{n}{2}$.
  - ***Diameter*** (max distance between two nodes), average distance between nodes.
  - Number of **connectivity components** / **small-conductance sets** / **cliques**.
  - ***Clustering coefficient*** of a node v: $C_v \coloneqq \frac{\text{number of edges between neighbors of } v}{\text{number of possible edges between them}}$.
    - Clustering coefficient of a graph: $C \coloneqq \frac{1}{n}\sum_v C_v = \frac{\text{number of closed triplets}}{\text{number of all triplets}}$.
    - "How cohesive the neighborhood is".
  - Subgroup cohesion: $\frac{\text{percent of connections within subgroup}}{\text{percent of connections between subgroup and the outside}}$.
- Node centrality metrics:
  - ***Degree*** (***activity***): number of edges.
  - ***Betweenness***: number of shortest paths containing the node.
    - ***Girvan-Newman algorithm*** iteratively removes the largest-betweenness edge in order to detect communities in complex networks, yielding a ***dendrogram***.



Example: a dendrogram of the tree of life

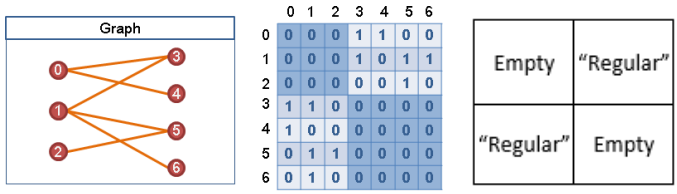
  - ***Closeness*** (***efficiency***): average distance to other (reachable) nodes.
  - ***Eigenvector centrality*** (*eigencentrality*): the first (largest) eigenvector of the adjacency matrix (note that all the entries are non-negative). In particular, $x_u = \frac{1}{\lambda}\sum_v a_{uv}x_v$.
  - ***Information centrality***: $IC(u) \coloneqq \frac{1}{\frac{1}{n}\sum_v d(u,v)} =$ harmonic average of the "information" $I_{uv} \coloneqq \frac{1}{d(u,v)}$ (intuitively measuring SNR where the noise $\propto d(u,v)$).
  - ***Random walk centrality*** (*Markov centrality*): expected number of steps required to reach the node in a random walk beginning at another random node.
  - Note: asymmetric centrality (e.g. *popularity*, *prestige*, *ranking*) should use these metrics (possibly under variations) wrt directed edges.

	Transfer	Serial	Parallel
Walks	Money exchange	Emotional support	Attitude influencing
Trails	Used Book	Gossip	E-mail broadcast
Paths	Mooch	Viral infection	Internet name-server
Geodesics	Package Delivery	Mitotic reproduction	<no process>

Examples for things that may flow in social networks (MIT).
Choice of centrality metric should correspond to the context of the connections.

Further models and methods

- ***Blockmodeling***: sort the adjacency matrix to have meaningful blocks.

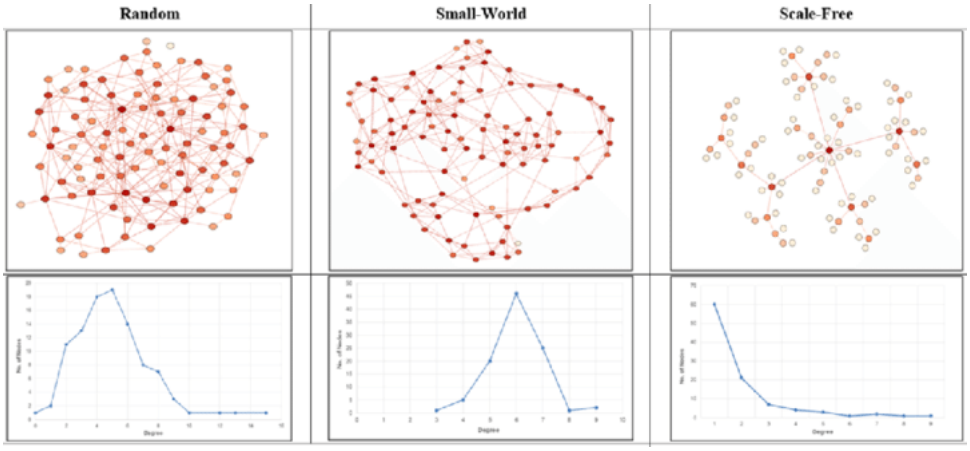


Example: a bipartite graph, its (sorted) adjacency matrix, and the corresponding block-model

- ***Statistical actor-oriented models*** (ego-centered approach): form hypotheses regarding local properties in the network (e.g. "there are more occurrences of a certain local pattern than there would have been in random") and test them using statistical tools, e.g. ***Exponential Random Graph Models*** (***ERGM***) and ***Markov Chain Monte Carlo*** (***MCMC***).

Small-world network

- ***Small-world*** is a model of network consisting of **dense clusters with sparse links between them**.
- Small-world networks keep both **small average distance** ("close to everyone") and **large clustering coefficient** ("non-sparse neighborhood") per total number of links.
  - Specifically $\boldsymbol{L} \coloneqq \boldsymbol{E}[\boldsymbol{d}(\boldsymbol{u},\boldsymbol{v})] \propto \boldsymbol{\log n}$, explaining the famous of 6 degrees of separation.
  - Allowed by relatively large number of nodes with very large degree (***hubs***), or equivalently – by fat-tailed distribution of the nodes' degree.
  - Analog to traffic routes: most roads are local, highroads connect cities, and air-lines connect countries.
- The opposite is a network with only local links (e.g. lattice graph, or network of connections between people in various centuries), resulting in large average distance (***large-world***).
- Several methods to construct small-world networks are available. ***Watts-Strogatz mechanism***, for example, constructs a small-world graph as a mix of lattice and random graph.
- The dependence on small number of hubs makes the network's average distance **robust to deletion of random nodes** (since most nodes are peripheral), even though it is sensitive to adversary deletion of the hubs.
  - The robustness to random perturbations is hypothesized to have made **relations between genes behaving as a small-world network**.
- Small-world metrics:
  - ***Small-worldness***: $\sigma \coloneqq \frac{C/C_{rand}}{L/L_{rand}}$ (clustering coefficient & average distance).
    - Small-world $\Leftrightarrow \sigma > 1$.
    - This metric is often too sensitive to the network's size.
  - ***Small-world index***: $SWI \coloneqq \frac{L-L_{lattice}}{L_{rand}-L_{lattice}} \cdot \frac{C-C_{lattice}}{C_{lattice}-C_{rand}}$ $(0 \leq SWI < 1)$.
    - SWI-ideal network (unreachable): $C = C_{lattice} \wedge L = L_{rand}$.
- ***Scale-free network*** (***ultra-small world***): $P(d_v \geq k) \sim k^{-\gamma}$ asymptotically, yielding $L \propto \log\log n$.



Random, small-world and scale-free networks – along with the corresponding distributions of degrees