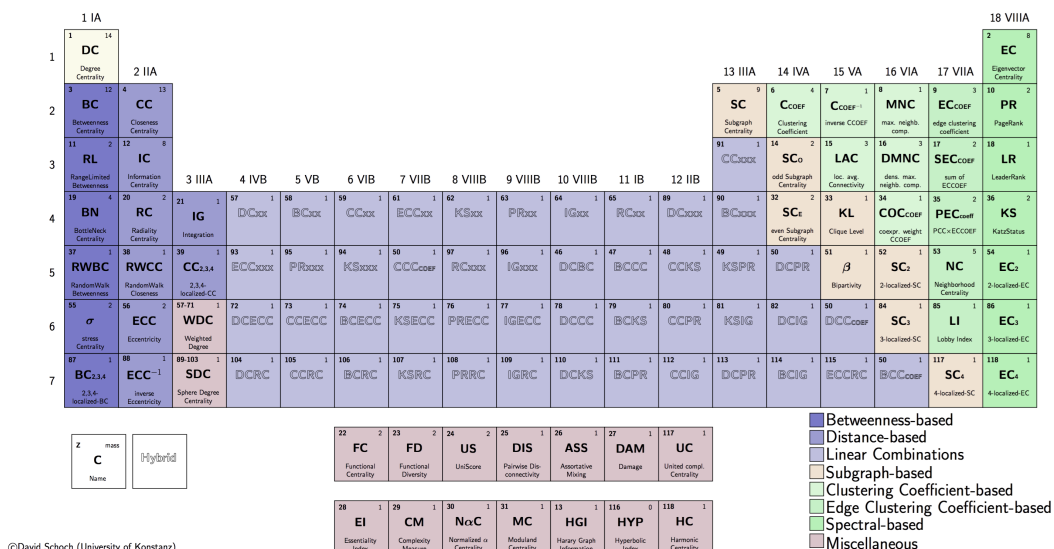


Brief history of graph learning

Since around the year 2000

Since the emergence of modern network science almost 30 years ago, researchers have employed *network analysis* techniques directly for graph learning tasks. It is important to note that we do not train a machine learning model in this context.

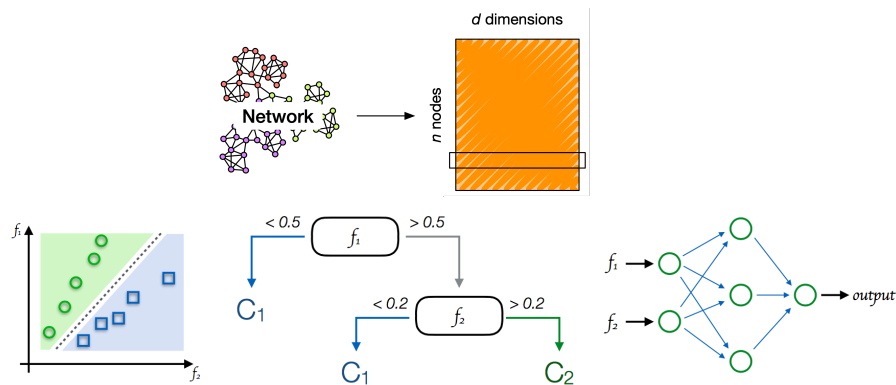
- *Node ranking* (e.g., node centrality measures, link analysis algorithms, graphlet orbits, ego-network analysis)
- *Node clustering* (e.g., community detection methods, blockmodeling by hierarchical clustering, stochastic block models)
- *Link prediction* (e.g., link bridging measures, node similarity indices, matrix factorization, dimensionality reduction)
- *Graph similarity* (e.g., graph kernel functions, graphlet degree distributions, portrait divergence)
- Etc.



Until around 10 years ago

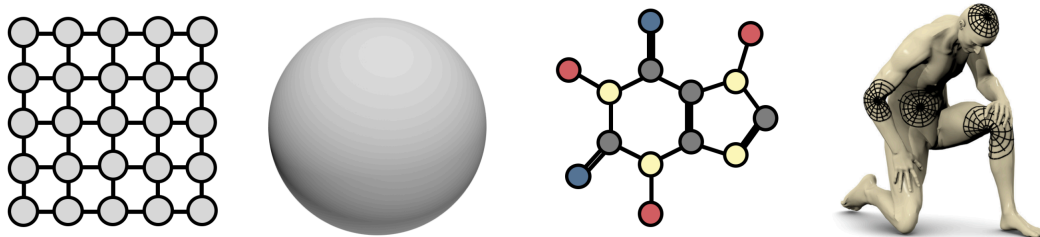
Alternatively, network analysis techniques can be used to generate informative features for nodes, edges, or (sub)graphs (a process known as *feature engineering*), which are then fed

into a machine learning model in tabular form. However, these features are task-dependent and must be redesigned for each new problem, a process that is both time-consuming and demands substantial domain expertise.



Modern graph deep learning

Graph representation learning has emerged as a way to apply machine learning models directly to graph-structured data ([Hamilton, 2020](#)). This is challenging since graphs do not have an inherent ordering or reference point like sequences (e.g., text, speech), grids (e.g., images, videos) or classical tabular data.



1. Dimensionality reduction & *matrix factorization* techniques (e.g., *NMF*, *SVD*, *PCA*)
 - decomposition of adjacency matrix A , graph Laplacian L or some other graph matrix
2. Methods based on restarting *random walks* (e.g., *node2vec*, *edge2vec*, *struct2vec*)
 - "similar" nodes, edges, or graphs have similar representations (i.e. embeddings) independently of the task
 - dense, low-dimensional node embeddings instead of sparse, high-dimensional node neighborhoods
3. Modern *graph neural networks* & *transformers* (e.g., *GCN*, *GAT*, *GIN*, *GraphSAGE*)
 - node, edge, or graph representations are learned for specific tasks (in transductive or inductive setting)