Inferring How Networks Evolved

858L

Inferring network mechanisms: The Drosophila melanogaster *protein interaction network*

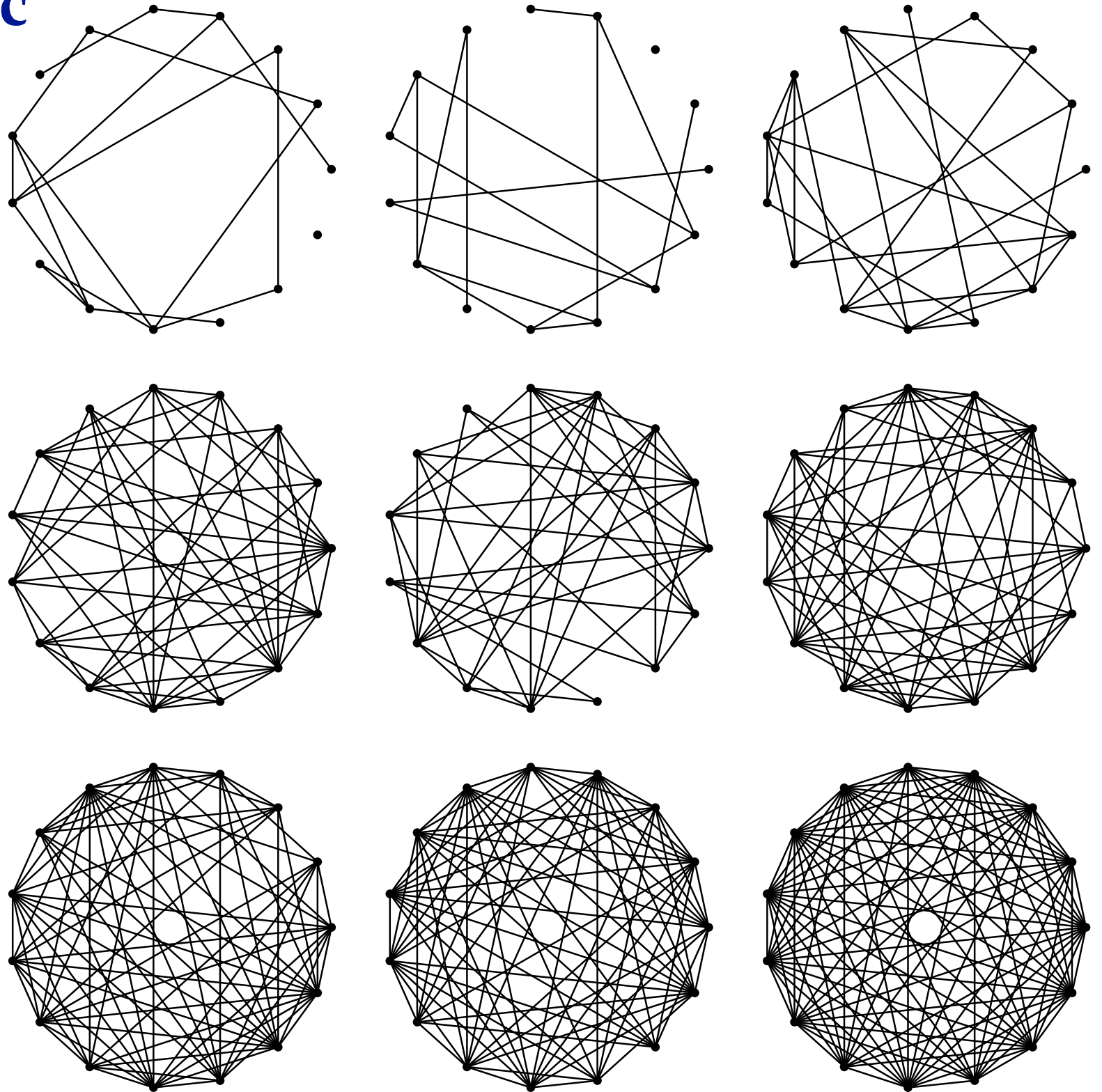Manuel Middendorf, Etay Ziv, and Chris H. Wiggins

RDS - Random Static

Erdos-Renyi (1960):
    Create *n* vertices
    Between every pair of
    vertices {u,v}, add an edge
    with probability *p*.
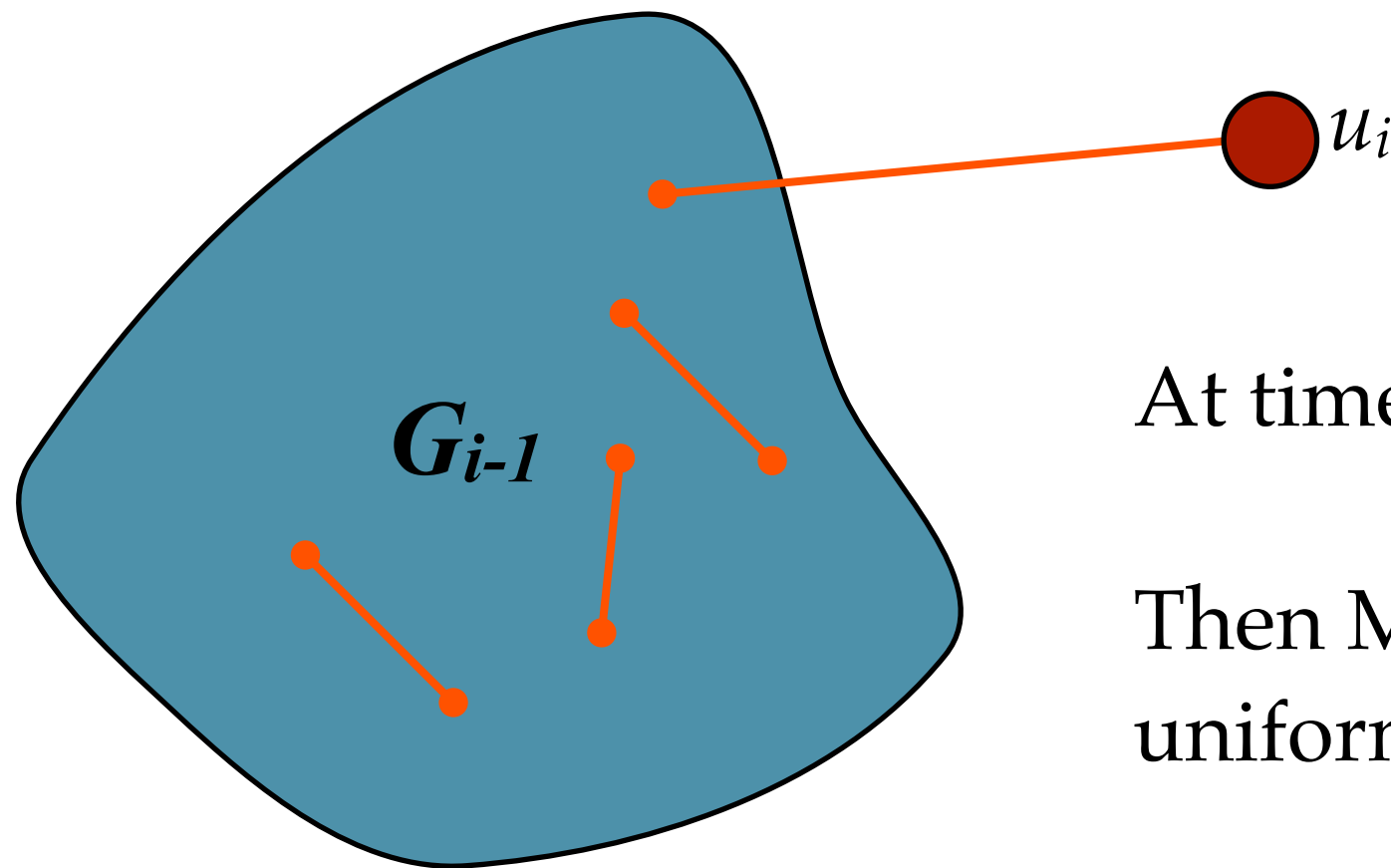
Expected degree is (*n*-1)*p*

ShowGraphArray[Partition[Table[RandomGraph[15, p], {p, 0.1, 0.9, 0.1}], 3]]

RDG - Random Growing

Parameters: n = number of nodes; M = expected # of edges.

G_{i-1}

u_i

At time i, a node is added.
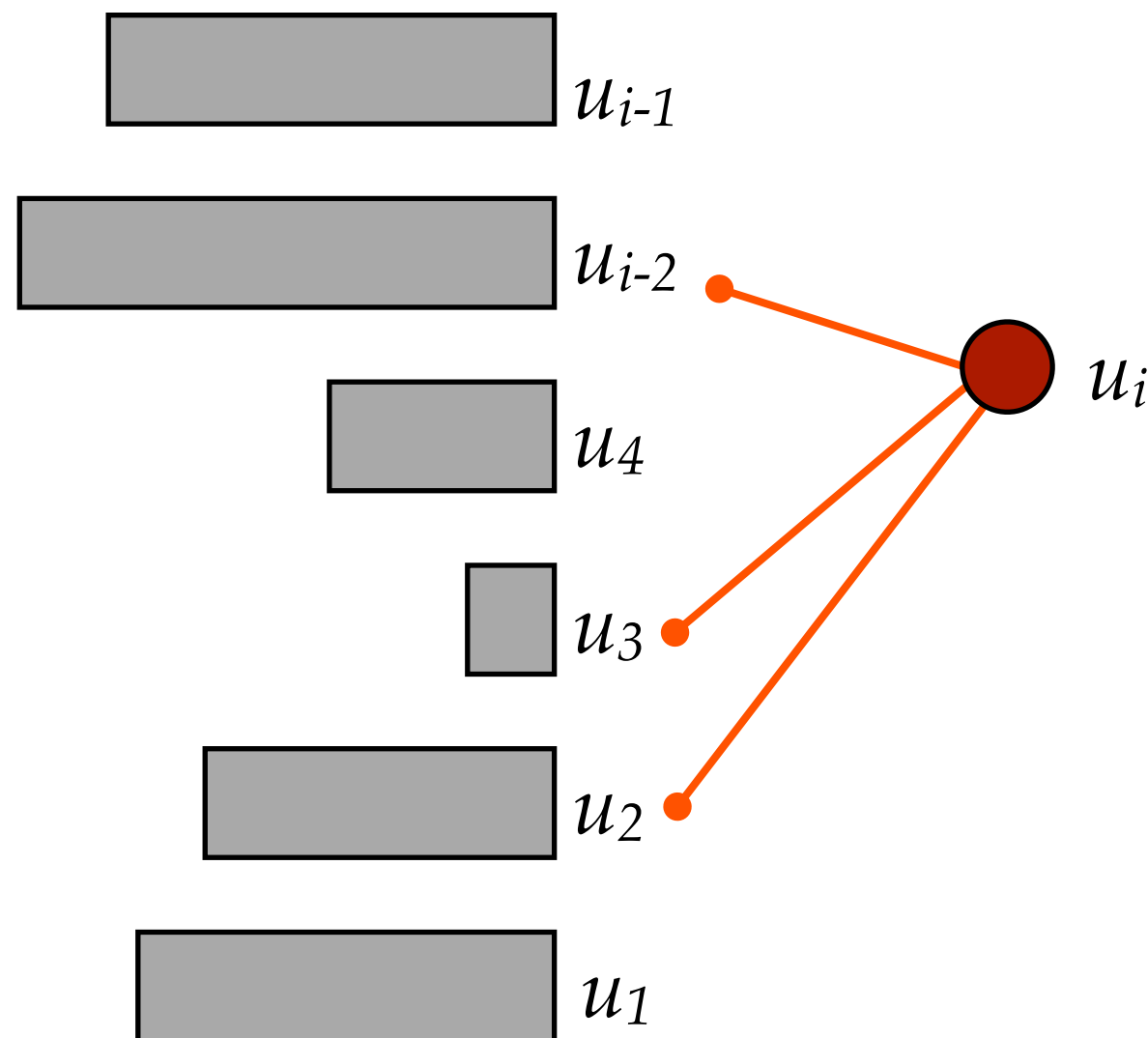
Then M/n edges are added uniformly at random.

Added edges might not involve u_i.

Nodes added earlier have more chances to be adjacent to edges.

LPA – Linear Preferential Attachment

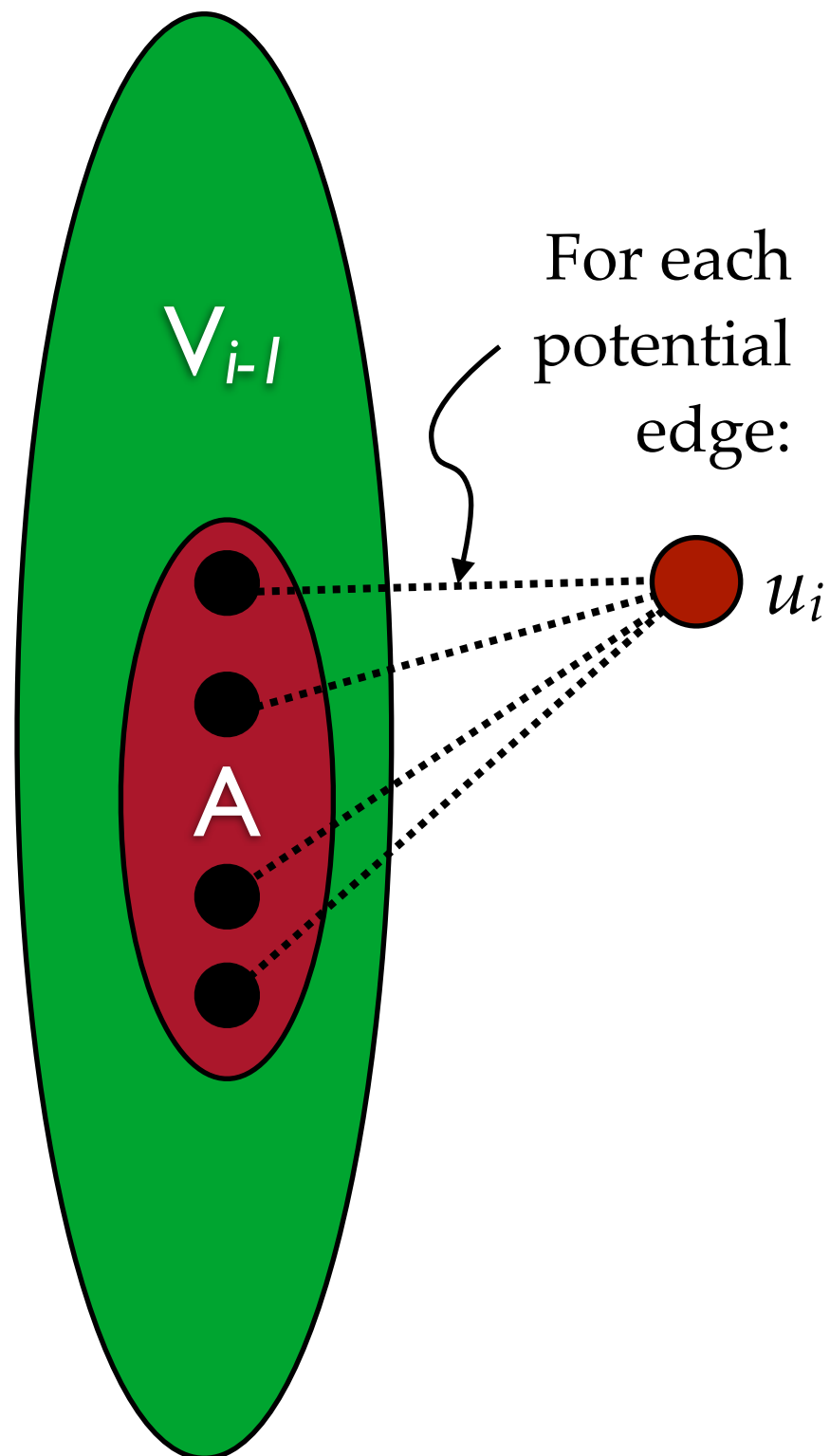**Parameters:** n = number of nodes; M = number of edges; a = smoothing parameter.

Histogram: degree(u_j) + a, normalized to be a probability distribution:

At time i, a node u_i is added.

M/n edges are added between u_i and the existing nodes drawn randomly according to the histogram.
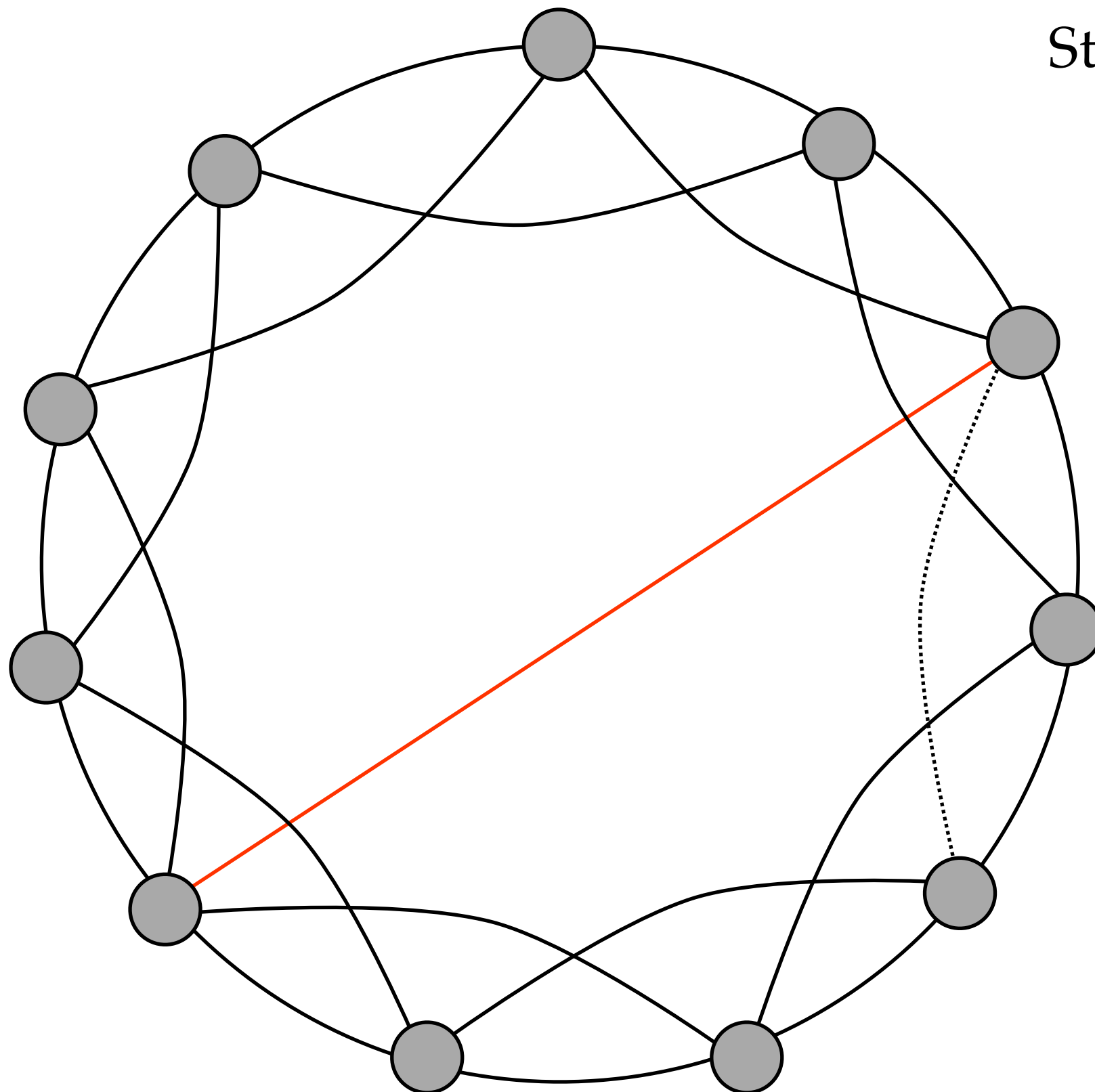
AGV – Aging Vertices

V_{i-1}

A

For each potential edge:

u_i

With probability μ, add the edge.

With probability (1-μ), add a random edge between u_i and any node in V_{i-1}, chosen according to the existing degrees.

Then: Pick a random node x in **A,** inversely proportional to the node degrees (i.e. prefer low-degree nodes)

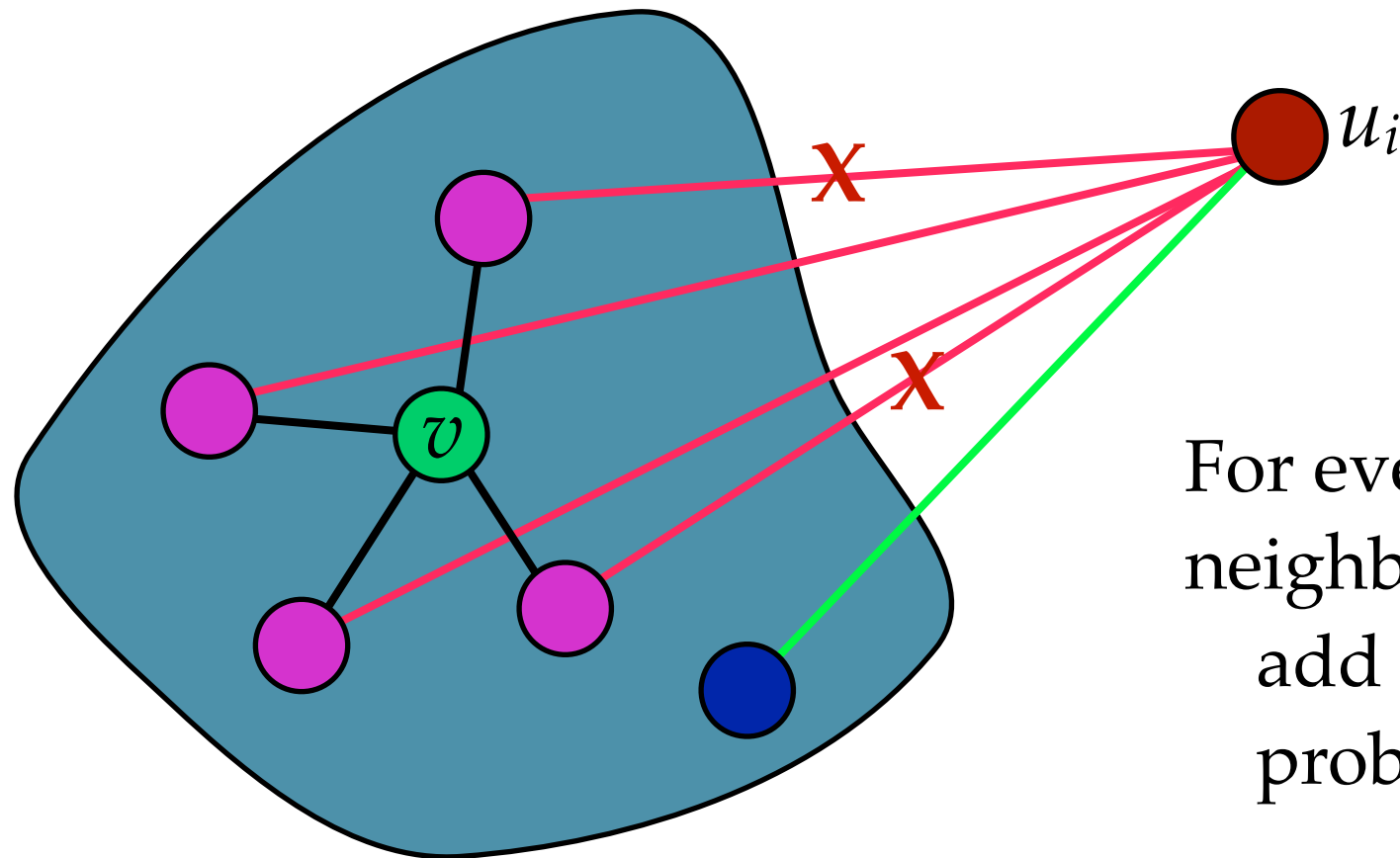Remove x from A and add u_i to A.

SMW – Small World Networks

Start with ring lattice

For each edge (i,j), in random order:
with probability q, rewire it to be an edge (i, v) for a random vertex v.

DMR – Duplication With Mutations

Choose a node v at random, and connect u_i to all the neighbors of v.
u_i is now a "clone" of v.

For each added edge, remove it with probability q_{del}

For every node w that is not now a
neighbor of u_i,
    add an edge (u_i, w) with
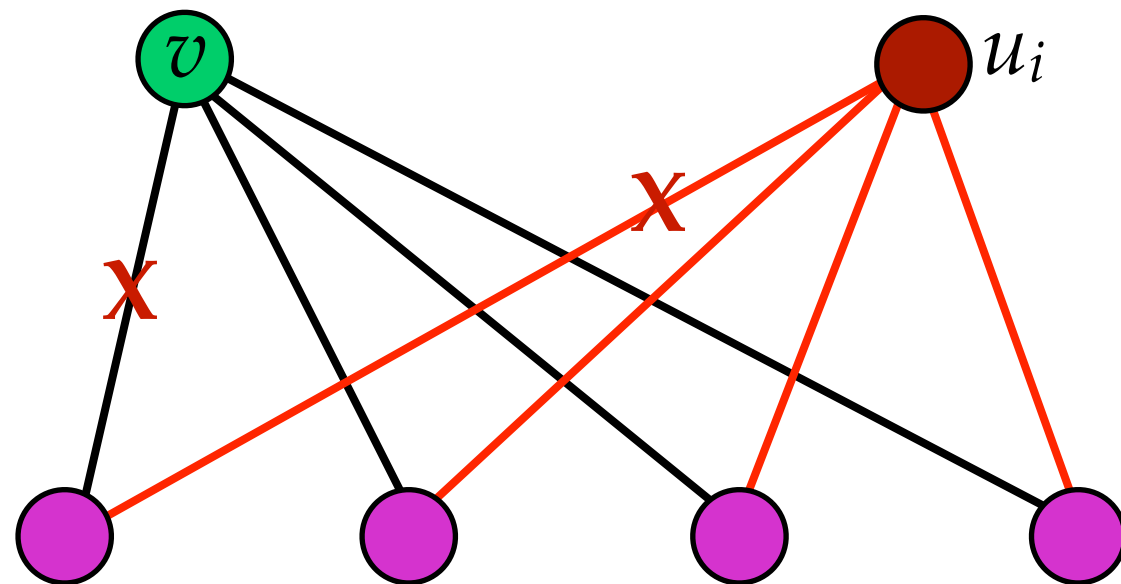   probability q_{new} / $(i-1)$

probability
decays over time.

DMC – Duplication, Mutation with Complementarity

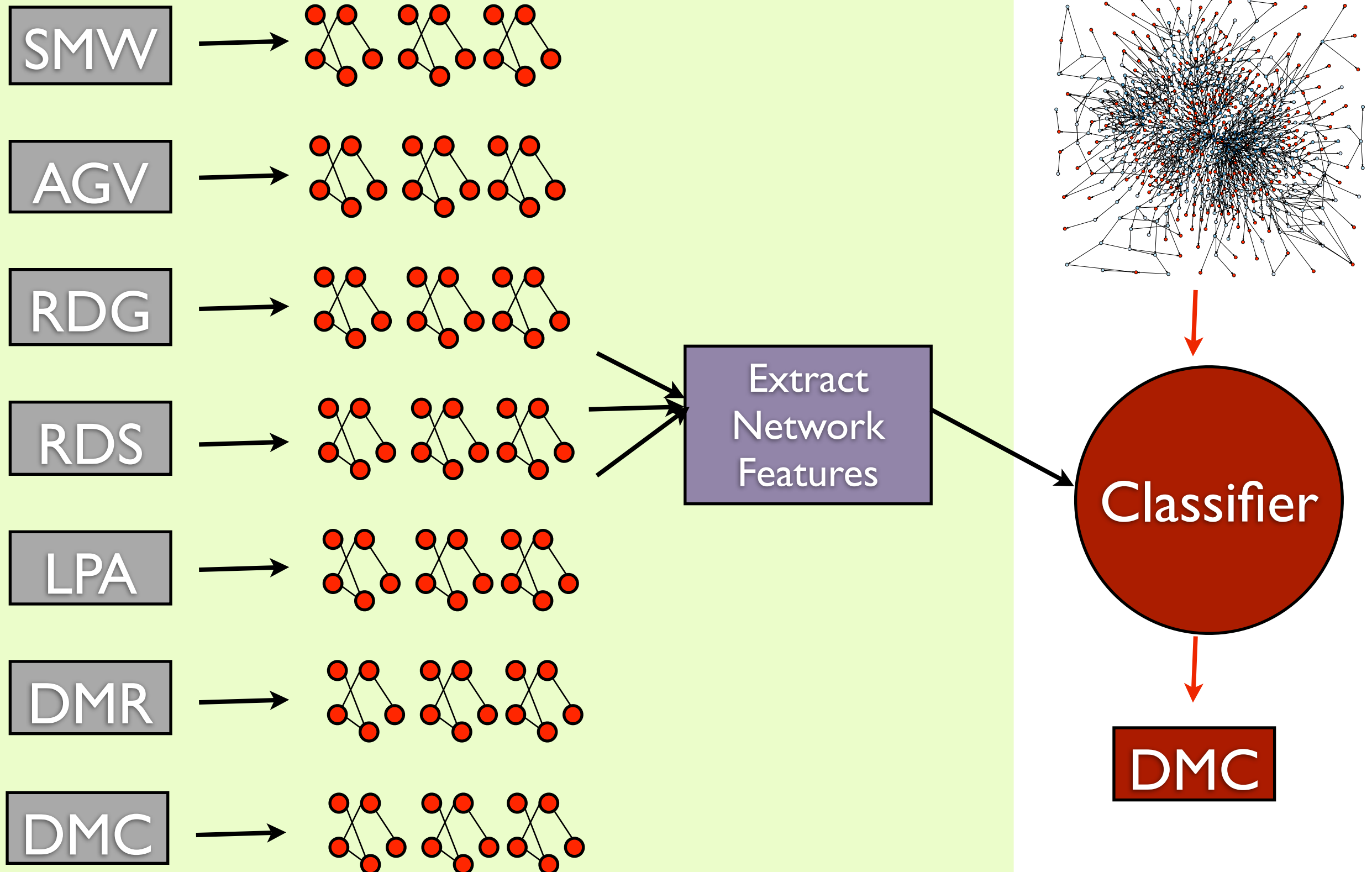Choose a node v at random, and connect u_i to all the neighbors of v. u_i is now a "clone" of v.

For every added edge, decide to delete with probability q_{del}

If you decide to delete, delete the **new or corresponding old** edge (choosing which one by flipping a coin):
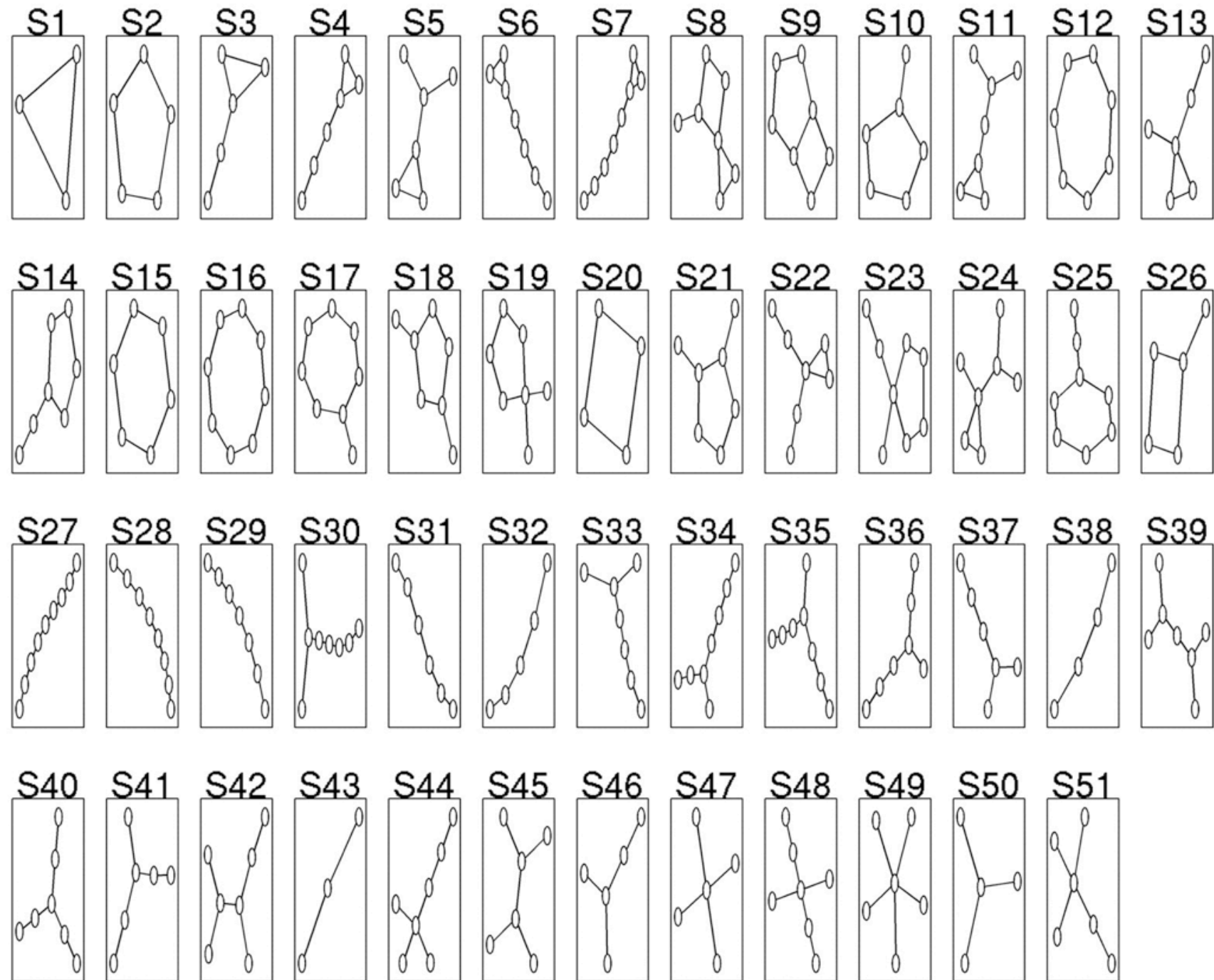

Finally, with probability q_{con}, add an edge (v, u_i)

Supervised Learning → Predict Network Models

Network Features – Subgraph Census

Count the occurrences of the following subgraphs:

Walks Of Length ≤ 8

    (148 possible graphs)

Subgraphs with ≤ 7 edges

    (130 possible graphs)

(Figure from Middendorf, 2004)

Classifier - Alternating Decision Trees

Trained via Boosting

To predict a class,
traverse the tree in the
following way:

Start a root node
Repeat until leaf:
    if current node is a decision node,
        move to the appropriate child based
        on the node's question
    else
        recursively move to **all** the children
        of the current node

(Figure from Middendorf, 2004)

Return vector of sums of all the weights visited.

Classifier - Alternating Decision Trees

Trained via Boosting

To predict a class,
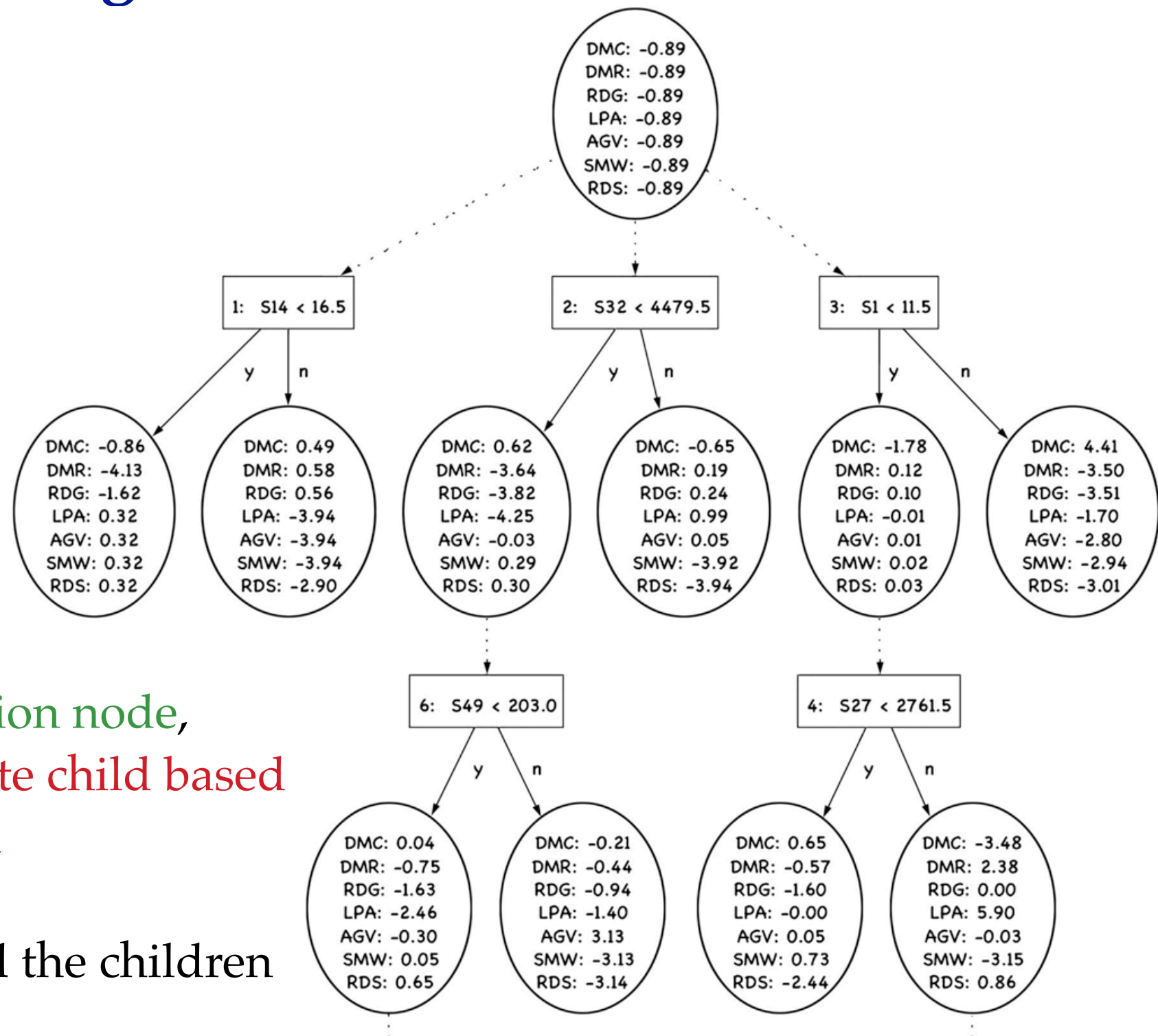traverse the tree in the
following way:
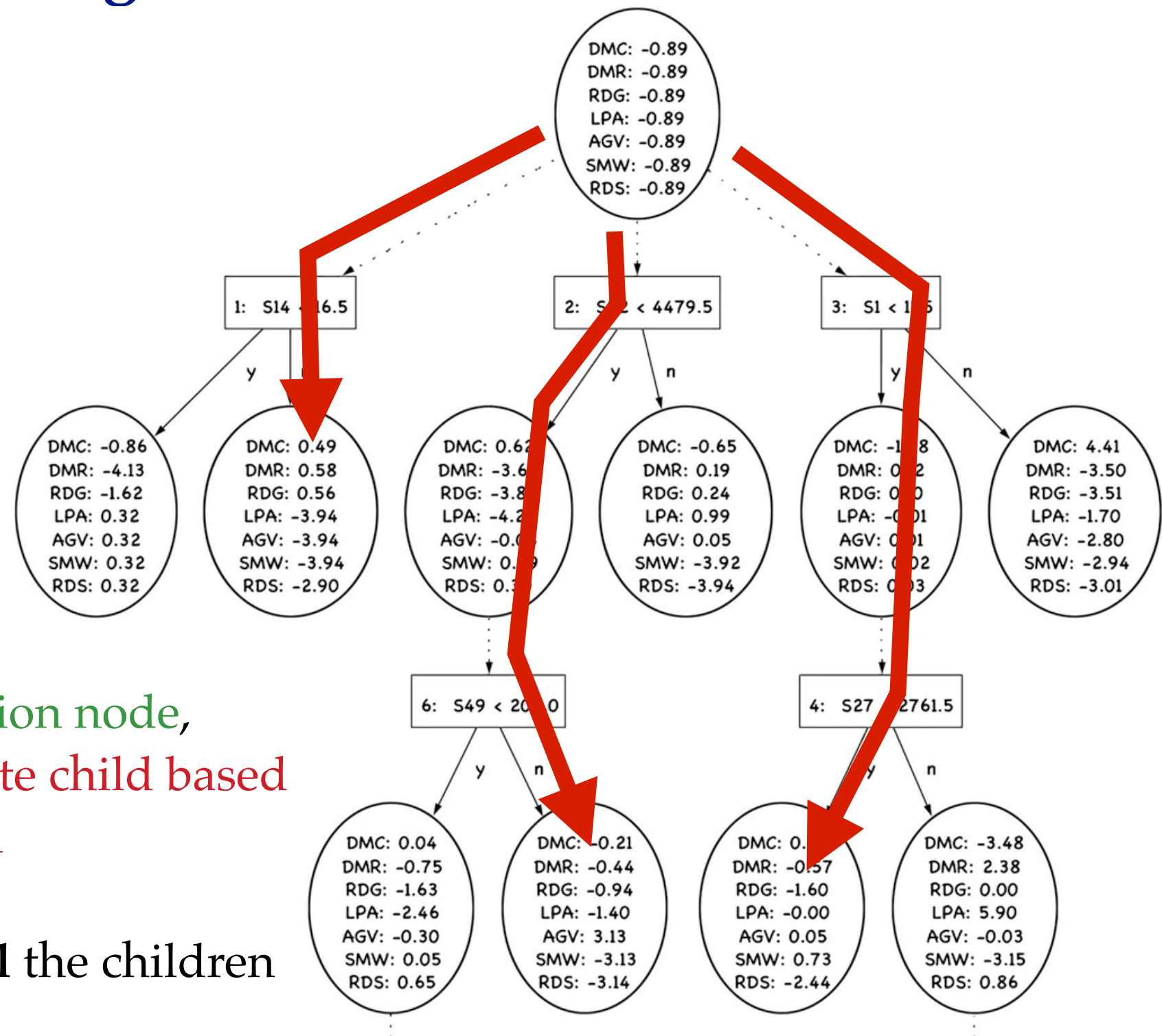
Start a root node
Repeat until leaf:
    if current node is a decision node,
        move to the appropriate child based
        on the node's question
    else
        recursively move to **all** the children
        of the current node

Return vector of sums of all the weights visited.

(Figure from Middendorf, 2004)

Error on Simulated Data

Prediction

	DMR	DMC	AGV	LPA	SMW	RDS	RDG
DMR	99.3						
DMC		99.7					
AGV		0.1	0.84	13.5	1.2	0.5	
LPA			10.3	89.6			0.1
SMW			0.6		99.0	0.4	
RDS			0.2		0.8	99.0	
RDG	0.9			0.1			99.0

Correct Answer

Percentage of time predicted *x* when correct answer was *y*.
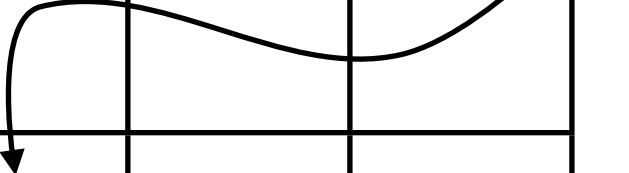
Drosophila Network

- Yeast 2-hybrid with probablities assigned by Giot et al. (34).

- Two different cutoffs for probability an edge really exists: $p^* = 0.5$ and $p^* = 0.65$

- (0.65 chosen because this is when the two largest components are merged).

- When $p^* = 0.65$, 3,359 vertices and 2,795 edges.

Predictions on Dros. Network:

| Rank	Eight-step subgraphs ($p* = 0.65$)		Subgraphs with up to seven edges ($p* = 0.65$)		Eight-step subgraphs ($p* = 0.5$)	
	Class	Score	Class	Score	Class	Score
1	DMC	8.2 ± 1.0	DMC	8.6 ± 1.1	DMC	0.8 ± 2.9
2	DMR	-6.8 ± 0.9	DMR	-6.1 ± 1.7	DMR	-2.1 ± 2.0
3	RDG	-9.5 ± 2.3	RDG	-9.3 ± 1.6	AGV	-3.1 ± 2.2
4	AGV	-10.6 ± 4.2	AGV	-11.5 ± 4.1	LPA	-10.1 ± 3.1
5	LPA	-16.5 ± 3.4	LPA	-14.3 ± 3.2	SMW	-20.6 ± 1.9
6	SMW	-18.9 ± 0.7	SMW	-18.3 ± 1.9	RDS	-22.3 ± 1.7
7	RDS	-19.1 ± 2.3	RDS	-19.9 ± 1.5	RDG	-22.5 ± 4.7

(Table from Middendorf, 2004)

stddev over cross-validation folds

Note: random growing preferred over preferential attachment

Which random processes can produce which subgraphs?

Rank score: % of random graphs that contained ≥ instances of the subgraph than exist in Dros.

0 means a random graph never contained this subgraph.
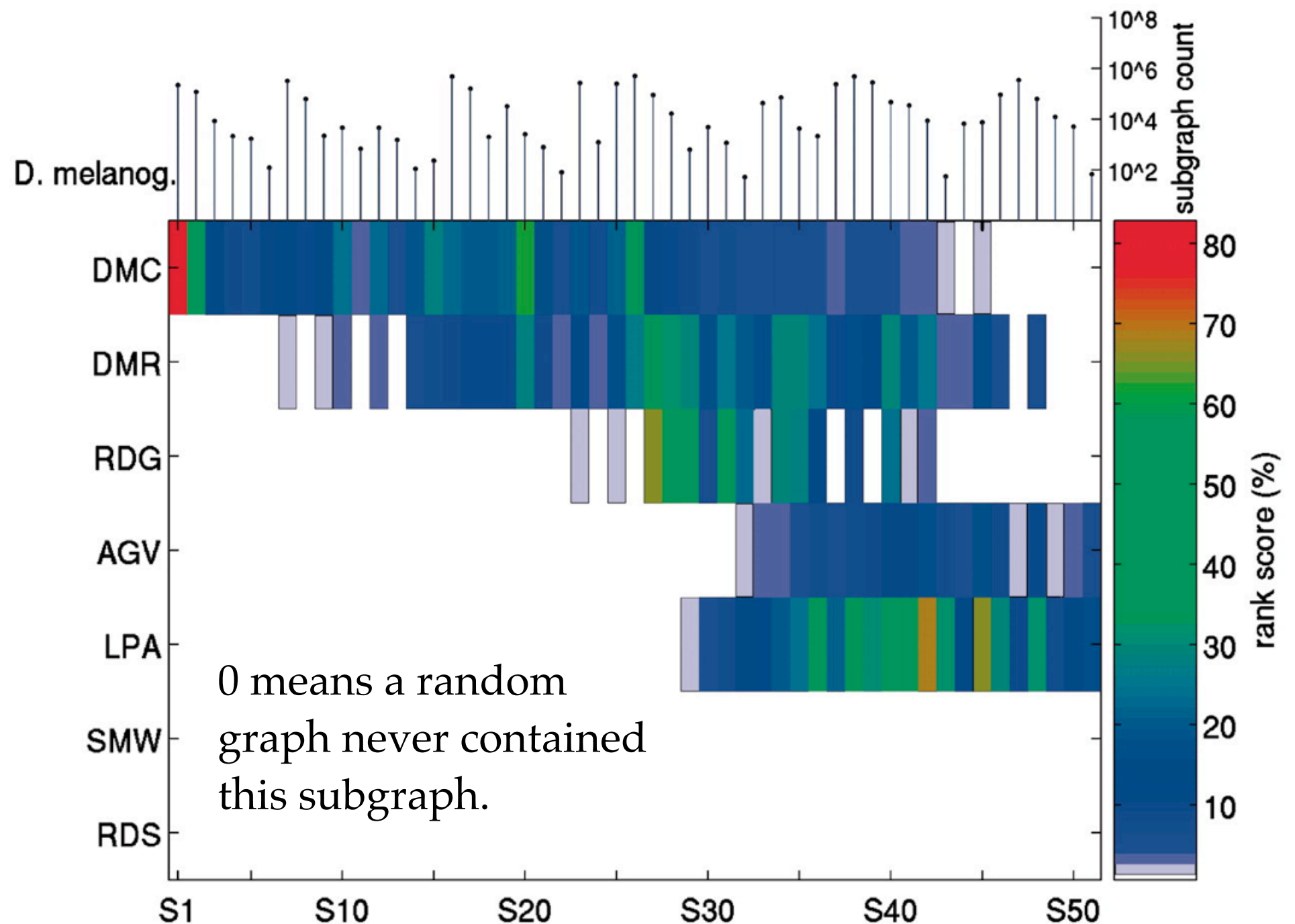
(Figure from Middendorf, 2004)

Robustness to Noise

Place where prediction changes

when fully randomized, looks like an Erdos-Renyi graph

$p = 1 - 10^{-3}$

(Figure from Middendorf, 2004)
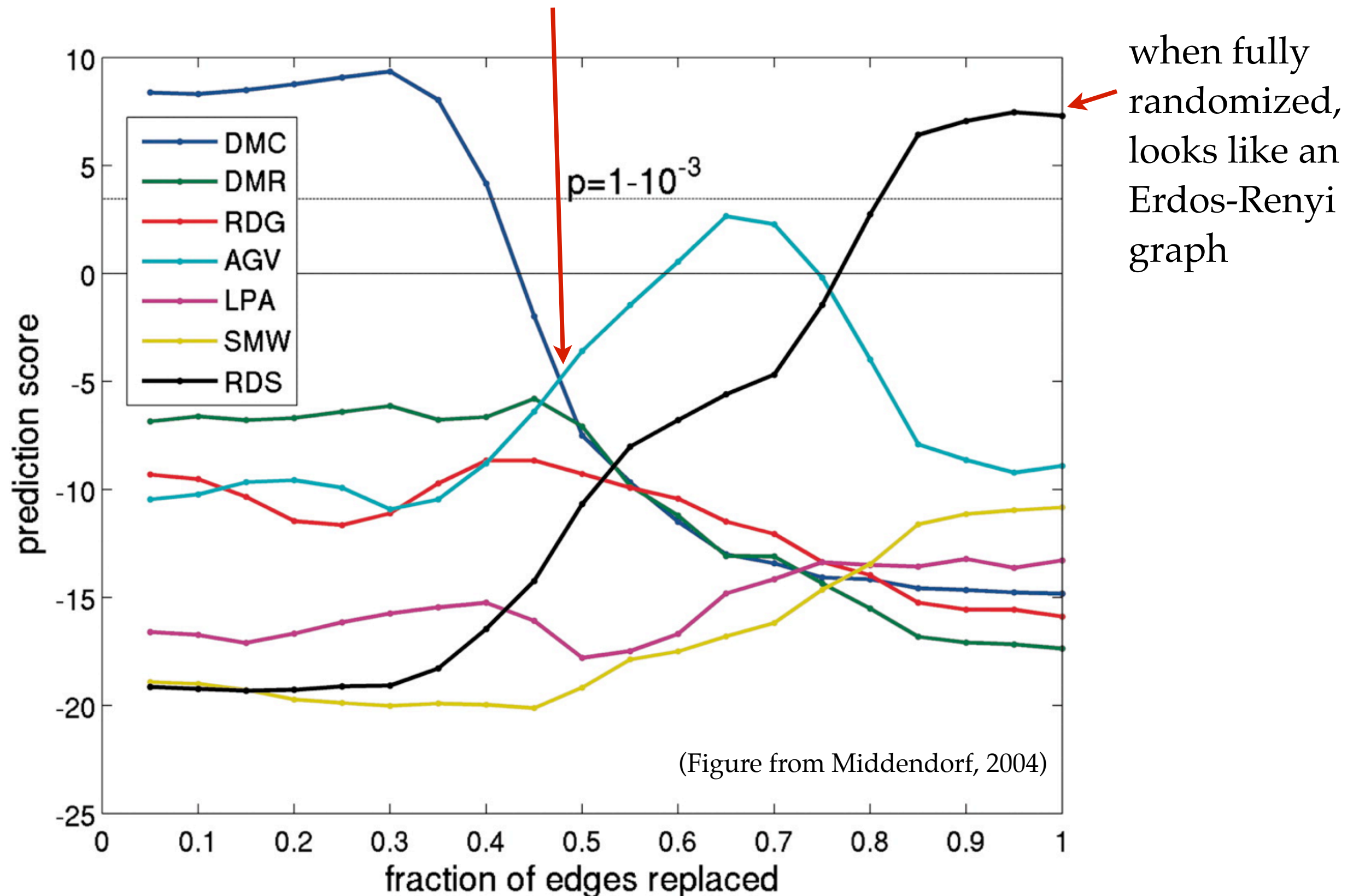
Summary

- Lots of random models for growing networks (there are still others we haven't covered)

- Each gives rise to networks with different properties.

- These networks can generally be distinguished by looking at the counts of various subgraphs that they contain.

- Using that idea, Middendorf et al. work backwards, and guess which model seems to fit real data.

- DMC seems like the best-fitting model for fly.