Handling Large Networks

The Scale Challenge

Many practical networks are massive:

- Internet: 50+ billion connected devices
- Human brain: 86 billion neurons, 100 trillion synapses
- Protein interaction networks: 20,000+ proteins, millions of interactions

Full network analysis is often computationally impossible or practically infeasible.

We need **network subsampling**

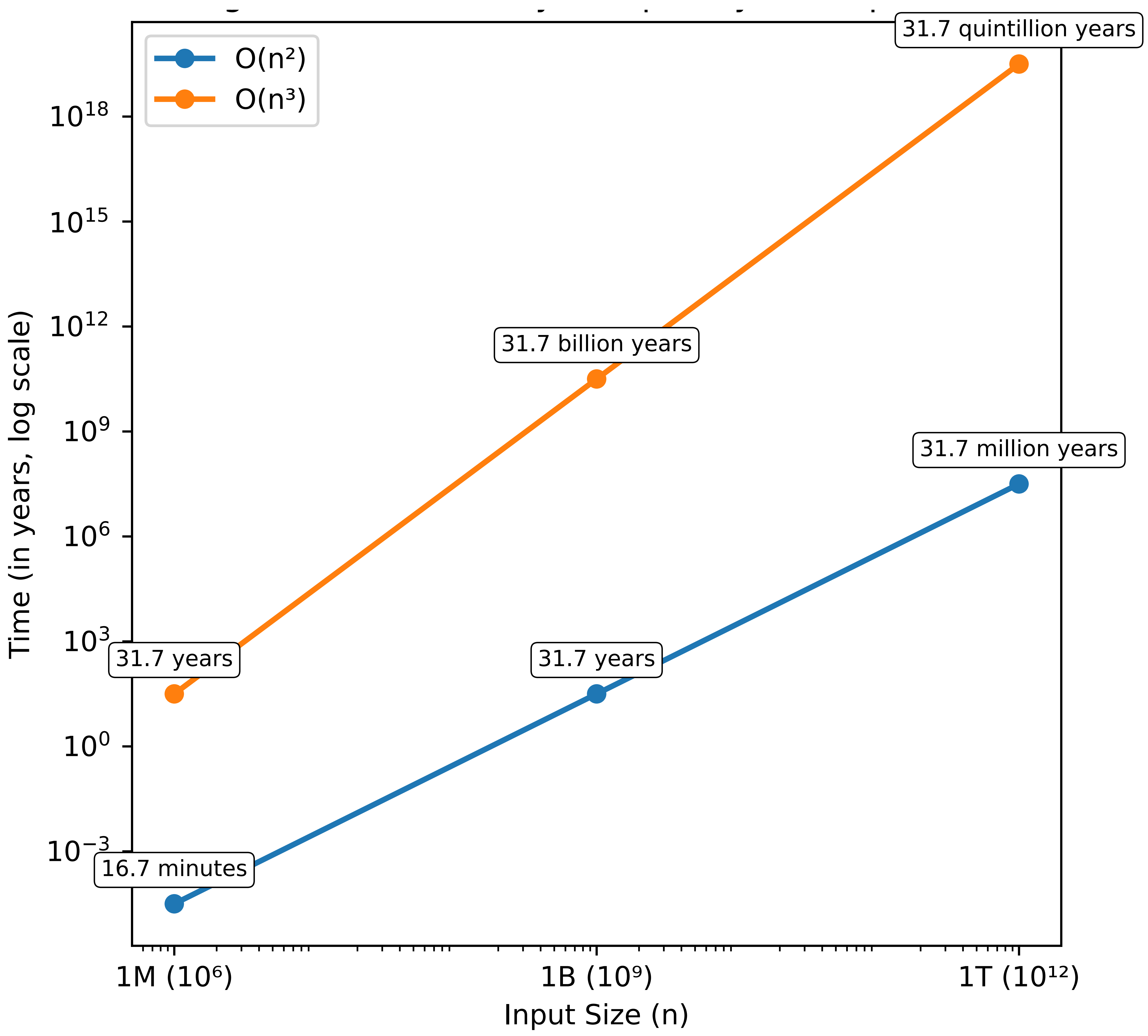
Computational Constraints

Some back-of-the-envelope calculations to grasp the constraints of large-scale networks

Memory:

- Adjacency matrix for n nodes: $O(n^2)$ space
- 1 million nodes → ~1 TB (1-bit cells)
- 1 billion nodes → ~1 PB (1-bit cells)

Computation (Time): many network algorithms scale as $O(x^2)$ or $O(x^3)$ — where x considers the number of nodes or edge (given n nodes, we can have ~n^2 edges).



Quality vs. Quantity Trade-offs

Complete Networks:

✅ No sampling bias

✅ Perfect structural preservation

❌ Computationally intractable

❌ May include noise/irrelevant connections

Sampled Networks:

❌ Introduces sampling bias

❌ May miss important patterns

✅ Computationally feasible

✅ Focus on relevant substructures

⚠️ Requires careful validation

How much can we subsample while preserving the insights we care about?

The Four Pillars of Network Subsampling Theory

1. **Representativeness**: does our sample reflect the network?
2. **Bias**: what systematic errors does our sampling introduce?
3. **Inference**: how do we make valid conclusions?
4. **Coverage**: what parts of the network can we actually reach?

These concepts are interconnected and determine the **validity** of our network analysis.

Representativeness

Does the subsample preserve (meaningful) network properties?

Typical network properties to preserve:

Degree distribution: $P(k)$ - probability a node has degree k — $O(n+m)$

Clustering coefficient: Local and global transitivity — resp. $O(n \cdot d_{max}^2)$ and $O(m^{1.5})$

Path length distribution: Distance patterns between nodes — $O(n^3)$

Centrality distributions: Important node identification — $O(n+m)$

If we run the same analysis on our sample vs. the full network, how similar are the results?

Mathematically, Let $G = (V, E)$ be the full network, $S \subseteq G$ be our sample. We want the measure of (meaningful) properties P on S and G such that $P(S) \approx P(G)$

Systematic Distortions and their Sources

- **Selection bias**: Who/what gets included?

  E.g., high-degree nodes more likely to be sampled, geographic clustering in spatial networks

- **Structural bias**: How does sampling change network topology?

  E.g., edge deletion creates **artificial components**, clustering coefficients systematically altered

Systematic Distortions and their Sources

High-degree nodes more likely to be sampled

- If sampling is based on **connectivity**, then **hubs** (nodes with many connections) are more likely to be included.
- Result: **skewed degree distribution**, the network appears more centralised or scale-free than it truly is.

This phenomenon is similar to snowball sampling, where high-degree nodes are overrepresented.

Systematic Distortions and their Sources

Geographic clustering in spatial networks

- In spatial networks (e.g., networks with a geographic basis, e.g., transportation), proximity can cause **local clusters** to be over-sampled.
- Result: overestimating local connectivity and underestimating long-range connections.

Systematic Distortions and their Sources

Edge deletion can:

- **disconnect the network**, creating **false components** that do not exist in the full graph
- **disrupt triangles**, artificially modifying the clustering coefficient
- increase **average shortest path** or **diameter**, making the network appear more fragmented or less efficient than it actually is.

Main Subsampling Techniques

Node-Based: preserve node-level properties

Edge-Based: better for structural properties

Traversal-Based: explore network systematically, good for connected structures

Naive Node-Based Sampling

Select nodes uniformly at random.

Algorithm: a) Number all nodes from 1 to n; b) generate a random sample of size n; c) include the selected nodes and their edges

Mathematically:

- Each node has probability $p(i) = 1/n$
- Independent selection

E.g., from a 1000-node network, randomly select 100 nodes

✅ Unbiased for node attributes

✅ Theoretically well-understood

✅ Easy to implement

✅ Reproducible (fixed a seed)

❌ Fragments network structure

❌ Poor for connectivity analysis

❌ Biased degree distribution

❌ Likely destroys paths

Degree-stratified Node-Based Sampling

Preserve degree distribution shape

Algorithm: a) partition nodes by degree classes: d_1 to d_{max} ; b) sample proportionally from each degree class.

Mathematically, sampling probability: $p(i)$ depends on $degree(i)$

✅ Preserves degree distribution (e.g., scale-free networks)

✅ More full-spectrum representation (better in clustered/weighed form for low-degree nodes)

✅ Samples tend to be more connected

❌ Need to compute degrees

❌ Low-degree nodes underrepresented (without clustered/weighed variant)

❌ More difficult to implement

❌ May distort clustering and paths

Naive Edge-Based Sampling

Select edges uniformly at random

Algorithm: a) list all edges from e_1 to e_m; b) randomly sample a fraction p of edges; c) include the endpoints (vertices) of the selected edges.

Mathematically, each edge has probability p of selection and the expected number of edges is $p \cdot m$; the expected number of nodes depends on the network structure

More efficient than node sampling for sparse networks

✅ Sampled subgraph more likely to be connected

✅ Likely preserve triangles and small pattern

✅ Efficient for sparse networks ($O(m)$ vs $O(n^2)$)

✅ Natural for interaction data (where edges are the focus)

❌ Biased toward high-degree nodes

❌ Difficult to predict resulting degrees

❌ Likely disrupts node-level properties

❌ Harder to correct for sampling bias

Induced Edge-Based Sampling

Sample edges, include all connected nodes

Algorithm: a) random edge sampling; b) include nodes connected to sampled edges; c) **include edges between included nodes**

Adds back edges that connect sampled nodes, even if those edges are not initially selected

Denser subgraph with more complete local structure

✅ Better clustering preservation (triangles)

✅ More realistic subnetworks (local communities)

✅ Better short paths preservation

✅ Natural stopping criterion

❌ Difficult to control final number of nodes and edges

❌ Bias toward hubs

❌ Computational overhead to identify all induced edges

Breadth-First Traversal-Based Sampling

Systematic layer-by-layer exploration

Algorithm: a) select one or more seed nodes; b) add all direct neighbors (layer 1); c) add neighbors of layer 1 nodes (layer 2); d) continue until desired sample size reached

Preserves network layer structure. Deterministic, given the seed selection. Generates a single component of a bounded diameter

✅ Guaranteed connectivity

✅ Preserves local structure

✅ Controllable expansion

✅ Good for local analysis (ego networks, local communities)

❌ Sample depends heavily on seed choice

❌ May miss distant network regions

❌ Tends to include similar-degree nodes

Depth-First Traversal-Based Sampling

Follow paths deeply before backtracking

Algorithm: a) start at seed node; b)follow random edge to unvisited neighbour; c) repeat the visit from new node; d) when stuck, backtrack to last node with unvisited neighbours; e) continue until sample size reached

Follows long paths through network, generating tree-like structures (resembling spanning trees) with large diameter

✅ Guaranteed connectivity

✅ Preserves global structures

✅ Low bias toward seed (lower seed proximity dependence than BFS).

✅ Controllable expansion

❌ Low local representativeness

❌ High diameter (actual density/ clustering vs high-diameter sample).

❌ Weak connectivity

❌ Unpredictable coverage (may miss large portions of the network)

Random-walk Traversal-Based Sampling

“Markovian” exploration of networks

Algorithm: a) Start at random node (e.g., chosen seed); b) at each step, move to random neighbour; c) continue for X steps; d) sample equals all visited nodes.

Mathematically, we build the Markov chain on the graph vertices where the transition probability is $P_{ij} = 1/d(i)$ if $(i,j) \in E$, 0 otherwise. Steady-state distribution (dependent on the degree of each node). Long-run behavior independent of starting point

✅ Well-understood Markov chain theory

✅ Memory efficient

✅ Can traverse very large components

❌ Visits high-degree nodes more frequently

❌ May take long time to cover network uniformly

❌ Stuck in starting component

❌ Different samples not independent

Evaluation and Validation

How do we know if our sample is good (i.e., valid)?

We can check **structural metrics**, i.e., does the sample preserve network topology?

⚠️ Quality assessment must be aligned with the intended use of the sampled network. ⚠️

Other methods include **task-specific** (does the sample support our analytical goals?) and **statistical** (how accurate are our estimates?) **metrics**, but we do not have the time to delve into those.

Structural Metrics

We can check essential structural — as in “preserving network topology” — properties by comparing network and sample under:

- **degree distribution** via Kolmogorov–Smirnov test (maximum difference between the two distributions, checking the likelihood they have the same underlying distribution), **moment matching** (compare mean, variance, skewness, etc.), **earth mover's distance** (how much one probability “mass” needs to move to match the other distribution);

- **clustering coefficient**: global and local clustering coefficient

- **path length**: average shortest path length, diameter, distribution of all pairwise distances

- **community structure**: modularity, number and size of communities

Pitfalls

The four deadly sins of subsampling:

1. **Ignoring network structure** in sampling design
2. **Assuming independence** when it is not the case
3. **Over-interpreting** results from biased samples
4. **Not accounting for missing data** mechanisms

Ignoring Network Structure

Common errors include (but are not limited to):

- Using **random sampling** on **scale-free networks** → **missing** important **hubs**
- **Traversal methods** to **disconnected networks** → **limited coverage**
- **Ignoring community structure** → samples may **over/under-represent groups**

Independence Assumptions

Network data can violate independence assumptions:

- **Spatial correlation**: nearby (geographically but also event-related, e.g., due to some external event) nodes have similar attributes
- **Homophily**: similar nodes tend to connect

Over-Interpreting Biased Samples

Consider how much the sample represents the actual data.

- Each sampling methods under- and over-represents some network traits
- Consider the properties (weaknesses in particular) of the sampling method to avoid drawing wrong general conclusions

Missing Data Mechanisms

Goes hand-in-hand with over-interpreting biased samples, i.e., also missing nodes and edges can skew the results of measures and lead to misrepresented elements, groups, and phenomena.

When applying a sampling method, know what data might be missing from the sample and account for it when interpreting the results — considering **why** data is missing and to possibly estimate **how much** is missing.