

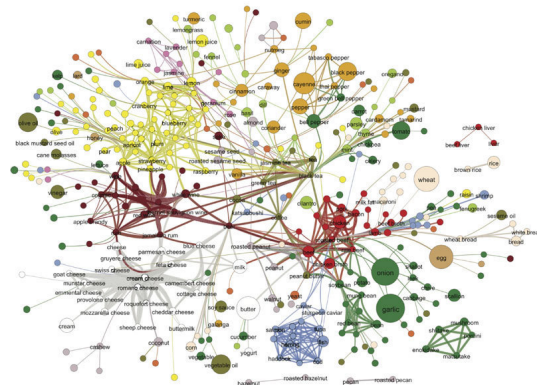
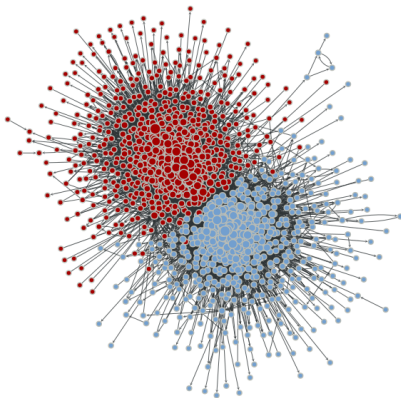
Learning tasks on graphs

Machine learning tasks on graphs operate at one of three levels, i.e., node, edge, or (sub)graph level.

Node-level tasks

Node-level tasks focus on making predictions about individual nodes within a graph. Typically, these tasks are performed on a single large network rather than multiple small graphs.

- *Node classification* (e.g., polarization of political blogs, hoaxes on *Wikipedia*)
- *Node regression* (e.g., age of users on social media, traffic volume in transportation)
- *Node ranking* (e.g., webpage ranking in *Google Search*, top influencers on *Instagram*)
- *Node clustering* (e.g., research areas of scientific papers, communities in social networks)

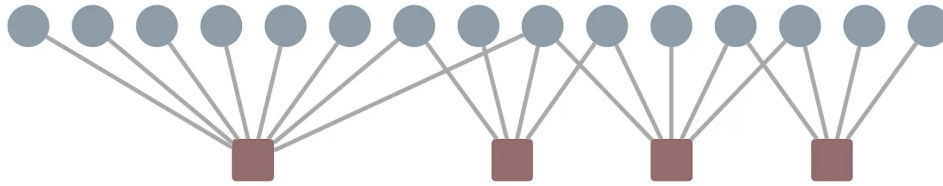


Edge-level tasks

Edge-level tasks focus on predicting new edges or inferring properties of existing edges in a graph. Typically, these tasks are performed on a single, often incomplete, network, such as bipartite graphs in the case of recommender systems.

- *Link prediction* (e.g., product recommendation on *Amazon*, song recommendation on *Spotify*, friendship suggestions on social media)

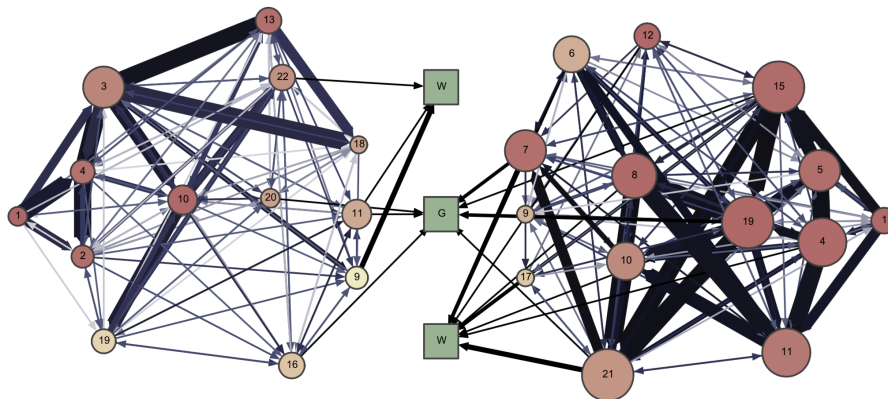
- *Edge classification* (e.g., type of bonds between atoms, road type in transportation networks, sentiment analysis in communication)
- *Edge regression* (e.g., strength of friendship ties on *Facebook*, movie rankings on *Netflix*, traffic flow in transportation)



(Sub)graph-level tasks

(Sub)graph-level tasks focus on predicting or inferring properties of individual graphs or their subgraphs. Typically, these tasks are performed on multiple small graphs, such as molecular graphs or ego-networks of users on social media.

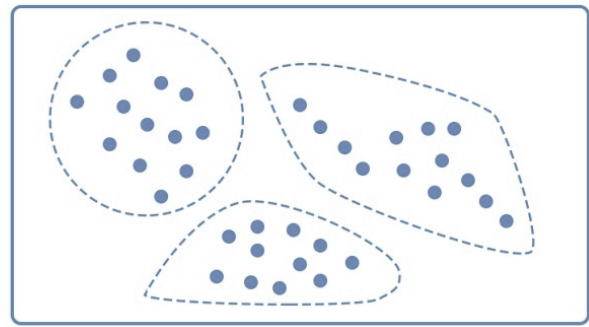
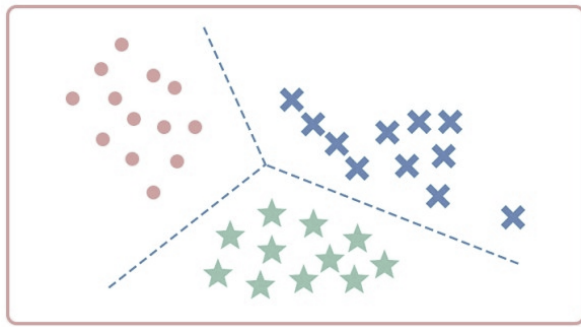
- *Graph classification* (e.g., playing strategy in football, toxicity or solubility of molecules)
- *Graph regression* (e.g., estimated time of arrival in *Google Maps*, material property estimation)
- *Graph generation* (e.g., good candidates for new drugs, realistic synthetic relational databases)



Graph learning settings

Supervised, unsupervised & self-supervised learning

Consider a node-level task, where the goal is to learn and predict a label y_i for each node i . In the case of node classification, the labels $y \in \{\text{foo}, \text{bar}, \text{baz}, \dots\}$ are categorical, whereas in node regression, the labels $y \in \mathbb{R}$ are real-valued. This is an instance of *supervised learning*, where the labels y are provided by an external source and serve as supervision for training a machine learning model.



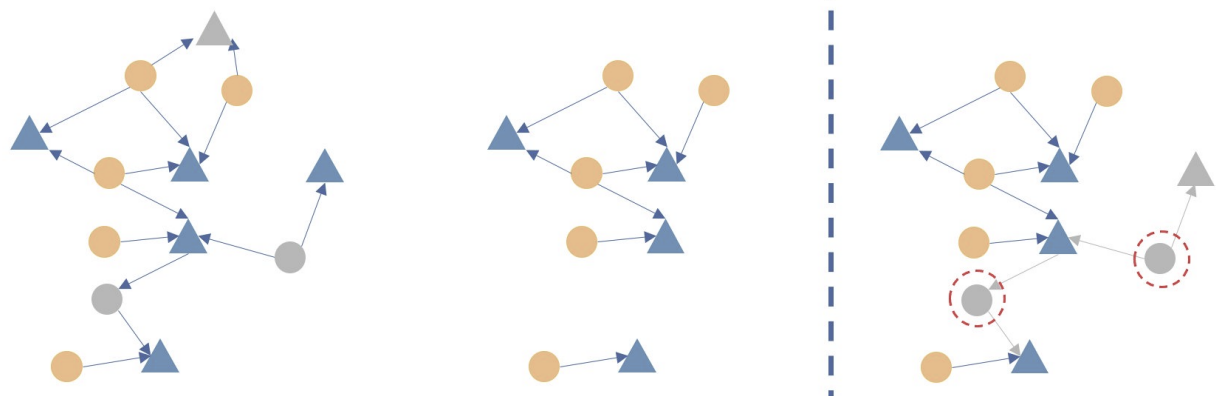
Sometimes we are not provided with external labels y , and our goal is to train the model without relying on task-specific supervision. This is an instance of *unsupervised learning*, such as node clustering.

However, the graph structure itself can still provide supervision signals, in which case we call this *self-supervised learning*. For example, we can train a model on a link prediction task, where we randomly remove some edges from the graph and try to predict them back, and then apply it to node clustering.

Note that the distinction between unsupervised and self-supervised learning paradigms is often subtle.

Transductive vs. inductive learning

In *transductive learning*, the model is trained on a fixed graph or multiple graphs that will not change at test time. For example, when predicting travel times in *Google Maps*, we may assume that the road network remains static.



In contrast, *inductive learning* is a more challenging problem, where the goal is to make predictions for new nodes, edges, or (sub)graphs that were not observed during training. For example, we might train a model to predict the toxicity of existing drugs, and then apply it to estimate the toxicity of new, yet-to-be-synthesized compounds.

Note that some learning tasks are well defined only in the inductive setting (e.g., graph classification).