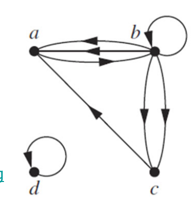
Graph Types

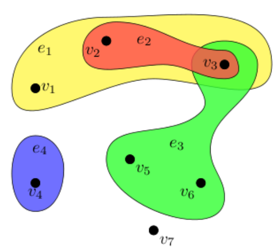
Peter M. Kogge

Types of Graphs

- **Graphs**: Sets (V,E) where E= {(u,v)}
  - Undirected: (u,v) = (v,u)
  - Directed: (u,v) != (v,u)
- **Networks**: Graphs with “weights”
- **Multi-graphs**: multiple edges permitted between same two vertices
  - https://en.wikipedia.org/wiki/Multigraph#/media/File:Multi-pseudograph.svg



- **Hyper-Graphs**: edges connect >2 vertices
  - k-uniform: all edges connect k vertices



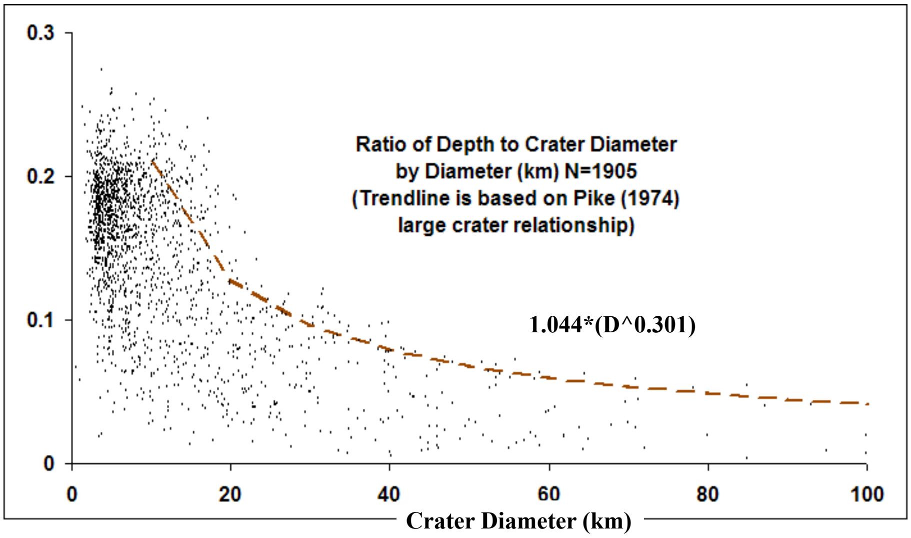
https://i.stack.imgur.com/htqk5.png

How to Characterize Graphs

- # of vertices; # of edges
- Vertex Degree: average, max, distribution

Power Law

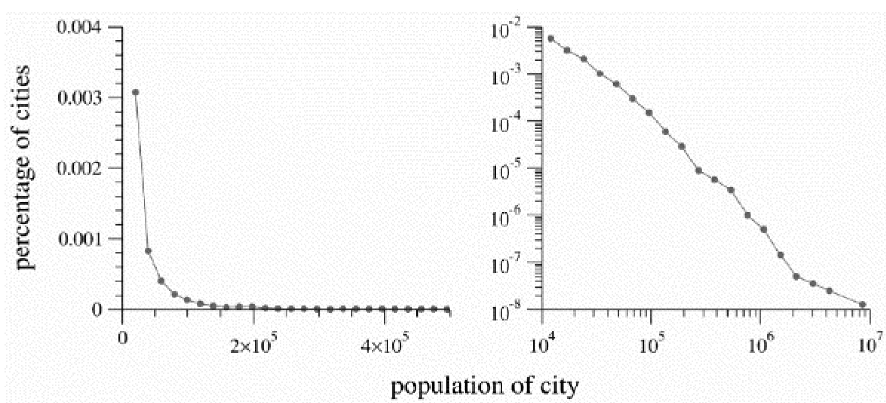
- **Power Law**: If $y = f(x)$ and
  - A change in one variable causes a proportional change in other, *independent* of initial value
  - i.e. y varies as a power of x (& thus v.v.)
  - E.g. area of a square: 2X length => 4X area
- Equivalent form: $y \approx ax^{-\gamma}$, a, $-\gamma$ constants
- **Scale Invariance**: $f(cx) = a(cx)^{-\gamma} = c^{-\gamma} f(x)$
  - scaling x by a constant factor scales y by a constant
- Power Law functions are **scale invariant**
- Plotting on log-log gives straight line



http://fisherka.csolutionshosting.net/astronote/plan/craterdepth/img/RatioDepDia.png

"Scale-free" means changing from km to miles doesn't change shape of curve

Histogram of U.S. City Population

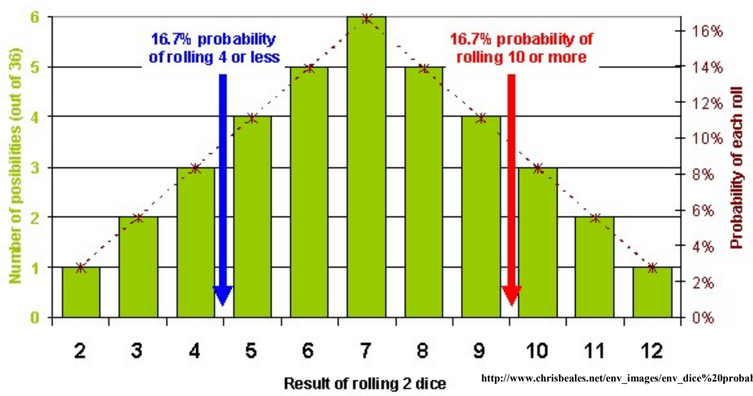


https://www.tandfonline.com/na101/home/literatum/publisher/tandf/journals/content/tcph20/2005/tcph20.v046.i05/00107510500052444/production/images/large/tcph_a_10062260_o_fig002g.jpeg

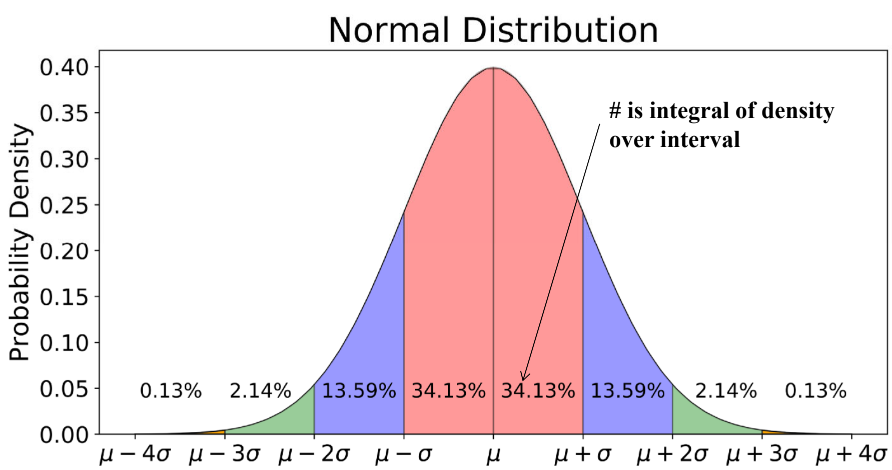
Probability Density (Probability Mass)

- function of a variable
  - whose integral over any interval
  - is probability that variable will lie within that interval.

36 possible results from rolling 2 dice



http://www.chrisbeales.net/env_images/env_dice%20probability%20graph.jpg



https://cdn-images-1.medium.com/max/1600/1*IdGgdrY_n_9_YfkaCh-dag.png

μ = mean
σ = std dev

Power Law Distributions

- If p(x) dx is prob. of some event having a value from x to x+dx
- Then if distribution is a power law
- Then on log log curve it's a straight line
- Then $\log(p(x)) \approx -\gamma * \log(x) + c$
- or $p(x) \approx Cx^{-\gamma}$, $C = e^c$
- These are distributions, thus $-\gamma$ negative
- Values on right-hand side are small

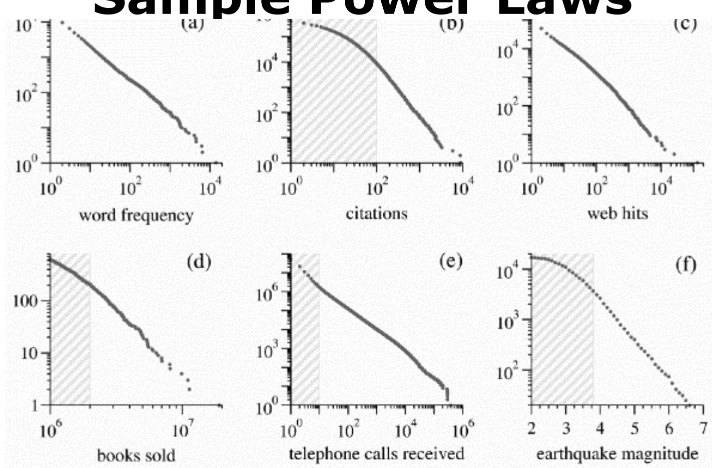
Cumulative Distributions of Power Law Distributions

- $P(x>z) = \int_z^\infty p(x)$
- If $p(x) = Cx^{-\gamma}$, then $P(x>z) = (C/(\gamma-1))x^{(-\gamma+1)}$
- Again a straight line on log-log graph
  - But different slope

Looking for Power Laws in Real Data

- Notional approach
  - Compute histogram of data, with constant bin size
  - Graph on log-log curve
  - Measure slope & y-intercept
- Problem:
  - With constant bins, righthand cases have few events
  - Result: very "noisy"
- Better: use logarithmic binning:
  - Intervals get bigger when moving right
  - E.g. have intervals grow by some factor at each step
- Best: compute cumulative distribution and graph on log-log
  - Slope is -γ+1 rather than γ
  - https://www.tandfonline.com/doi/full/10.1080/00107510500052444?scroll=top&needAccess=true

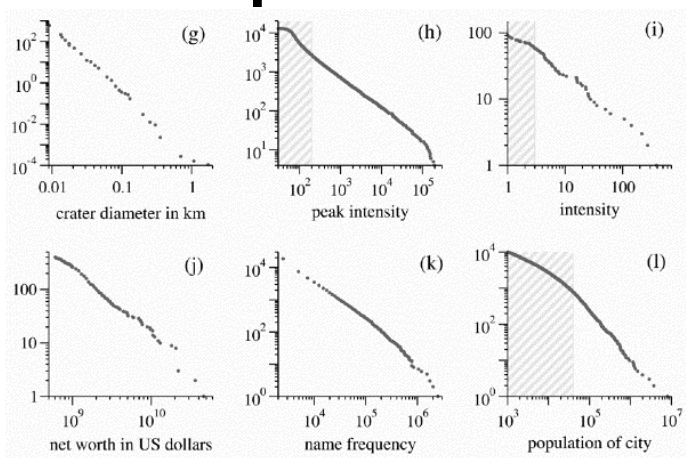
Sample Power Laws



(a) Numbers of occurrences of words in the novel *Moby Dick* by Hermann Melville.
(b) Numbers of citations to scientific papers published in 1981, from time of publication until June 1997.
(c) Numbers of hits on web sites by 60000 users of the America Online Internet service for the day of 1 December 1997
(d) Numbers of copies of bestselling books sold in the US between 1895 and 1965.
(e) Number of calls received by AT&T telephone customers in the US for a single day.
(f) Magnitude of earthquakes in California between January 1910 and May 1992. .

https://www.tandfonline.com/na101/home/literatum/publisher/tandf/journals/content/tcph20/2005/tcph20.v046.i05/00107510500052444/production/images/large/tcph_a_10062260_o_fig004g.jpeg

More Sample Power Laws

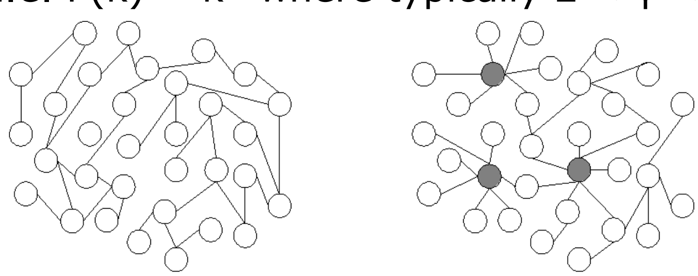


(g) Diameter of craters on the moon. Vertical axis is measured per square kilometre.
(*h*) Peak gamma-ray intensity of solar flares in counts/sec, measured from Earth orbit between Feb1980 and Nov1989.
(i) Intensity of wars from 1816 to 1980, measured as battle deaths per 10000 of population of participating countries.
(j) Aggregate net worth in dollars of the richest individuals in the US in October 2003.
(k) Frequency of occurrence of family names in the US in the year 1990.
(*l*) Populations of US cities in the year 2000.

https://www.tandfonline.com/na101/home/literatum/publisher/tandf/journals/content/tcph20/2005/tcph20.v046.i05/00107510500052444/production/images/large/tcph_a_10062260_o_fig004g.jpeg

Scale Free Graphs

- Degree distribution P(k) follows power law
  - P(k) = # of vertices with degree k
- I.e. $P(k) \approx k^{-\gamma}$ where typically $2 < \gamma < 3$



(a) Random network (b) Scale-free network

https://upload.wikimedia.org/wikipedia/commons/7/77/Scale-free_network_sample.png

Erdős–Rényi Graph Model

- 1959
- Goal: generate "Random Graphs"
- Assumption: all graphs with fixed # vertices and edges are equally likely
- **G(n, p) model**:
  - n vertices
  - Thus n(n-1)/2 possible edges
  - Probability of each edge being in a graph is p
  - Probability of any particular graph = $p^M(1-p)^{\binom{n}{2}-M}$
- Expected # edges is $\binom{n}{2}p$

Barabási–Albert Graph Model

- 1999
- Generate scale-free power law graphs using:
  - Assumption of Growth of # of vertices over time
  - **Preferential Attachment:** more connected a vertex is, more likely to receive more edges
- Algorithm:
  - Assume initially m_0 vertices
  - Add new vertices one at a time
  - Probability that new vertex connected to vertex v is
  - $p_v = k_v/\Sigma_j k_j$, where k_v is degree of v
- Resulting $P(k) = k^{-3}$

Kronecker Graphs

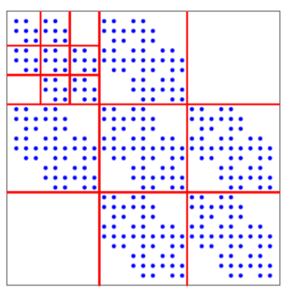
- Given real graph, generate a synthetic graphs with matching properties
- Algorithm: Build graph as adjacency matrix
  - Start with **initiator graph** K_1 with N_1 vertices, E_1 edges
  - Recursively build K_2, K_3, … where K_j has N_1^j vertices
  - At each step take Kronecker product of K_j with itself
- Resulting graphs maintain many properties of original

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} \doteq \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \dots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \dots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \dots & a_{n,m}\mathbf{B} \end{pmatrix}$$

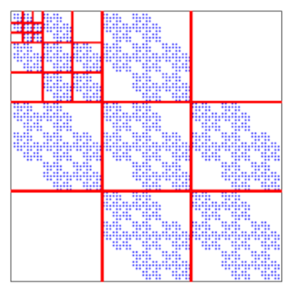
Examples

1	1	0
1	1	1
0	1	1

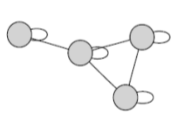
(d) Adjacency matrix of K_1



(a) K_3 adjacency matrix (27×27)



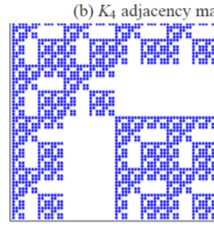
(b) K_4 adjacency matrix (81×81)



Initiator K_1

1	1	1	1
1	1	0	0
1	0	1	1
1	0	1	1

K_1 adjacency matrix

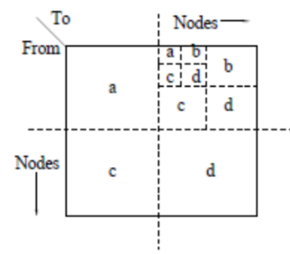


K_3 adjacency matrix

https://cs.stanford.edu/~jure/pubs/kronecker-jmlr10.pdf

R-MAT Graph Generators

- Used in Grap500 graph generation
  - See D. Chakrabarti, Y. Zhan, and C. Faloutsos, R-MAT: A recursive model for graph mining, SIAM Data Mining 2004
- Algorithm: construct Adjacency Matrix
  - Recursively divide matrix into 4 submatric labelled a,b,c,d
  - Each submatrix has a probability
  - To add an edge, select source (and target vertex by
    - Use probabilities to select submatrix of full matri
    - Use probabilities to select submatrix of submatrix
    - …
    - Stop on 1x1 submatrix
- To smooth out fluctuations in degree distribution, add some "noise" at each step



p(a) = 0.57
p(b)=p(c) = 0.19
p(d) = 0.05

Representing Sparse Adjacency Matrices

- Adjacency matrix for N vertex graph is N^2
  - One row/column per vertex
  - E.g. Graph500 graphs have 4 billion vertices
  - That's 1.6×10^{19} elements
- Most real adjacency matrices VERY sparse
  - # of 1's per row = degree of that vertex
  - Graph500 have on average 32 1's per row
- Would rather not store $O(N^2)$ elements when all but $O(N)$ are zero

Common Approaches

- **Dictionary of Keys**
  - Each non-zero recorded as (r,c,v) pair, in random order
  - Good for generating edge set dynamically
  - But slow when need to iterate
    - E.g. step through edges leaving/arriving at some vertex
- **Coordinate List**:
  - Again (r,c,v) pairs but sorted first by row then by column
  - Improved random access time
- **List of Lists**:
  - One list per row, with (column, value) as element
  - Typically list is sorted by column number
  - Good for accessing by row, bad if by column

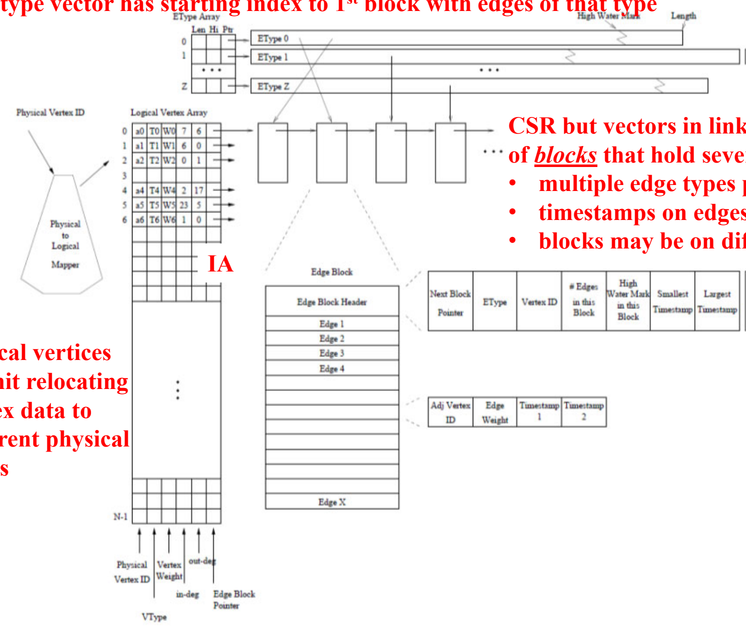
For adjacency matrices "value" = 1

Common Approaches

- **Compressed Sparse Row (CSR, CRS)**:
  - Three 1D vectors
    - A: length NNZ (# of non-zeros) of values
      - Listed in order of matching column index
      - Non needed for adjacency matrices where all values = 1
    - AJ: length NNZ – column # for each non-zero
    - IA: length N+1 (N # of vertices) of indices into A
      - IA[i] points to 1st non-zero for row I
    - IA[i+1]-IA[I]-1 = # of non-zeros for row I
  - Fast access to individual rows of matrix
  - Slow access to columns
  - Expensive incremental adding of edges to graph
- **Compressed Sparse Column(CSC, CCS)**
  - Same as CSR but for columns rather than rows

Stinger

Etype vector has starting index to 1st block with edges of that type



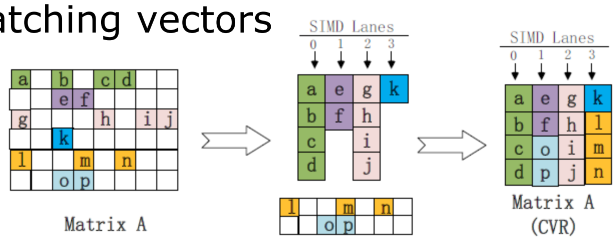
CSR but vectors in linked lists of *blocks* that hold several edges;
- **multiple edge types permitted**
- **timestamps on edges**
- **blocks may be on different nodes**

Logical vertices permit relocating vertex data to different physical nodes

23

Compressed Vector Representation (CVR)

- Many microprocessors have "short vector" SIMD instructions
  - If values are in consecutive memory/registers
  - Then one "SIMD" instruction can perform lots of ops
- CVR reformats matrix so elements in consecutive order
- When doing SpMV-like need to create matching vectors



http://delivery.acm.org/10.1145/3170000/3168818/cgo18-p58.pdf?ip=129.74.153.250&id=3168818&acc=ACTIVE%20SERVICE&key=EA62C54EFA59E1BA%2E367AD4ADD93F5D2F%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1539193774_ecb3de8c8c575e521adc8a4447bf2240

Graphs Types

24