Training Camp on "Knowledge Graph Completion"

– Sapienza University, M.Sc. Degree in Data Science –

Fabio Galasso, Laura Laurenti, Alessio Sampieri
Sapienza University of Rome

Ilaria Bordino, Francesco Gullo, Lorenzo Severini
UniCredit Services
"AI, Data & Analytics ICT" Department
"Applied Research & Innovation" unit

https://www.kaggle.com/c/unicredittrainingcamp/overview/lectures

June 30th – July 2nd, 2021

Day 1: Traditional ML Methods for Graph completion

Schedule

- Introduction to vanilla graphs
  - Definition of (vanilla) graph
  - Main graph properties
- Introduction to knowledge graphs
  - Definition of knowledge graphs
  - Applications
- Link prediction Problem
- ML Pipelines for link prediction
- Compute topological features in (knowledge) graphs
  - Distance based features
  - Local neighbourhood overlap features
  - Global neighbourhood overlap features
- Evaluate the KG completion task
- Lab
  - Introduction to NetworkX and how to compute graph topological features
  - Build a classifier for link prediction

Expected outcome at the end of Day 1

Capability of:

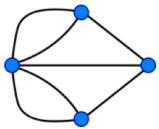
- loading a knowledge graph with NetworkX and extract some topological node/edge features
- build a model with graph topological features for the link prediction problem

References

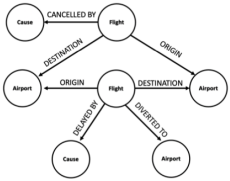
- [*survey paper*] A survey of link prediction in complex networks
- [*course*] Machine Learning with Graphs – Lecture 2 (Traditional Methods for ML on Graphs)
- [*framework*] NetworkX
- [*framework*] scikit-learn

Graphs: A Simple but Powerful Model

- Entities: Set of vertices
- Pairwise relations among vertices: set of edges
- Can add directions, weights, labels, timestamps



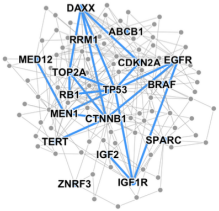
- Graphs can model many real-world datasets:
  - People who are friends
  - Computers that are interconnected
  - Web pages that point to each other
  - Proteins that interact
- Graph Theory started in the 18th century, with L. Euler
- The problem of Konigsberg bridges
- Since then, graphs have been studied extensively



Event Graphs

Image credit: SalientNetworks

Computer Networks



Disease Pathways

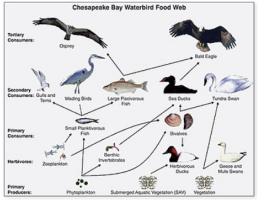


Image credit: Wikipedia

Food Webs

Image credit: Pinterest

Particle Networks

Image credit: visitlondon.com

Underground Networks

Image credit: Medium

Social Networks

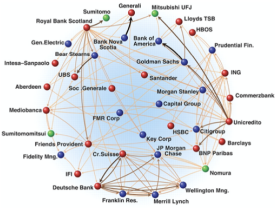


Image credit: Science

Economic Networks

Image credit: Lumen Learning

Communication Networks

Citation Networks

Image credit: Missoula Current News

Internet

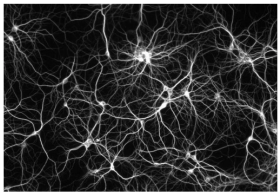


Image credit: The Conversation

Networks of Neurons

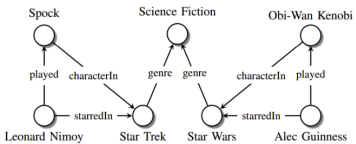


Image credit: Maximilian Nickel et al

Knowledge Graphs

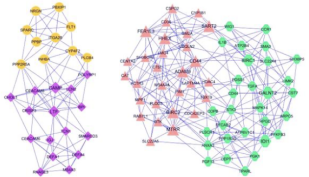


Image credit: ese.wustl.edu

Regulatory Networks

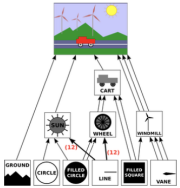


Image credit: math.hws.edu

Scene Graphs

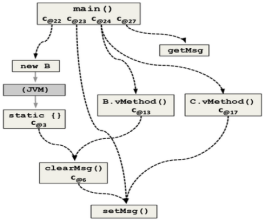
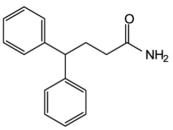


Image credit: ResearchGate

Code Graphs



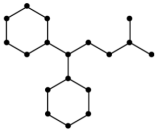


Image credit: MDPI

Molecules

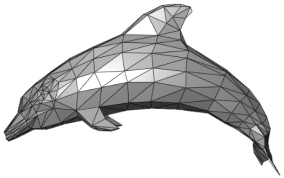


Image credit: Wikipedia

3D Shapes

Social networks

- links denote a social interaction
  - networks of acquaintances
  - collaboration networks
    - actor networks
    - co-authorship networks
    - director networks
  - phone-call networks
  - e-mail networks
  - IM networks

Knowledge and Information Networks

- nodes store information, links associate information
  - citation network (directed, acyclic)
  - the web (directed)
  - peer-to-peer networks
  - word networks
  - networks of trust
  - software graphs
  - bluetooth networks
  - home page/blog networks

Technological Networks

- networks built for distribution of a commodity
- the internet, power grids, telephone networks
- airline networks, transportation networks

Biological Networks

- biological systems represented as networks
  - protein-protein interaction networks
  - gene regulation networks
  - gene co-expression networks
  - metabolic pathways
  - the food web
  - neural networks

Graph: definition

A graph (or network) G is made by.

- V is a set of **vertices** (or nodes) v. They are the **entities** of the graph.
- $E \subseteq V \times V$ is a set of pair (u, v) called **edges**. They represents the pairwise **relations** among vertices

  - $n = |V|$ is the number of vertices
  - $m = |E|$ is the number of edges

Pairwise relations E can be:

- unordered $\rightarrow$ **undirected** graphs
- with direction (ordered) $\rightarrow$ **directed** graphs
- with weights $\rightarrow$ **weighted** graphs

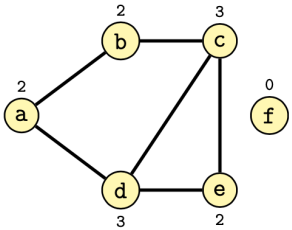
Degree of a node

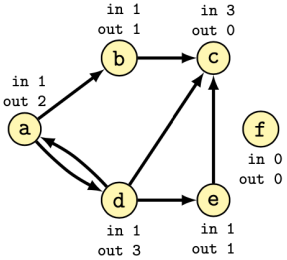
Neighborhood

v is a **neighbor** of u if there exists an edge (u, v) in E (the set of neighbors of u is called neighborhood).

Degree

- in an undirected graph, the degree of a node v is its number of *neighbors* i.e. the number of incident edges to v
- in a directed graph, we called in-degree the number of incoming edges and out-degree the number of outgoing edges.





Credits: A. Montresor

Triangles

Triangle

A **triplet** is three nodes that are connected by either two (open triplet) or three (closed triplet) undirected ties. A triangle is a closed triplet.

Clustering coefficient

The **clustering coefficient** of a graph is the ratio between the number of closed triplets and the number of all triplets (open and closed) .

Paths

Path

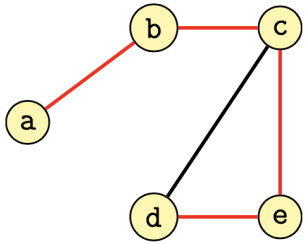
In a graph (directed or not), a path P of length k is a sequence of nodes $u_0, u_1, \ldots, u_k$ such that $(u_i, u_{i+1}) \in E$ for $0 \leq k \leq k-1$.

Reachability

A node v is reachable from u if there exists at least a path from u to v (in undirected graph, the reachability relation is symmetric)

Shortest path

The path with the minimum length between two nodes is called **shortest path**.
The longest shortest path is called the diameter of the graph.



Credits: A. Montresor

- (a, b, c, e, d) is **path** between a and d
- (a, b, c, d) is the **shortest path** between a and d
- (a, b, c, d) is the **diameter** of the graph

Visit

Breadth First Search (BFS)

visit all the nodes in increasing distance from the source (i.e the node where the visit start)

Depth First Search (DFS)

explores as far as possible along each branch before backtracking.

Complexity: $O(n + m)$

Connectivity

Connectivity in undirected graph

An undirected graph G is connected if every node is reachable from any other node.

Connectivity in directed graph

- A directed graph is **strongly connected** if every node is reachable from any other node.
- A directed graph is **weakly connected** if the corresponding undirected graph (i.e. removing the orientation to every edge) is connected.

How we can verify if a graph is connected?

1. Perform a DFS from a node
2. if at the end of the DFS any node is visited, the graph is connected (otherwise it is composed by more than one connected component)

Centrality

Centrality metrics

Centrality metrics measure the **importance** of a node within a graph according to the the position of the node in the network (and the network structure)

Taxonomy for centrality measures

Several definitions of centrality indices based on how they are computed:

- Geometric: node degree, closeness centrality, Lin's index, harmonic centrality, eccentricity, etc.
- Path-based: number of (shortest) paths passing thorough a node, coverage centrality, betweenness and its variant, etc.
- Spectral: left dominant eigenvector of a matrix derived from the graph, Katz index, PageRank, HITS, etc.

Heterogeneous graph

Definition

A heterogeneous graph is $G = (V, E, R, T)$

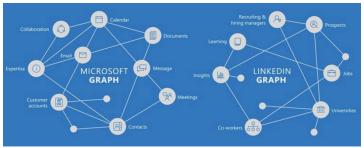
- Nodes with node types $v_i \in V$
- Edges with relation types $(v_i, r, v_j) \in E$
- Node type $T(v_i)$
- Relation type $r \in R$

A **Knowledge Graph** is an example of a heterogeneous graph

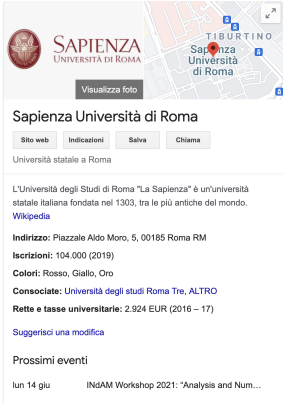
- Nodes are entities
- Nodes are labeled with their types
- (Directed) Edges between two nodes capture relationships between entities
- The first entity of the relation is called **head**, the second **tail**

Examples of knowledge graphs

- Google Knowledge Graph
- Amazon Product Graph
- Facebook Graph API
- IBM Watson
- Microsoft Satori
- LinkedIn Knowledge Graph



Credits: LinkedIn



Google KG

Common characteristics:

- Massive: millions of nodes and edges
- Incomplete: many true edges are missing

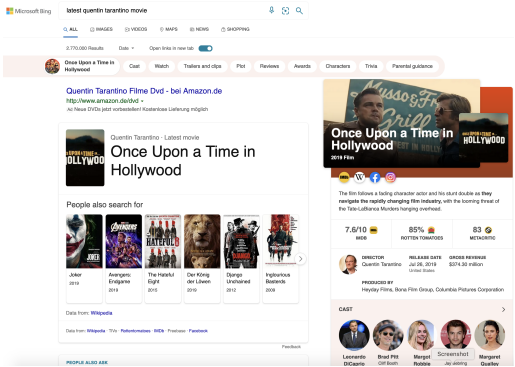
Application of KG

Applications

- Question answering and conversation agents
- Information extraction
- Content recommendation

Publicly available KGs

FreeBase, Wikidata, Dbpedia, YAGO, NELL



Example of information extraction using Bing KG

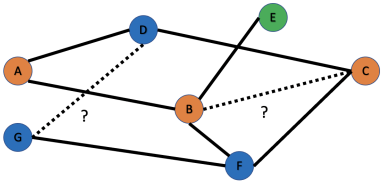
Link prediction problem

Definition

- The task is to predict new links based on existing links.
- The key is to design features for a pair of nodes.

Two formulations of the link prediction task:

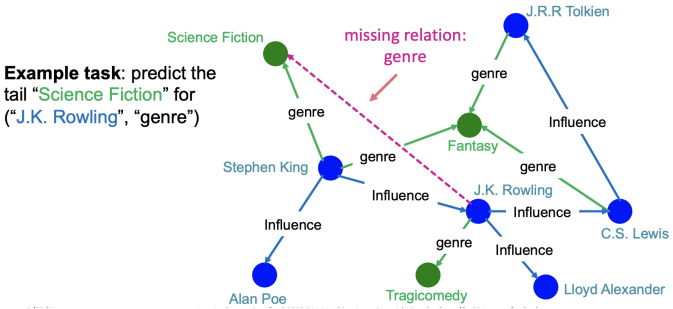
1. Links missing at random: Remove a random set of links and then aim to predict them
2. Links over time: given a graph with edges up to time t, predict the links are going to appear at time $t + k$ with $k > 0$



KG completion

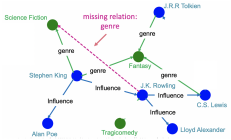
Given an enormous KG, can we complete the KG?

- For a given (**head**, **relation**), we predict missing **tails**.
  - (Note this is slightly different from link prediction task)



A pipeline to predict new links

Training Set



Validation Set

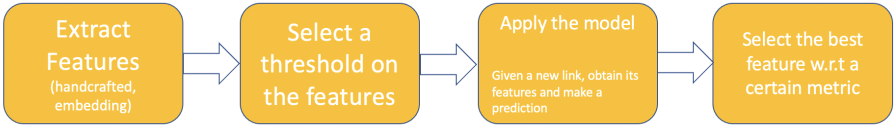
C.S Lewis - [Influence] - J.R.R Tolkien
J.K. Rowling -[genre] - Tragicomedy
C.S Lewis - [Influence] - Stephen King
Stephen King -[genre] - Tragicomedy

Test Set

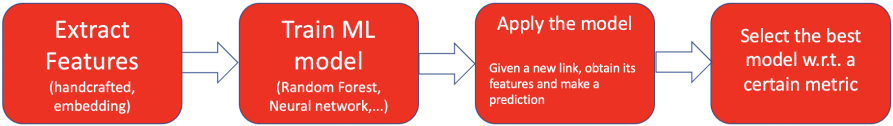
J.K. Rowling -[genre] - Science Fiction
Stephen King -[Influence] - C.S Lewis

Traditional ML Pipeline for link prediction

Threshold on handcrafted features



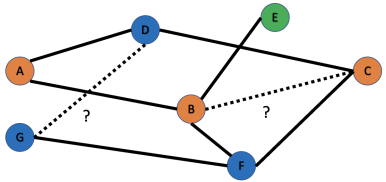
Handcrafted features + ML model



Distance based feature

Shortest-path

Compute the minimum length path between two nodes



$SP_{A,C} = 3$
$SP_{G,F} = 1$

Limitation

Does not capture the degree of neighborhood overlap

Local neighborhood overlap

Captures the number of neighbors shared between two nodes v_1 and v_2. Let N be the size of the neighborood of v_1

- Common neighbors

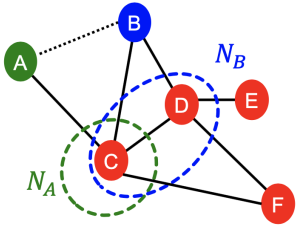
$$CN(v_1, v_2) = |N(v_1) \cap N(v_2)|$$

- Jaccard coefficient

$$JC(v_1, v_2) = \frac{|N(v_1) \cap N(v_2)|}{|N(v_1) \cup N(v_2)|}$$

- Adamic-Adar coefficient

$$AA(v_1, v_2) = \sum_{u \in N(v_1) \cap N(v_2)} \frac{1}{\log |N(u)|}$$



- $CN(A, B) = |N(A) \cap N(B)| = |\{C\}| = 1$
- $JC(A, B) = \frac{|N(A) \cap N(B)|}{|N(A) \cap N(B)|} = \frac{|\{C\}|}{|\{C, D\}|} = \frac{1}{2}$
- $AA(A, B) = \frac{1}{log|N(C)|} = \frac{1}{log(4)}$

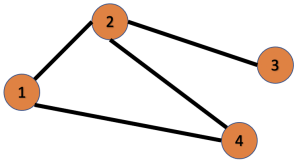
Limitation

Metric is always zero if the two nodes do not have any neighbors in common. However, the two nodes may still potentially be connected.

Global neighborhood overlap

Global neighborhood overlap metrics resolve the limitation by considering the entire graph

A: Adjacency matrix
$A_{ij} = 1$ if $i \in N(j)$



Example:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Katz index

- counts the number of paths of all lengths between two nodes
- Use adjacency matrix to compute the number of paths between two nodes
- $KI = \sum_{i=1}^{\infty} \beta^i \mathbf{A}^i = (\mathbf{I} - \beta\mathbf{A})^{-1} - \mathbf{I}$ where $0 < \beta < 1$ is a discount factor

Evaluate the link prediction model

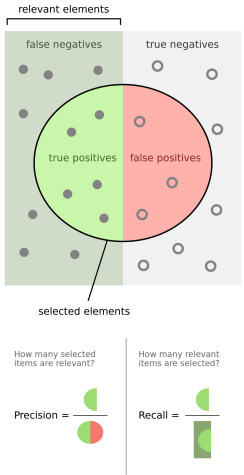
In the **Kaggle** competition, we evaluate the KG completion task using the **F1-score**

- $Precision = \frac{TruePositive}{TruePositive + FalsePositive}$
- $Recall = \frac{TruePositive}{TruePositive + FalseNegative}$
- $F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$

Triples	**Label**
C.S Lewis - [Influence] - J.R.R Tolkien	True
J.K. Rowling -[genre] - Tragicomedy	False
C.S Lewis - [Influence] - Stephen King	True
Stephen King -[genre] - Tragicomedy	False

$$Precision = \frac{1}{1+1} \quad Recall = \frac{1}{1+1}$$

$$F1 = 2 \cdot \frac{\frac{1}{2} \cdot \frac{1}{2}}{\frac{1}{2} + \frac{1}{2}} = \frac{1}{2}$$



Credits: Wikipedia