Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 4: Graph Data

Graphs

A *graph* $G = (V, E)$ comprises a finite nonempty set V of *vertices* or *nodes*, and a set $E \subseteq V \times V$ of *edges* consisting of *unordered* pairs of vertices.

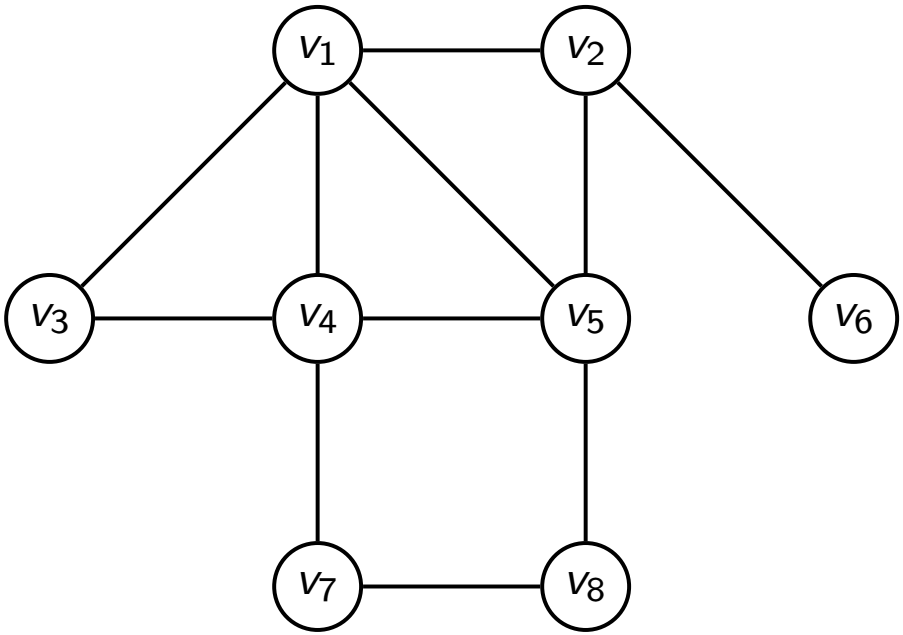
The number of nodes in the graph G, given as $|V| = n$, is called the *order* of the graph, and the number of edges in the graph, given as $|E| = m$, is called the *size* of G.

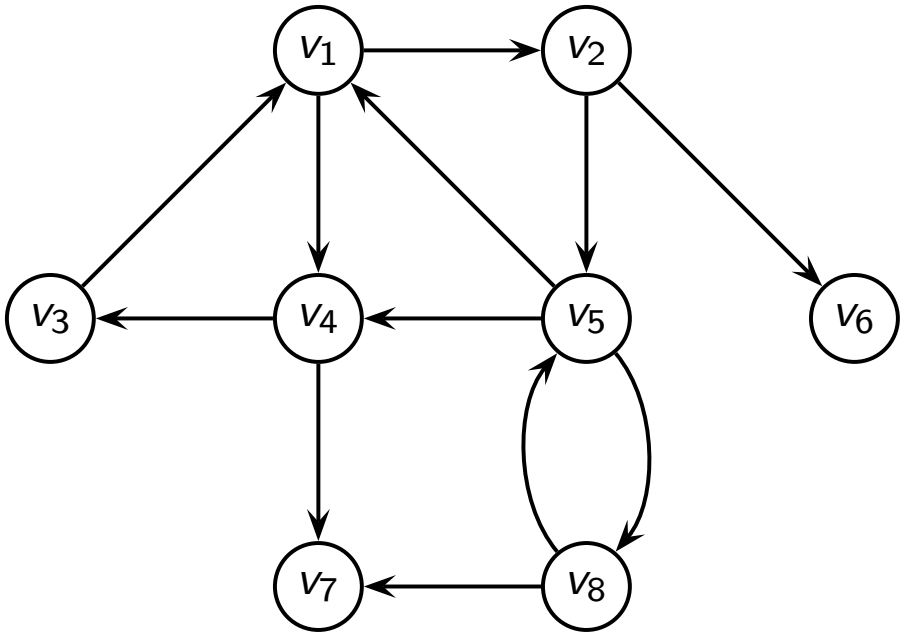
A *directed graph* or *digraph* has an edge set E consisting of *ordered* pairs of vertices.

A *weighted graph* consists of a graph together with a weight w_{ij} for each edge $(v_i, v_j) \in E$.

A graph $H = (V_H, E_H)$ is called a *subgraph* of $G = (V, E)$ if $V_H \subseteq V$ and $E_H \subseteq E$.

Undirected and Directed Graphs





Degree Distribution

The *degree* of a node $v_i \in V$ is the number of edges incident with it, and is denoted as $d(v_i)$ or just d_i.

The *degree sequence* of a graph is the list of the degrees of the nodes sorted in non-increasing order.

Let N_k denote the number of vertices with degree k. The *degree frequency distribution* of a graph is given as

$$(N_0, N_1, \ldots, N_t)$$

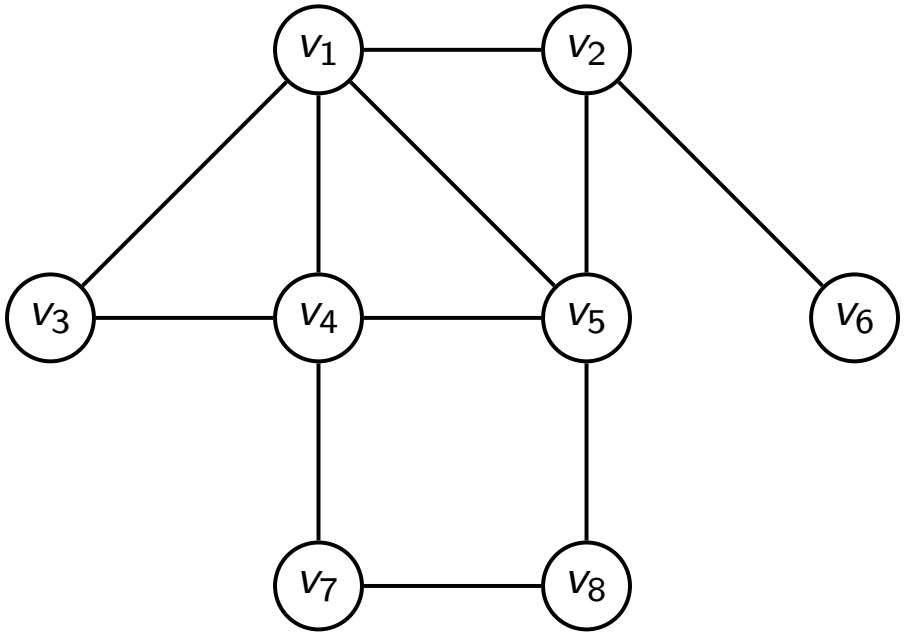
where t is the maximum degree for a node in G.

Let X be a random variable denoting the degree of a node. The *degree distribution* of a graph gives the probability mass function f for X, given as

$$\big(f(0), f(1), \ldots, f(t)\big)$$

where $f(k) = P(X = k) = \frac{N_k}{n}$ is the probability of a node with degree k.

Degree Distribution



The degree sequence of the graph is

$$(4, 4, 4, 3, 2, 2, 2, 1)$$

Its degree frequency distribution is

$$(N_0, N_1, N_2, N_3, N_4) = (0, 1, 3, 1, 3)$$

The degree distribution is given as

$$\big(f(0), f(1), f(2), f(3), f(4)\big) = (0, 0.125, 0.375, 0.125, 0.375)$$

Path, Distance and Connectedness

A *walk* in a graph G between nodes x and y is an ordered sequence of vertices, starting at x and ending at y,

$$x = v_0,\ v_1,\ \ldots,\ v_{t-1},\ v_t = y$$

such that there is an edge between every pair of consecutive vertices, that is, $(v_{i-1}, v_i) \in E$ for all $i = 1, 2, \ldots, t$. The length of the walk, t, is measured in terms of *hops* – the number of edges along the walk.

A *path* is a walk with *distinct* vertices (with the exception of the start and end vertices). A path of minimum length between nodes x and y is called a *shortest path*, and the length of the shortest path is called the *distance* between x and y, denoted as $d(x, y)$. If no path exists between the two nodes, the distance is assumed to be $d(x, y) = \infty$.

Two nodes v_i and v_j are *connected* if there exists a path between them. A graph is *connected* if there is a path between all pairs of vertices. A *connected component*, or just *component*, of a graph is a maximal connected subgraph.

A directed graph is *strongly connected* if there is a (directed) path between all ordered pairs of vertices. It is *weakly connected* if there exists a path between node pairs only by considering edges as undirected.

Adjacency Matrix

A graph $G = (V, E)$, with $|V| = n$ vertices, can be represented as an $n \times n$, symmetric binary *adjacency matrix*, $\boldsymbol{A}$, defined as

$$\boldsymbol{A}(i,j) = \begin{cases} 1 & \text{if } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$$

If the graph is directed, then the adjacency matrix $\boldsymbol{A}$ is not symmetric.

If the graph is weighted, then we obtain an $n \times n$ *weighted adjacency matrix*, $\boldsymbol{A}$, defined as

$$\boldsymbol{A}(i,j) = \begin{cases} w_{ij} & \text{if } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$$

where w_{ij} is the weight on edge $(v_i, v_j) \in E$.

Graphs from Data Matrix

Many datasets that are not in the form of a graph can still be converted into one.

Let $\boldsymbol{D} = \{\boldsymbol{x}_i\}_{i=1}^n$ (with $\boldsymbol{x}_i \in \mathbb{R}^d$), be a dataset. Define a weighted graph $G = (V, E)$, with edge weight

$$w_{ij} = sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

where $sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denotes the similarity between points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

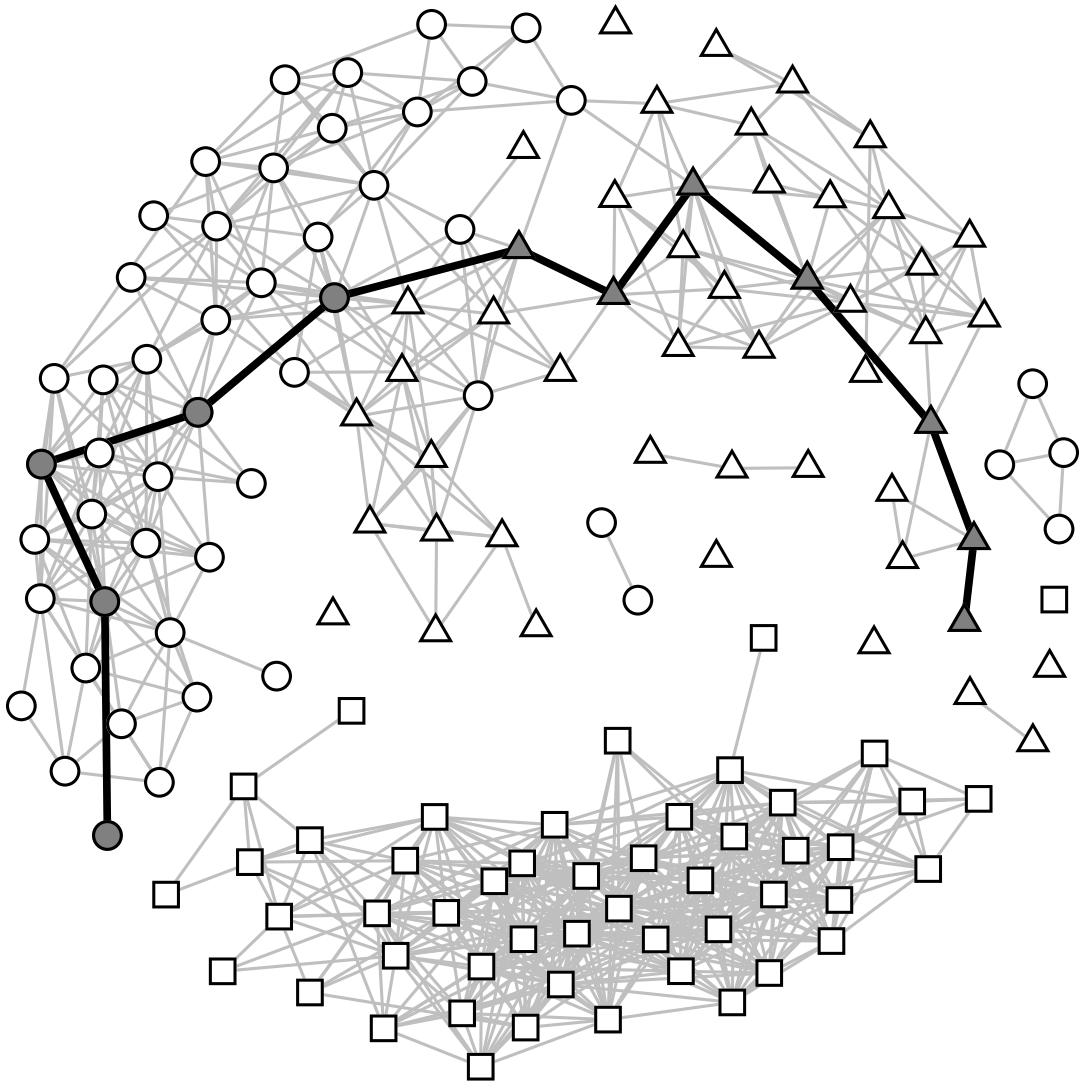
For instance, using the Gaussian similarity

$$w_{ij} = sim(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right\}$$

where σ is the spread parameter.

Iris Similarity Graph: Gaussian Similarity

$\sigma = 1/\sqrt{2}$; edge exists iff $w_{ij} \geq 0.777$
order: $|V| = n = 150$; size: $|E| = m = 753$



Topological Graph Attributes

Graph attributes are *local* if they apply to only a single node (or an edge), and *global* if they refer to the entire graph.

Degree: The degree of a node $v_i \in G$ is defined as

$$d_i = \sum_j \boldsymbol{A}(i,j)$$

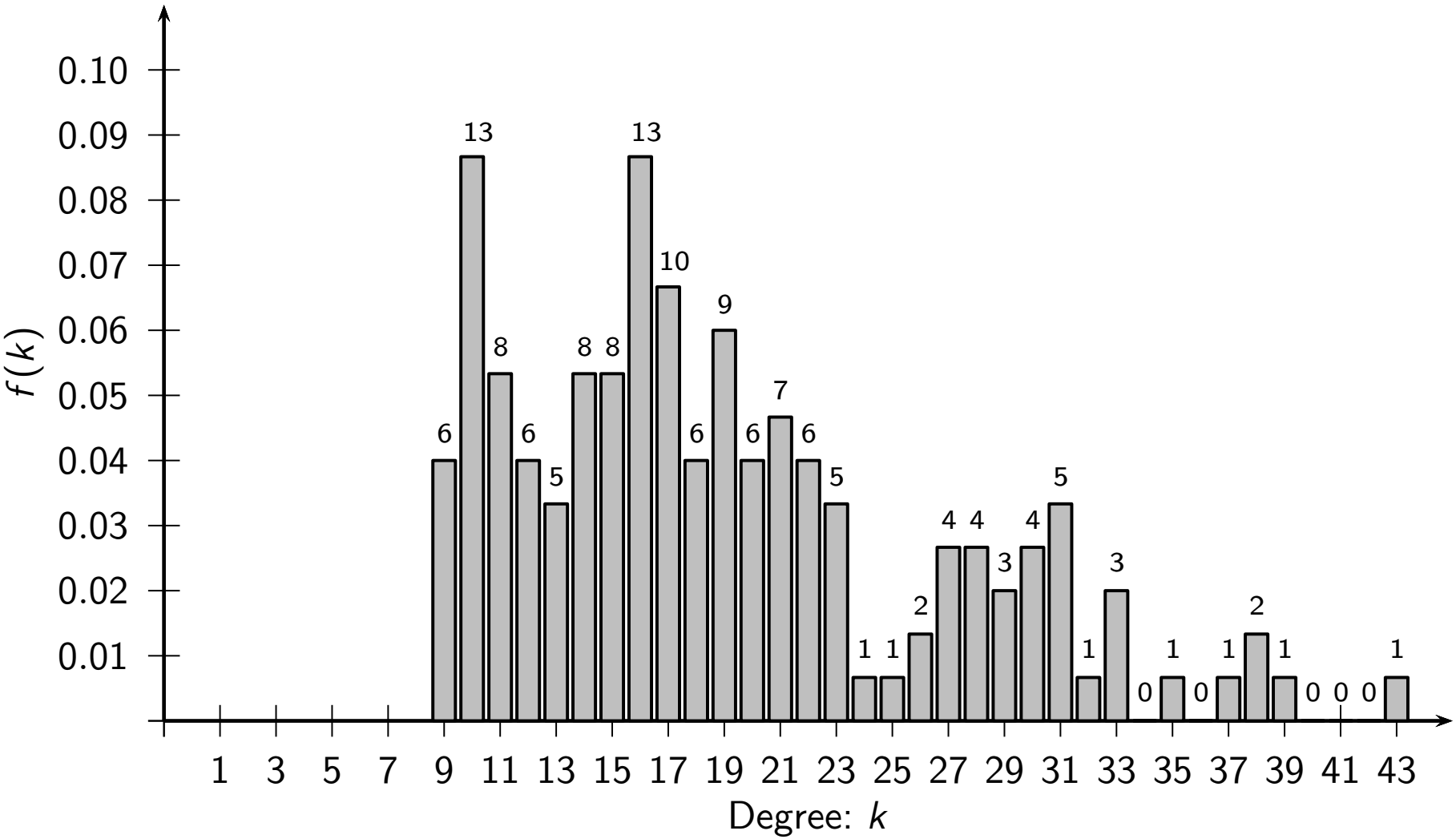
The corresponding global attribute for the entire graph G is the *average degree*:

$$\mu_d = \frac{\sum_i d_i}{n}$$

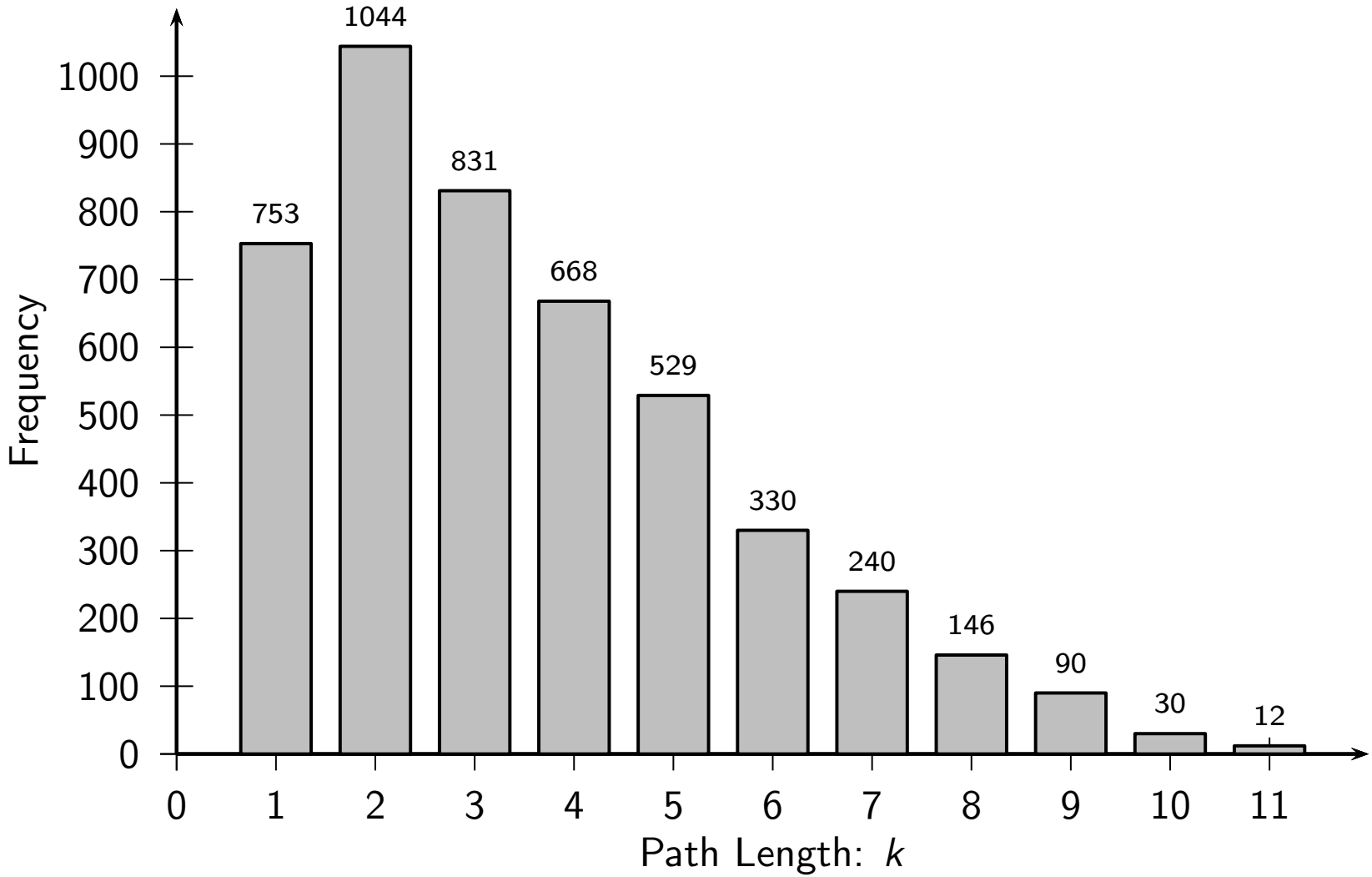
Average Path Length: The *average path length* is given as

$$\mu_L = \frac{\sum_i \sum_{j>i} d(v_i, v_j)}{\binom{n}{2}} = \frac{2}{n(n-1)} \sum_i \sum_{j>i} d(v_i, v_j)$$

Iris Graph: Degree Distribution



Iris Graph: Path Length Histogram



Eccentricity, Radius and Diameter

The *eccentricity* of a node v_i is the maximum distance from v_i to any other node in the graph:

$$e(v_i) = \max_j \{d(v_i, v_j)\}$$

The *radius* of a connected graph, denoted $r(G)$, is the minimum eccentricity of any node in the graph:

$$r(G) = \min_i \{e(v_i)\} = \min_i \left\{ \max_j \{d(v_i, v_j)\} \right\}$$

The *diameter*, denoted $d(G)$, is the maximum eccentricity of any vertex in the graph:

$$d(G) = \max_i \{e(v_i)\} = \max_{i,j} \{d(v_i, v_j)\}$$

For a disconnected graph, values are computed over the connected components of the graph.

The diameter of a graph G is sensitive to outliers. *Effective diameter* is more robust; defined as the minimum number of hops for which a large fraction, typically 90%, of all connected pairs of nodes can reach each other.

Clustering Coefficient

The *clustering coefficient* of a node v_i is a measure of the density of edges in the neighborhood of v_i.

Let $G_i = (V_i, E_i)$ be the subgraph induced by the neighbors of vertex v_i. Note that $v_i \notin V_i$, as we assume that G is simple.

Let $|V_i| = n_i$ be the number of neighbors of v_i, and $|E_i| = m_i$ be the number of edges among the neighbors of v_i. The clustering coefficient of v_i is defined as

$$C(v_i) = \frac{\text{no. of edges in } G_i}{\text{maximum number of edges in } G_i} = \frac{m_i}{\binom{n_i}{2}} = \frac{2 \cdot m_i}{n_i(n_i - 1)}$$

The *clustering coefficient* of a graph G is simply the average clustering coefficient over all the nodes, given as

$$C(G) = \frac{1}{n} \sum_i C(v_i)$$

$C(v_i)$ is well defined only for nodes with degree $d(v_i) \geq 2$, thus define $C(v_i) = 0$ if $d_i < 2$.

Transitivity and Efficiency

Transitivity of the graph is defined as

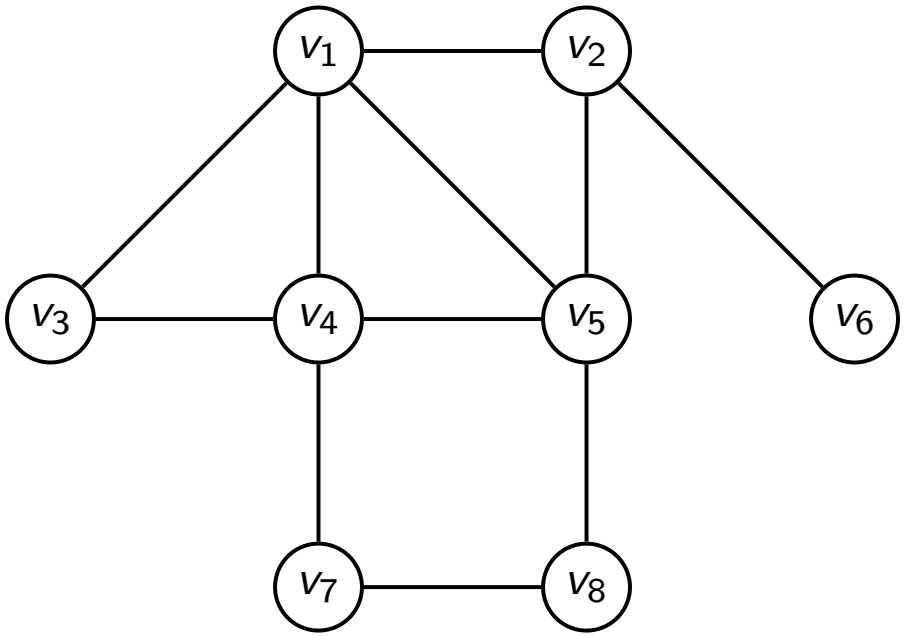
$$T(G) = \frac{3 \times \text{no. of triangles in } G}{\text{no. of connected triples in } G}$$

where the subgraph composed of the edges (v_i, v_j) and (v_i, v_k) is a *connected triple* centered at v_i, and a connected triple centered at v_i that includes (v_j, v_k) is called a *triangle* (a complete subgraph of size 3).

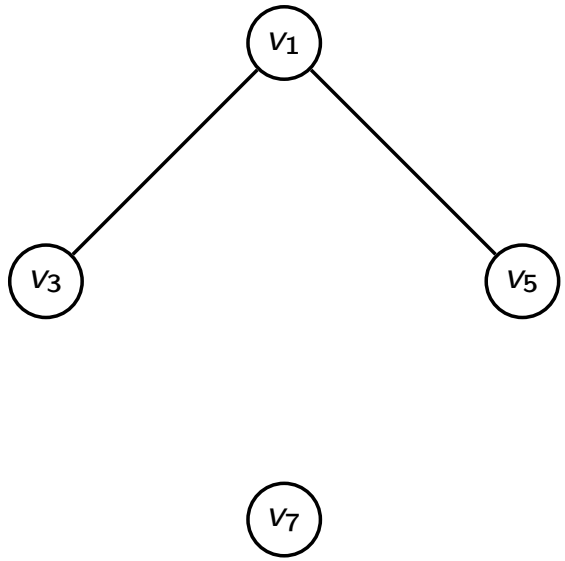
The *efficiency* for a pair of nodes v_i and v_j is defined as $\frac{1}{d(v_i, v_j)}$. If v_i and v_j are not connected, then $d(v_i, v_j) = \infty$ and the efficiency is $1/\infty = 0$.
The *efficiency* of a graph G is the average efficiency over all pairs of nodes, whether connected or not, given as

$$\frac{2}{n(n-1)} \sum_i \sum_{j>i} \frac{1}{d(v_i, v_j)}$$

Clustering Coefficient



Subgraph induced by node v_4:



The clustering coefficient of v_4 is

$$C(v_4) = \frac{2}{\binom{4}{2}} = \frac{2}{6} = 0.33$$

The clustering coefficient for G is

$$C(G) = \frac{1}{8}\left(\frac{1}{2} + \frac{1}{3} + 1 + \frac{1}{3} + \frac{1}{3}\right) = \frac{2.5}{8} = 0.3125$$

Centrality Analysis

A centrality is a function $c: V \to \mathbb{R}$, that induces a ranking on V.

Degree Centrality: The simplest notion of centrality is the degree d_i of a vertex v_i – the higher the degree, the more important or central the vertex.

Eccentricity Centrality: Eccentricity centrality is defined as:

$$c(v_i) = \frac{1}{e(v_i)} = \frac{1}{\max_j \{d(v_i, v_j)\}}$$

The less eccentric a node is, the more central it is.

Closeness Centrality: closeness centrality uses the sum of all the distances to rank how central a node is

$$c(v_i) = \frac{1}{\sum_j d(v_i, v_j)}$$

Betweenness Centrality

The betweenness centrality measures how many shortest paths between all pairs of vertices include v_i. It gives an indication as to the central "monitoring" role played by v_i for various pairs of nodes.

Let η_{jk} denote the number of shortest paths between vertices v_j and v_k, and let $\eta_{jk}(v_i)$ denote the number of such paths that include or contain v_i.

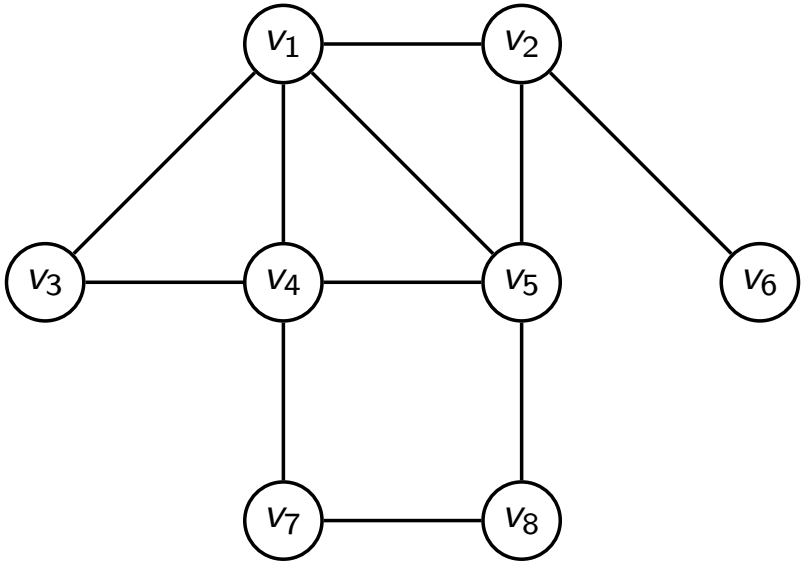
The fraction of paths through v_i is denoted as

$$\gamma_{jk}(v_i) = \frac{\eta_{jk}(v_i)}{\eta_{jk}}$$

The betweenness centrality for a node v_i is defined as

$$c(v_i) = \sum_{j \neq i} \sum_{\substack{k \neq i \\ k > j}} \gamma_{jk} = \sum_{j \neq i} \sum_{\substack{k \neq i \\ k > j}} \frac{\eta_{jk}(v_i)}{\eta_{jk}}$$

Centrality Values



Centrality	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
Degree	4	3	2	4	4	1	2	2
Eccentricity	0.5	0.33	0.33	0.33	0.5	0.25	0.25	0.33
$e(v_i)$	2	3	3	3	2	4	4	3
Closeness	0.100	0.083	0.071	0.091	0.100	0.056	0.067	0.071
$\sum_j d(v_i, v_j)$	10	12	14	11	10	18	15	14
Betweenness	4.5	6	0	5	6.5	0	0.83	1.17

Prestige or Eigenvector Centrality

Let $p(u)$ be a positive real number, called the *prestige* score for node u. Intuitively the more (prestigious) the links that point to a given node, the higher its prestige.

$$p(v) = \sum_u \boldsymbol{A}(u,v) \cdot p(u)$$
$$= \sum_u \boldsymbol{A}^T(v,u) \cdot p(u)$$

Across all the nodes, we have

$$\boldsymbol{p}' = \boldsymbol{A}^T \boldsymbol{p}$$

where $\boldsymbol{p}$ is an n-dimensional prestige vector.
By recursive expansion, we see that

$$\boldsymbol{p}_k = \boldsymbol{A}^T \boldsymbol{p}_{k-1} = \left(\boldsymbol{A}^T\right)^2 \boldsymbol{p}_{k-2} = \cdots = \left(\boldsymbol{A}^T\right)^k \boldsymbol{p}_0$$

where $\boldsymbol{p}_0$ is the initial prestige vector. It is well known that the vector $\boldsymbol{p}_k$ converges to the dominant eigenvector of $\boldsymbol{A}^T$.

Computing Dominant Eigenvector: Power Iteration

The dominant eigenvector of $\boldsymbol{A}^T$ and the corresponding eigenvalue can be computed using the *power iteration* method.

It starts with an initial vector $\boldsymbol{p}_0$, and in each iteration, it multiplies on the left by $\boldsymbol{A}^T$, and scales the intermediate $\boldsymbol{p}_k$ vector by dividing it by the maximum entry $\boldsymbol{p}_k[i]$ in $\boldsymbol{p}_k$ to prevent numeric overflow.

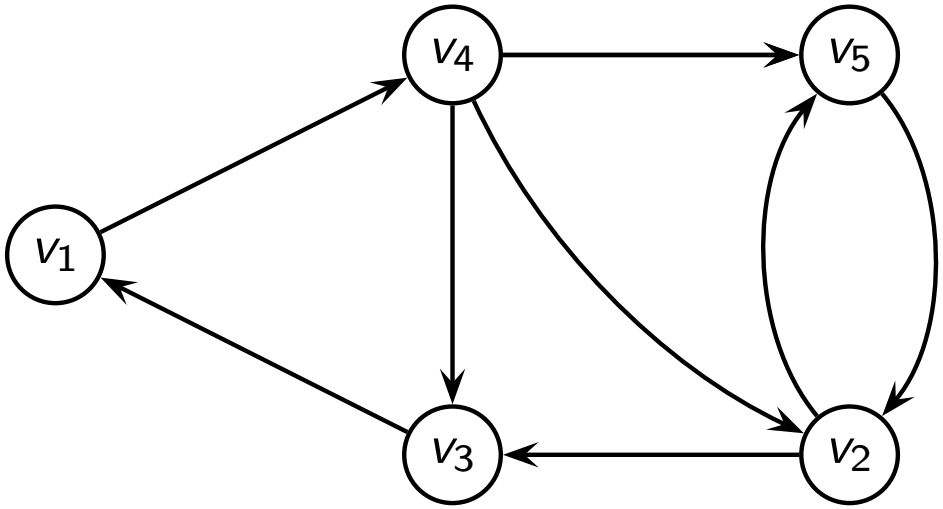
The ratio of the maximum entry in iteration k to that in $k-1$, given as $\lambda = \frac{\boldsymbol{p}_k[i]}{\boldsymbol{p}_{k-1}[i]}$, yields an estimate for the eigenvalue.

The iterations continue until the difference between successive eigenvector estimates falls below some threshold $\epsilon > 0$.

PowerIteration $(\boldsymbol{A}, \epsilon)$:

$k \leftarrow 0$ // iteration
$\boldsymbol{p}_0 \leftarrow 1 \in \mathbb{R}^n$ // initial vector
repeat
  $k \leftarrow k+1$ $\boldsymbol{p}_k \leftarrow \boldsymbol{A}^T \boldsymbol{p}_{k-1}$ // eigenvector estimate
  $i \leftarrow \arg\max_j \{\boldsymbol{p}_k[j]\}$ // maximum value index
  $\lambda \leftarrow \boldsymbol{p}_k[i] / \boldsymbol{p}_{k-1}[i]$ // eigenvalue estimate
  $\boldsymbol{p}_k \leftarrow \frac{1}{\boldsymbol{p}_k[i]} \boldsymbol{p}_k$ // scale vector
until $\|\boldsymbol{p}_k - \boldsymbol{p}_{k-1}\| \leq \epsilon$
$\boldsymbol{p} \leftarrow \frac{1}{\|\boldsymbol{p}_k\|} \boldsymbol{p}_k$ // normalize eigenvector
return $\boldsymbol{p}, \lambda$

Power Iteration for Eigenvector Centrality: Example



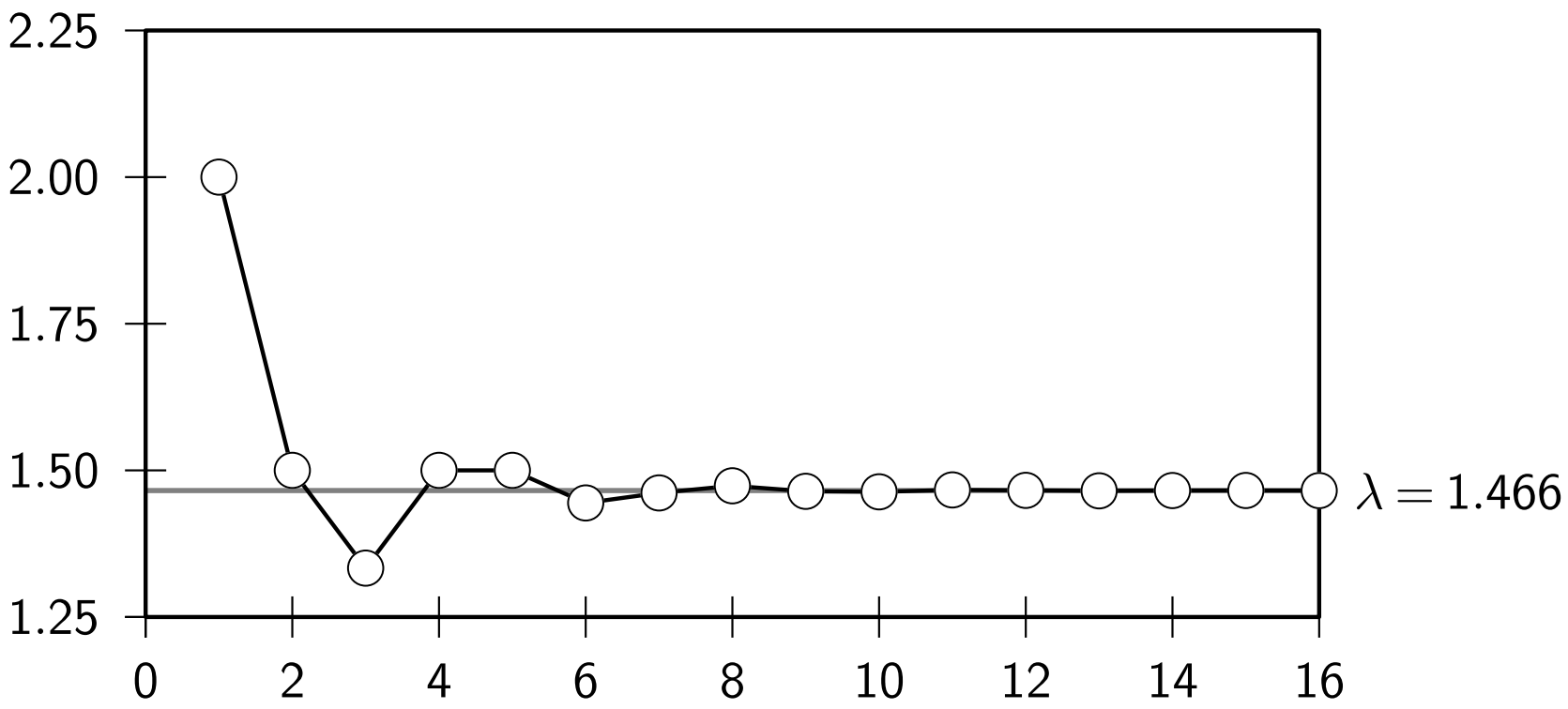
$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \qquad \boldsymbol{A}^T = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Power Method via Scaling

$\boldsymbol{p}_0$	$\boldsymbol{p}_1$	$\boldsymbol{p}_2$	$\boldsymbol{p}_3$
$\begin{pmatrix}1\\1\\1\\1\\1\end{pmatrix}$	$\begin{pmatrix}1\\2\\2\\1\\2\end{pmatrix} \rightarrow \begin{pmatrix}0.5\\1\\1\\0.5\\1\end{pmatrix}$	$\begin{pmatrix}1\\1.5\\1.5\\0.5\\1.5\end{pmatrix} \rightarrow \begin{pmatrix}0.67\\1\\1\\0.33\\1\end{pmatrix}$	$\begin{pmatrix}1\\1.33\\1.33\\0.67\\1.33\end{pmatrix} \rightarrow \begin{pmatrix}0.75\\1\\1\\0.5\\1\end{pmatrix}$
λ	2	1.5	1.33

$\boldsymbol{p}_4$	$\boldsymbol{p}_5$	$\boldsymbol{p}_6$	$\boldsymbol{p}_7$
$\begin{pmatrix}1\\1.5\\1.5\\0.75\\1.5\end{pmatrix} \rightarrow \begin{pmatrix}0.67\\1\\1\\0.5\\1\end{pmatrix}$	$\begin{pmatrix}1\\1.5\\1.5\\0.67\\1.5\end{pmatrix} \rightarrow \begin{pmatrix}0.67\\1\\1\\0.44\\1\end{pmatrix}$	$\begin{pmatrix}1\\1.44\\1.44\\0.67\\1.44\end{pmatrix} \rightarrow \begin{pmatrix}0.69\\1\\1\\0.46\\1\end{pmatrix}$	$\begin{pmatrix}1\\1.46\\1.46\\0.69\\1.46\end{pmatrix} \rightarrow \begin{pmatrix}0.68\\1\\1\\0.47\\1\end{pmatrix}$
1.5	1.5	1.444	1.462

Convergence of the Ratio to Dominant Eigenvalue



PageRank

PageRank is based on (normalized) prestige combined with a *random jump* assumption. The PageRank of a node v recursively depends on the PageRank of other nodes that point to it.

Normalized Prestige: Define $\boldsymbol{N}$ as the *normalized adjacency matrix*

$$\boldsymbol{N}(u,v) = \begin{cases} \frac{1}{od(u)} & \text{if } (u,v) \in E \\ 0 & \text{if } (u,v) \notin E \end{cases}$$

where $od(u)$ is the out-degree of node u.
Normalized prestige is given as

$$p(v) = \sum_u \boldsymbol{N}^T(v,u) \cdot p(u)$$

Random Jumps: In the random surfing approach, there is a small probability of jumping from one node to any of the other nodes in the graph. The normalized adjacency matrix for a fully connected graph is

$$\boldsymbol{N}_r = \frac{1}{n} 1_{n \times n}$$

where $1_{n \times n}$ is the $n \times n$ matrix of all ones.

PageRank: Normalized Prestige and Random Jumps

The PageRank vector is recursively defined as

$$\begin{aligned} \boldsymbol{p}' &= (1-\alpha)\boldsymbol{N}^T\boldsymbol{p} + \alpha\boldsymbol{N}_r^T\boldsymbol{p} \\ &= \left((1-\alpha)\boldsymbol{N}^T + \alpha\boldsymbol{N}_r^T\right)\boldsymbol{p} \\ &= \boldsymbol{M}^T\boldsymbol{p} \end{aligned}$$

α denotes the probability of random jumps. The solution is the dominant eigenvector of $\boldsymbol{M}^T$, where $\boldsymbol{M} = (1-\alpha)\boldsymbol{N} + \alpha\boldsymbol{N}_r$ is the combined normalized adjacency matrix.

Sink Nodes: If $od(u) = 0$, then only random jumps from u are allowed. The modified $\boldsymbol{M}$ matrix is given as

$$\boldsymbol{M}_u = \begin{cases} \boldsymbol{M}_u & \text{if } od(u) > 0 \\ \frac{1}{n}1_n^T & \text{if } od(u) = 0 \end{cases}$$

where 1_n is the n-dimensional vector of all ones.

Hub and Authority Scores (HITS)

The *authority score* of a page is analogous to PageRank or prestige, and it depends on how many "good" pages point to it. The *hub score* of a page is based on how many "good" pages it points to. In other words, a page with high authority has many hub pages pointing to it, and a page with high hub score points to many pages that have high authority.

Let $a(u)$ be the authority score and $h(u)$ the hub score of node u. We have:

$$a(v) = \sum_u \boldsymbol{A}^T(v,u) \cdot h(u)$$

$$h(v) = \sum_u \boldsymbol{A}(v,u) \cdot a(u)$$

In matrix notation, we obtain

$$\boldsymbol{a}' = \boldsymbol{A}^T \boldsymbol{h} \qquad\qquad \boldsymbol{h}' = \boldsymbol{A}\boldsymbol{a}$$

Recursively, we have:

$$\boldsymbol{a}_k = \boldsymbol{A}^T \boldsymbol{h}_{k-1} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{a}_{k-1}) = (\boldsymbol{A}^T\boldsymbol{A})\boldsymbol{a}_{k-1}$$

$$\boldsymbol{h}_k = \boldsymbol{A}\boldsymbol{a}_{k-1} = \boldsymbol{A}(\boldsymbol{A}^T \boldsymbol{h}_{k-1}) = (\boldsymbol{A}\boldsymbol{A}^T)\boldsymbol{h}_{k-1}$$

The authority score converges to the dominant eigenvector of $\boldsymbol{A}^T\boldsymbol{A}$, whereas the hub score converges to the dominant eigenvector of $\boldsymbol{A}\boldsymbol{A}^T$.

Small World Property

Real-world graphs exhibit the *small-world* property that there is a short path between any pair of nodes. A graph G exhibits small-world behavior if the average path length μ_L scales logarithmically with the number of nodes in the graph, that is, if

$$\mu_L \propto \log n$$

where n is the number of nodes in the graph.

A graph is said to have *ultra-small-world* property if the average path length is much smaller than $\log n$, that is, if $\mu_L \ll \log n$.

Scale-free Property

In many real-world graphs it has been observed that the empirical degree distribution $f(k)$ exhibits a *scale-free* behavior captured by a power-law relationship with k, that is, the probability that a node has degree k satisfies the condition

$$f(k) \propto k^{-\gamma}$$

Taking the logarithm on both sides gives

$$\log f(k) = \log(\alpha k^{-\gamma})$$
$$\text{or } \log f(k) = -\gamma \log k + \log \alpha$$

which is the equation of a straight line in the log-log plot of k versus $f(k)$, with $-\gamma$ giving the slope of the line.

A power-law relationship leads to a scale-free or scale invariant behavior because scaling the argument by some constant c does not change the proportionality.

Clustering Effect

Real-world graphs often also exhibit a *clustering effect*, that is, two nodes are more likely to be connected if they share a common neighbor. The clustering effect is captured by a high clustering coefficient for the graph G.

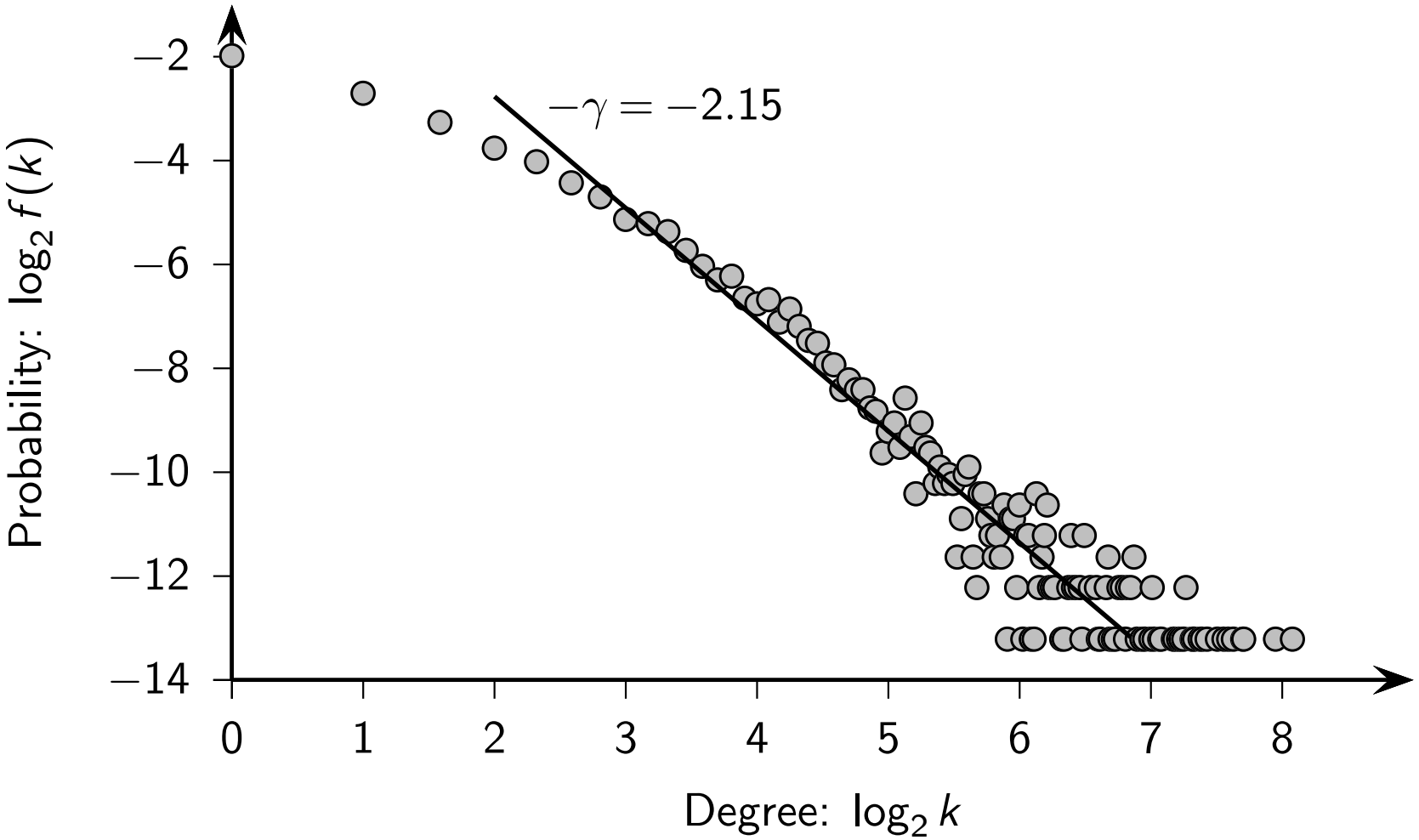
Let $C(k)$ denote the average clustering coefficient for all nodes with degree k; then the clustering effect also manifests itself as a power-law relationship between $C(k)$ and k:

$$C(k) \propto k^{-\gamma}$$

In other words, a log-log plot of k versus $C(k)$ exhibits a straight line behavior with negative slope $-\gamma$.

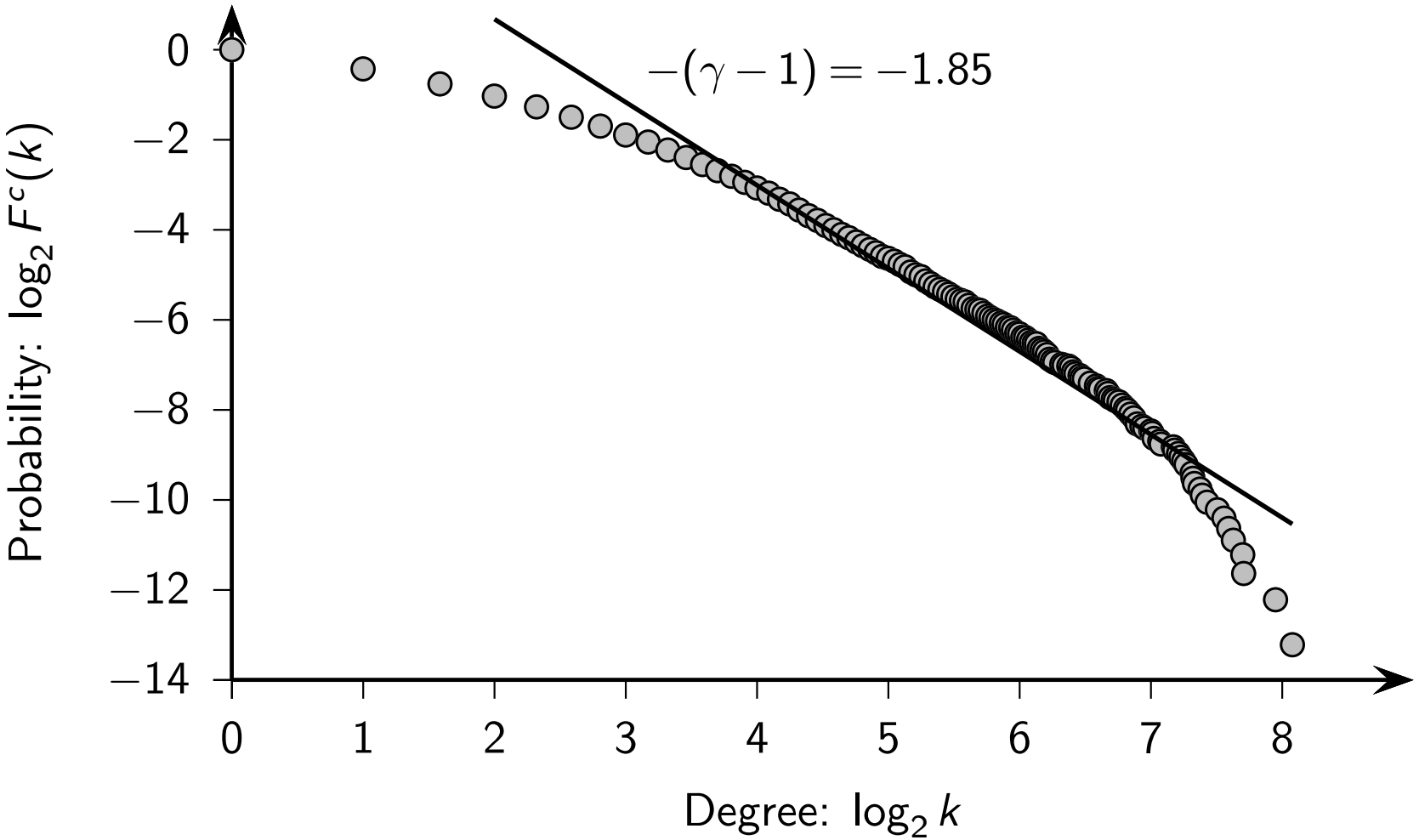
Degree Distribution: Human Protein Interaction Network

$|V| = n = 9521$, $|E| = m = 37060$

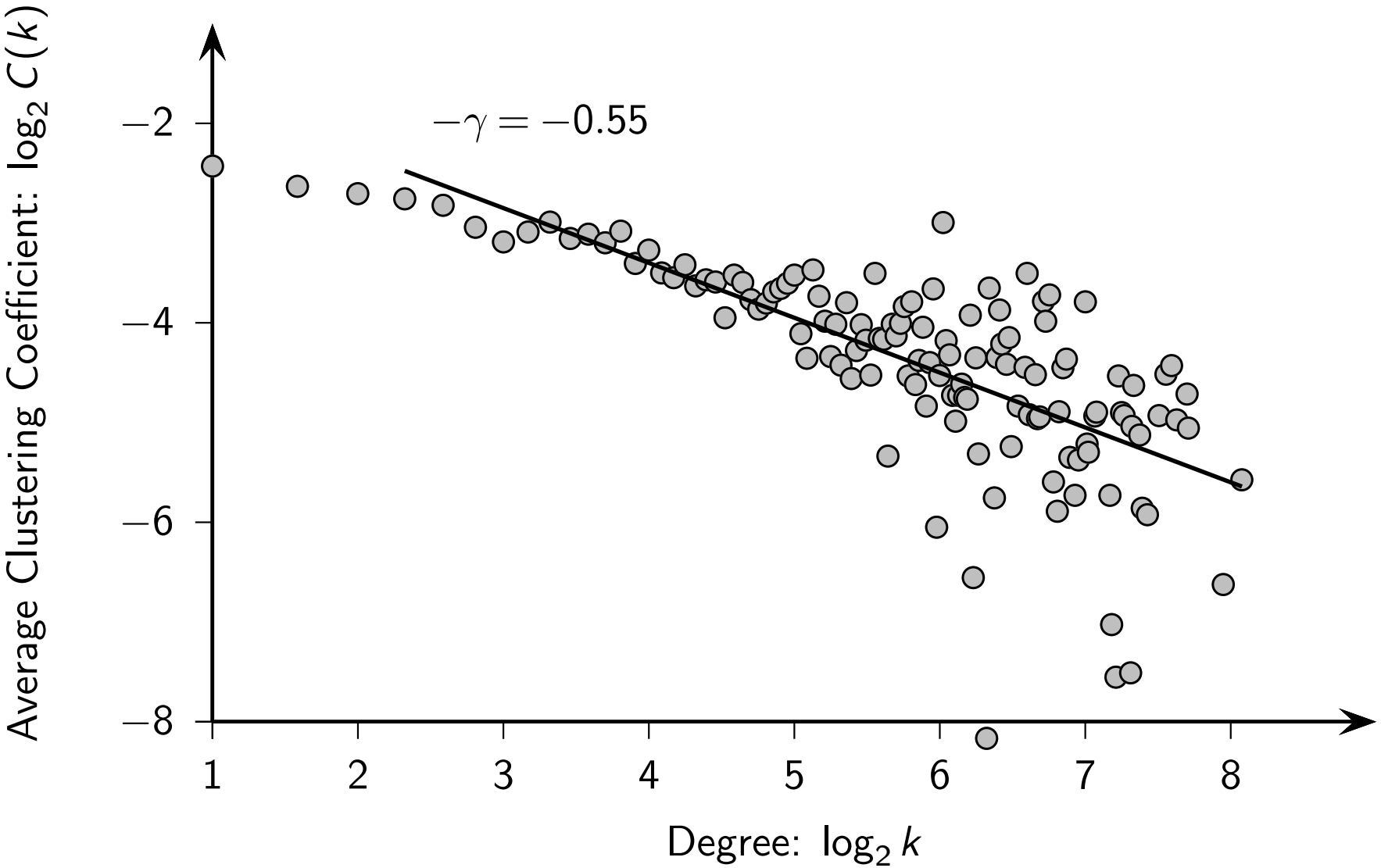


Cumulative Degree Distribution

$F^c(k) = 1 - F(k)$ where $F(k)$ is the CDF for $f(k)$



Average Clustering Coefficient



Erdös–Rényi Random Graph Model

The ER model specifies a collection of graphs $\mathcal{G}(n, m)$ with n nodes and m edges, such that each graph $G \in \mathcal{G}$ has equal probability of being selected:

$$P(G) = \frac{1}{\binom{M}{m}} = \binom{M}{m}^{-1}$$

where $M = \binom{n}{2} = \frac{n(n-1)}{2}$ and $\binom{M}{m}$ is the number of possible graphs with m edges (with n nodes).

Random Graph Generation: Randomly select two distinct vertices $v_i, v_j \in V$, and add an edge (v_i, v_j) to E, provided the edge is not already in the graph G. Repeat the process until exactly m edges have been added to the graph.

Let X be a random variable denoting the degree of a node for $G \in \mathcal{G}$. Let p denote the probability of an edge in G

$$p = \frac{m}{M} = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)}$$

Random Graphs: Average Degree

Degree of a node follows a binomial distribution with probability of success p, given as

$$f(k) = P(X = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

since a node can be connected to $n-1$ other vertices.

The average degree μ_d is then given as the expected value of X:

$$\mu_d = E[X] = (n-1)p$$

The variance of the degree is

$$\sigma_d^2 = var(X) = (n-1)p(1-p)$$

Random Graphs: Degree Distribution

As $n \to \infty$ and $p \to 0$ the expected value and variance of X can be rewritten as

$$E[X] = (n-1)p \simeq np \text{ as } n \to \infty$$
$$var(X) = (n-1)p(1-p) \simeq np \text{ as } n \to \infty \text{ and } p \to 0$$

The binomial distribution can be approximated by a Poisson distribution with parameter λ, given as

$$f(k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

where $\lambda = np$ represents both the expected value and variance of the distribution.

Thus, ER random graphs do not exhibit power law degree distribution.

Random Graphs: Clustering Coefficient and Diameter

Clustering Coefficient: Consider a node v_i with degree k. Since p is the probability of an edge, the expected number of edges m_i among the neighbors of a node v_i is simply

$$m_i = \frac{pk(k-1)}{2}$$

The clustering coefficient is

$$C(v_i) = \frac{2m_i}{k(k-1)} = p$$

which implies that $C(G) = \frac{1}{n}\sum_i C(v_i) = p$. Since for sparse graphs we have $p \to 0$, this means that ER random graphs do not show clustering effect.

Diameter: Expected degree of a node is $\mu_d = \lambda$, so in one hop a node can reach λ nodes. Coarsely, in k hops it can reach λ^k nodes. Thus, we have

$$\sum_{k=1}^{t} \lambda^k \leq n, \text{ which implies that } t = \log_\lambda n$$

It follows that the diameter of the graph is

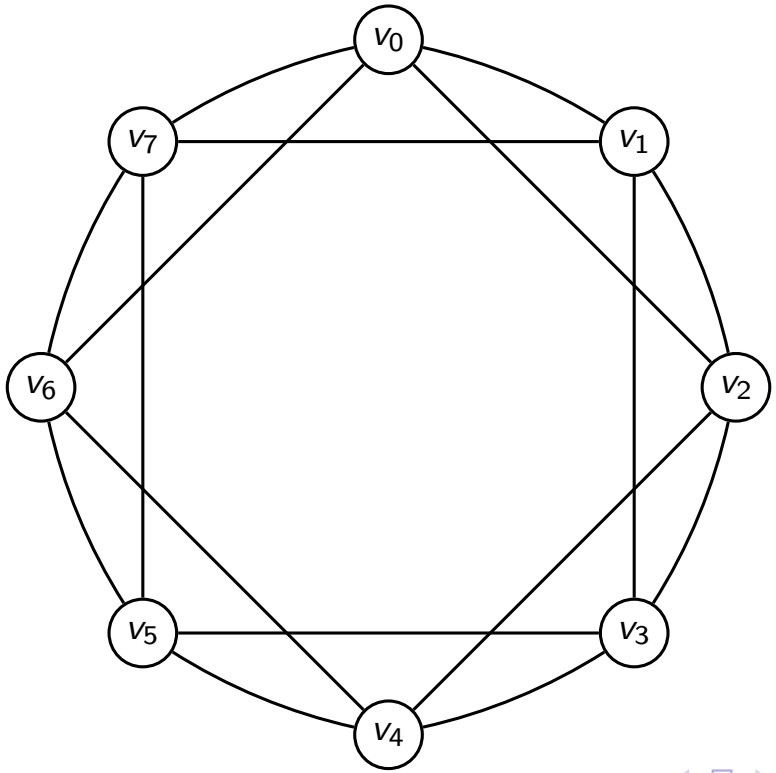
$$d(G) \propto \log n$$

Thus, ER random graphs are small-world.

Watts–Strogatz Small-world Graph Model

The Watts–Strogatz (WS) model starts with a regular graph of degree $2k$, having n nodes arranged in a circular layout, with each node having edges to its k neighbors on the right and left.
The regular graph has high local clustering. Adding a small amount of randomness leads to the emergence of the small-world phenomena.

Watts–Strogatz Regular Graph: $n = 8,\ k = 2$



WS Regular Graph: Clustering Coefficient and Diameter

The clustering coefficient of a node v is given as

$$C(v) = \frac{m_v}{M_v} = \frac{3k-3}{4k-2}$$

As k increases, the clustering coefficient approaches $\frac{3}{4}$ because $C(G) = C(v) \to \frac{3}{4}$ as $k \to \infty$. The WS regular graph thus has a high clustering coefficient.

The diameter of a regular WS graph is given as

$$d(G) = \begin{cases} \lceil \frac{n}{2k} \rceil & \text{if } n \text{ is even} \\ \lceil \frac{n-1}{2k} \rceil & \text{if } n \text{ is odd} \end{cases}$$

The regular graph has a diameter that scales linearly in the number of nodes, and thus it is not small-world.

Random Perturbation of Regular Graph

Edge Rewiring: For each edge (u, v) in the graph, with probability r, replace v with another randomly chosen node avoiding loops and duplicate edges.

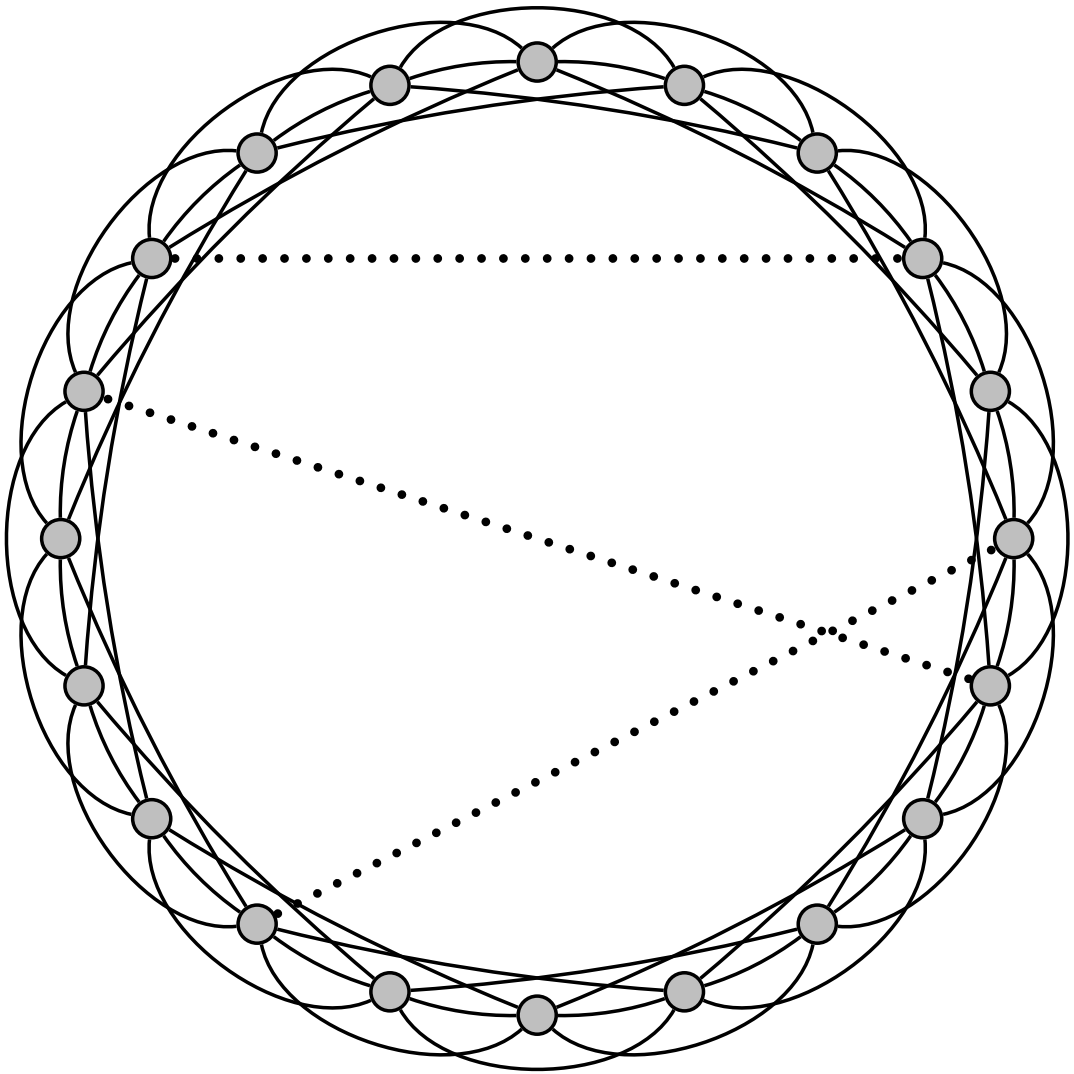
The WS regular graph has $m = kn$ total edges, so after rewiring, rm of the edges are random, and $(1 - r)m$ are regular.

Edge Shortcuts: Add a few *shortcut* edges between random pairs of nodes, with r being the probability, per edge, of adding a shortcut edge.

The total number of random shortcut edges added to the network is $mr = knr$. The total number of edges in the graph is $m + mr = (1 + r)m = (1 + r)kn$.

Watts–Strogatz Graph: Shortcut Edges

$n = 20,\ k = 3$



Properties of Watts–Strogatz Graphs

Degree Distribution: Let X denote the random variable denoting the number of shortcuts for each node. Then the probability of a node with j shortcut edges is given as

$$f(j) = P(X = j) = \binom{n'}{j} p^j (1-p)^{n'-j}$$

with $E[X] = n'p = 2kr$ and $p = \frac{2kr}{n-2k-1} = \frac{2kr}{n'}$.

The expected degree of each node in the network is therefore $2k + E[X] = 2k(1+r)$. The degree distribution is not a power law.

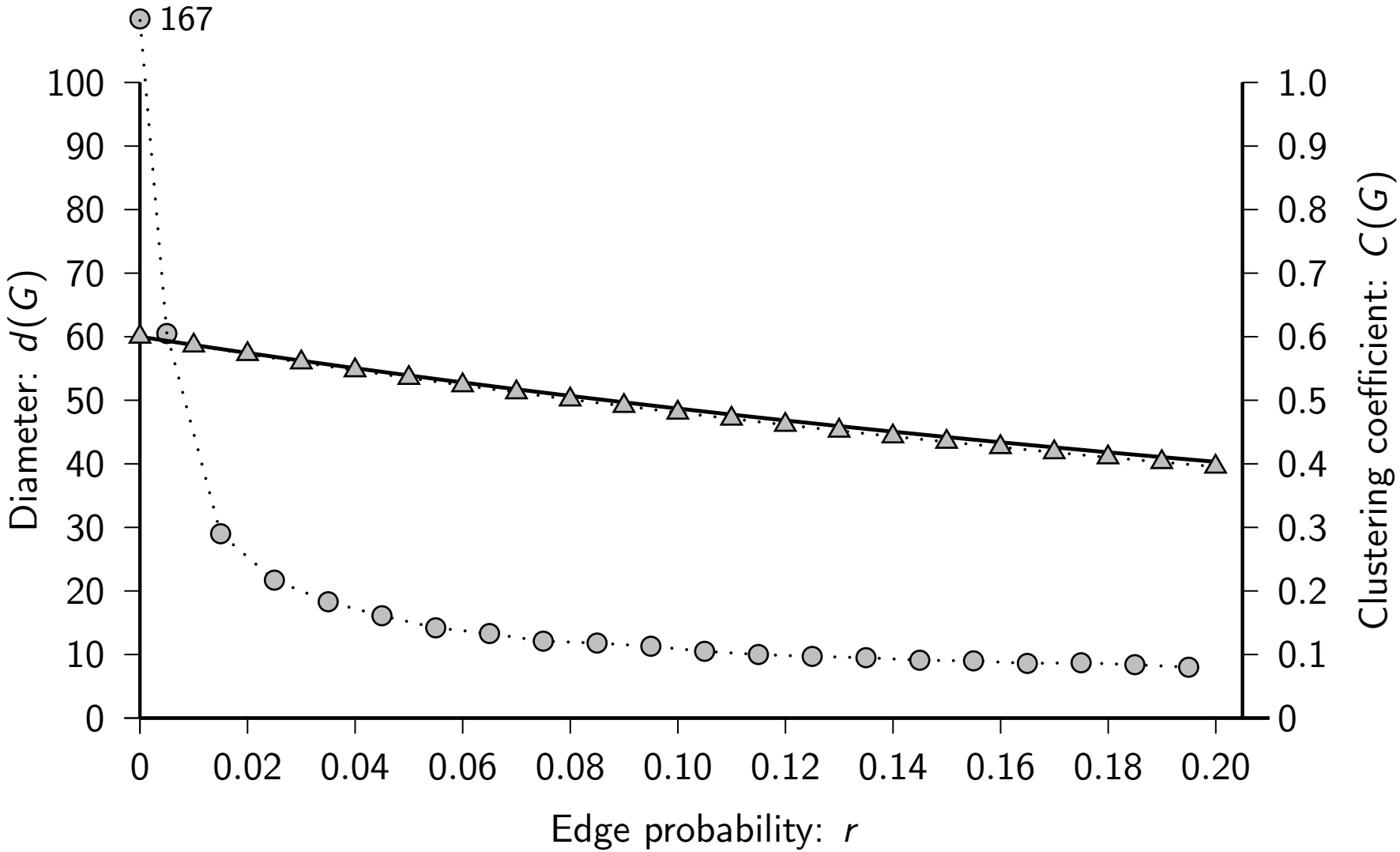
Clustering Coefficient: The clustering coefficient is

$$C(v) \simeq \frac{3(k-1)}{(1+r)(4kr+2(2k-1))} = \frac{3k-3}{4k-2+2r(2kr+4k-1)}$$

Thus, for small values of r the clustering coefficient remains high.

Diameter: Small values of shortcut edge probability r are enough to reduce the diameter from $O(n)$ to $O(\log n)$.

Watts-Strogatz Model: Diameter (circles) and Clustering Coefficient (triangles)



Barabási–Albert Scale-free Model

The Barabási–Albert (BA) yields a scale-free degree distribution based on *preferential attachment*; that is, edges from the new vertex are more likely to link to nodes with higher degrees.

Let G_t denote the graph at time t, and let n_t denote the number of nodes, and m_t the number of edges in G_t.

Initialization: The BA model starts with G_0, with each node connected to its left and right neighbors in a circular layout. Thus $m_0 = n_0$.

Growth and Preferential Attachment: The BA model derives a new graph G_{t+1} from G_t by adding exactly one new node u and adding $q \leq n_0$ new edges from u to q distinct nodes $v_j \in G_t$, where node v_j is chosen with probability $\pi_t(v_j)$ proportional to its degree in G_t, given as

$$\pi_t(v_j) = \frac{d_j}{\sum_{v_i \in G_t} d_i}$$

Barabási–Albert Graph

$n_0 = 3,\ q = 2,\ t = 12$

At $t = 0$, start with 3 vertices v_0, v_1, and v_2 fully connected (shown in gray).

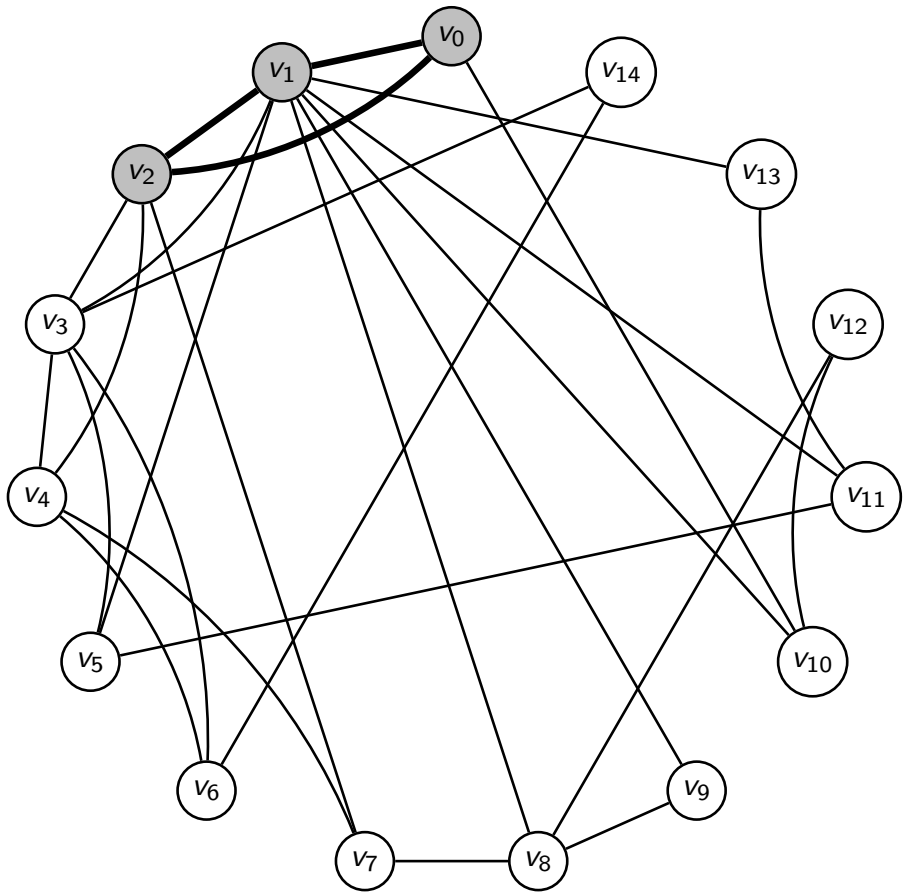
At $t = 1$, vertex v_3 is added, with edges to v_1 and v_2, chosen according to the distribution

$$\pi_0(v_i) = 1/3 \text{ for } i = 0, 1, 2$$

At $t = 2$, v_4 is added. Nodes v_2 and v_3 are preferentially chosen according to the probability distribution

$$\pi_1(v_0) = \pi_1(v_3) = \frac{2}{10} = 0.2$$

$$\pi_1(v_1) = \pi_1(v_2) = \frac{3}{10} = 0.3$$



Properties of the BA Graphs

Degree Distribution: The degree distribution for BA graphs is given as

$$f(k) = \frac{(q+2)(q+1)q}{(k+2)(k+1)k} \cdot \frac{2}{(q+2)} = \frac{2q(q+1)}{k(k+1)(k+2)}$$

For constant q and large k, the degree distribution scales as

$$f(k) \propto k^{-3}$$

The BA model yields a power-law degree distribution with $\gamma = 3$, especially for large degrees.

Diameter: The diameter of BA graphs scales as

$$d(G_t) = O\left(\frac{\log n_t}{\log \log n_t}\right)$$

suggesting that they exhibit *ultra-small-world* behavior, when $q > 1$.

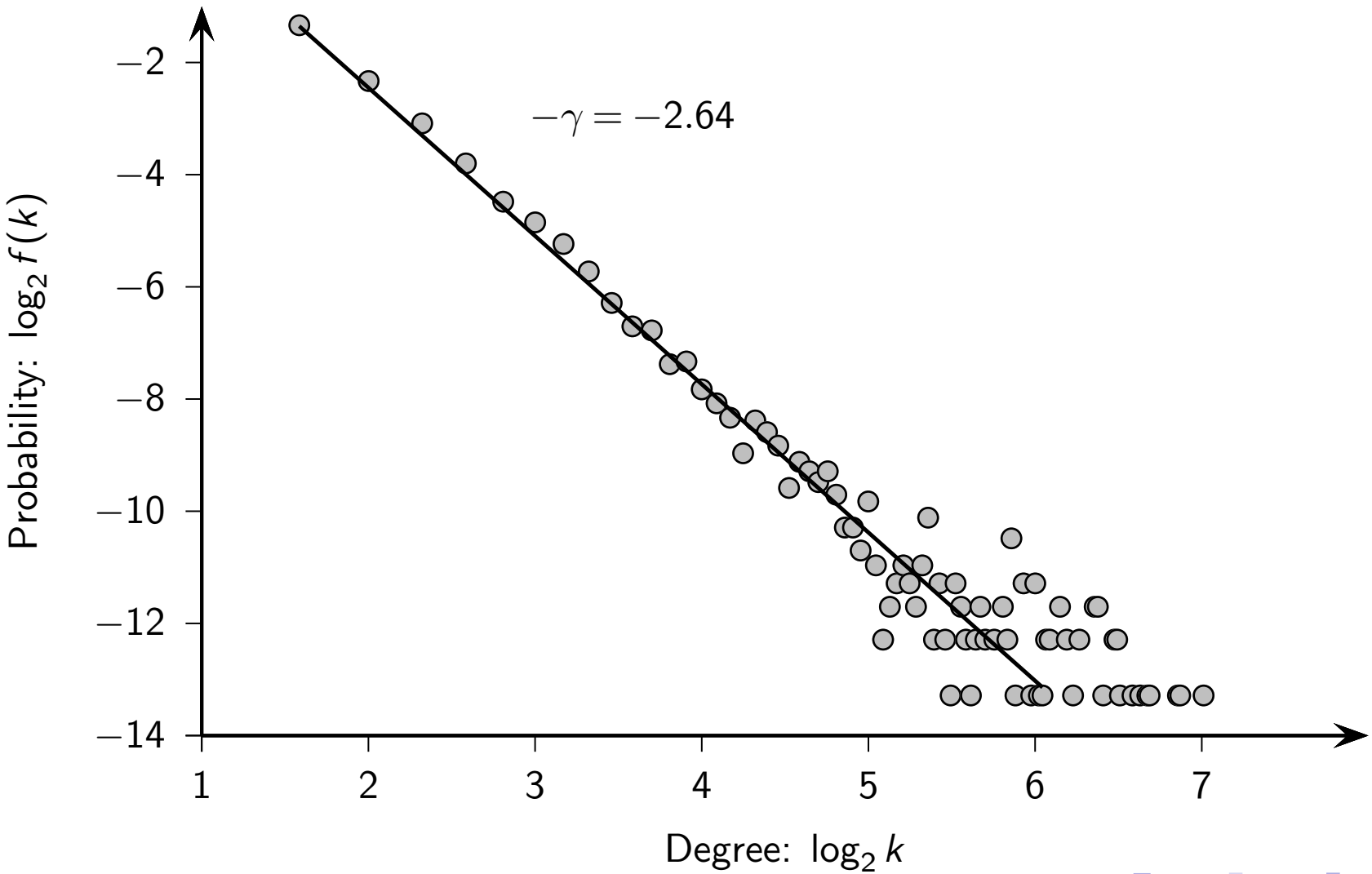
Clustering Coefficient: The expected clustering coefficient of the BA graphs scales as

$$E[C(G_t)] = O\left(\frac{(\log n_t)^2}{n_t}\right)$$

which is only slightly better than for random graphs.

Barabási–Albert Model: Degree Distribution

$n_0 = 3, t = 997, q = 3$



Data Mining and Machine Learning: Fundamental Concepts and Algorithms

dataminingbook.info

Mohammed J. Zaki[1] Wagner Meira Jr.[2]

[1]Department of Computer Science
Rensselaer Polytechnic Institute, Troy, NY, USA

[2]Department of Computer Science
Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Chapter 4: Graph Data