Graph Neural Netowrks

Saeed Saremi

Assigned reading: Chapter 13

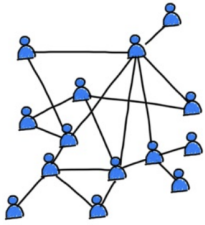
November 19, 2024

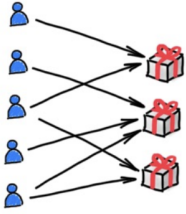
OUTLINE

- graph-structured data and graphs
- adjacency matrix and the permutation symmetry
- Graph Neural Networks (GNN)
  - convolutional networks revisited
  - graph convolutional networks
  - graph attention networks
  - message-passing neural networks
- expressive power of GNNs and the Weisfeiler[1]-Lehman test

[1]Weisfeiler, B. (1976). *On Construction and Identification of Graphs*, Springer.

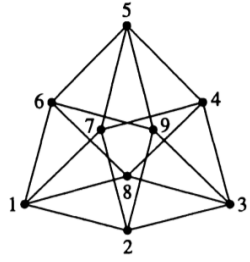
GRAPH-STRUCTURED DATA





- Graphs are mathematical abstractions that can describe complex systems of interactions.
- Examples include molecular graphs, social networks, and recommender systems.
- Every machine learning problem we have considered so far in the course – e.g., regression, classification, and generative modeling – can be posed for graph-structured data.

- A *graph* G an ordered pair of disjoint sets (V, E) such that E is a subset of the set $V^{(2)}$ of unordered pairs of V.
- The set V is the set of *vertices* and E is the set of *edges*. An edge $\{i, j\}$ is said to join the vertices i and j and is denoted by ij (or ji).
- If $ij \in E(G)$, then i and j are adjacent, or neighbouring, vertices of G. The set of all *neighbours* of node i is denoted by $\Gamma(i)$.
- We usually think of a graph as a collection of vertices some of which are joined by edges.
- We like to draw small graphs! For example, the graph with vertices $V = \{1, 2, ..., 9\}$ and edges $E = \{12, 23, 34, 45, 56, 61, 17, 72, 29, 95, 57, 74, 48, 83, 39, 96, 68, 81\}$ is immediately comprehended by looking at



ADJACENCY MATRIX

- There are several matrices naturally associated with a graph.
- The adjacency matrix $A = A(G) = (a_{ij})$ of a graph G is the $n \times n$ matrix given by

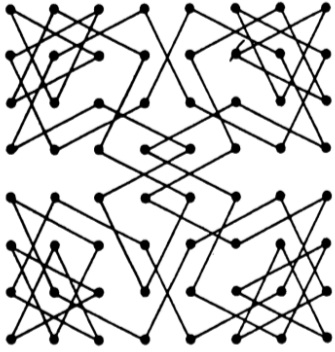
$$a_{ij} = \begin{cases} 1 & \text{if } ij \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

- An example for a 5 node graph:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

- Take the vertices of a pentagon and draw the graph! (How should we label the vertices?)

CANONICAL ORDERING



PERMUTATION MATRIX

- The adjacency matrix defines the structure of a graph. Naively, we can "flatten" it and use an MLP to solve our machine learning tasks. However, the problem is that labeling of vertices is arbitrary and changes the adjacency matrix.
- This is formalized by the permutation π which is a bijection from $[n]$ to $[n]$:

$$\pi : [n] \to [n],$$

which we can represent with the two-line notation, e.g.,

$$\pi = \begin{matrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 5 & 4 \end{matrix}$$

- We can represent the permutation π with a matrix $P(\pi) = (p_{ij})$:

$$p_{ij} = \begin{cases} 1 & \text{if } \pi(i) = j, \\ 0 & \text{otherwise.} \end{cases}$$

- For the example above:

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

PERMUTATION EFFECTS

- Let's assume we have feature vectors $x_i \in \mathbb{R}^d$ for all vertices in the graph, which we can represent compactly with the matrix X in $\mathbb{R}^{n \times d}$:

$$X = \begin{pmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{pmatrix}$$

- Now, the relabeling of the graph with the permutation π has the following effect on X:

$$\tilde{X} = PX$$

(The proof is simple, as P permutes the rows.)

- For the adjacency matrix, both the rows and columns become permuated, therefore the new (post permutation) adjacency matrix is given by:

$$\tilde{A} = PAP^\top$$

PERMUTATION INVARIANCE & EQUIVARIANCE

- Let's assume we make a graph-wide prediction f on the graph-structured data. Since the specific ordering we choose for the graph is arbitrary, our prediction must be invariant to node label reordering:

$$f(PX, PAP^\top) = f(X, A)$$

  for any P. (How large is the permutation group?)

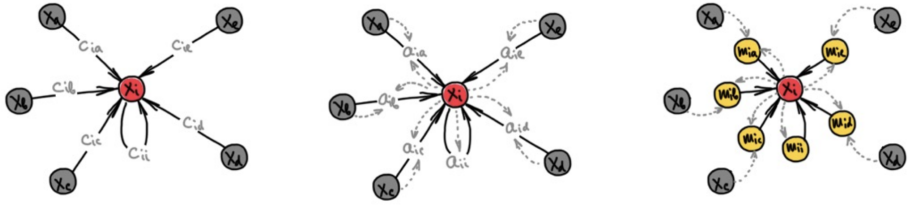
- We may also want to make node-specific predictions g. Again since specific ordering is arbitrary, under permutations of the node labels our predictions should permute the same way. Therefore, node predictions should be equivariant with respect to node reordering:

$$g(PX, PAP^\top) = Pg(X, A)$$

- Data augmentation is not an an option since the number of allowed permutations $n!$ is exponentially large:

$$n! \approx (n/e)^n$$

NEURAL MESSAGE PASSING



- Inspired by multilayer perceptrons, we define graph neural networks (GNN) by constructing a computational notion of *layer* for graph-structured data that can be applied repeatedly.
- The construction of layer should respect invariance (or equivariance) under permutation. In addition, graphs come in various sizes which our GNN should be able to handle.
- A general framework to construct GNNs goes around the message passing formalism.[2]

[2] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). "Neural message passing for quantum chemistry," In *International Conference on Machine Learning*.

CONVNETS REVISITED