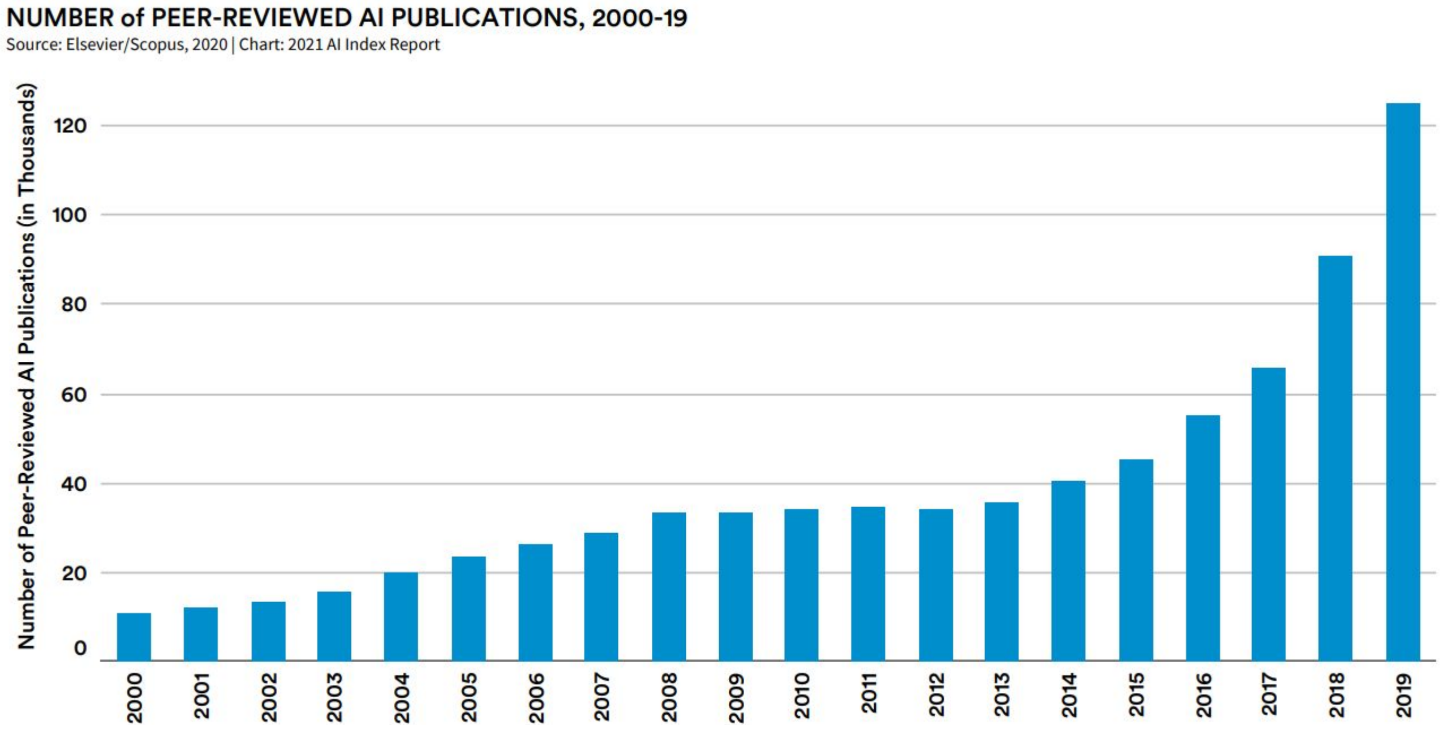
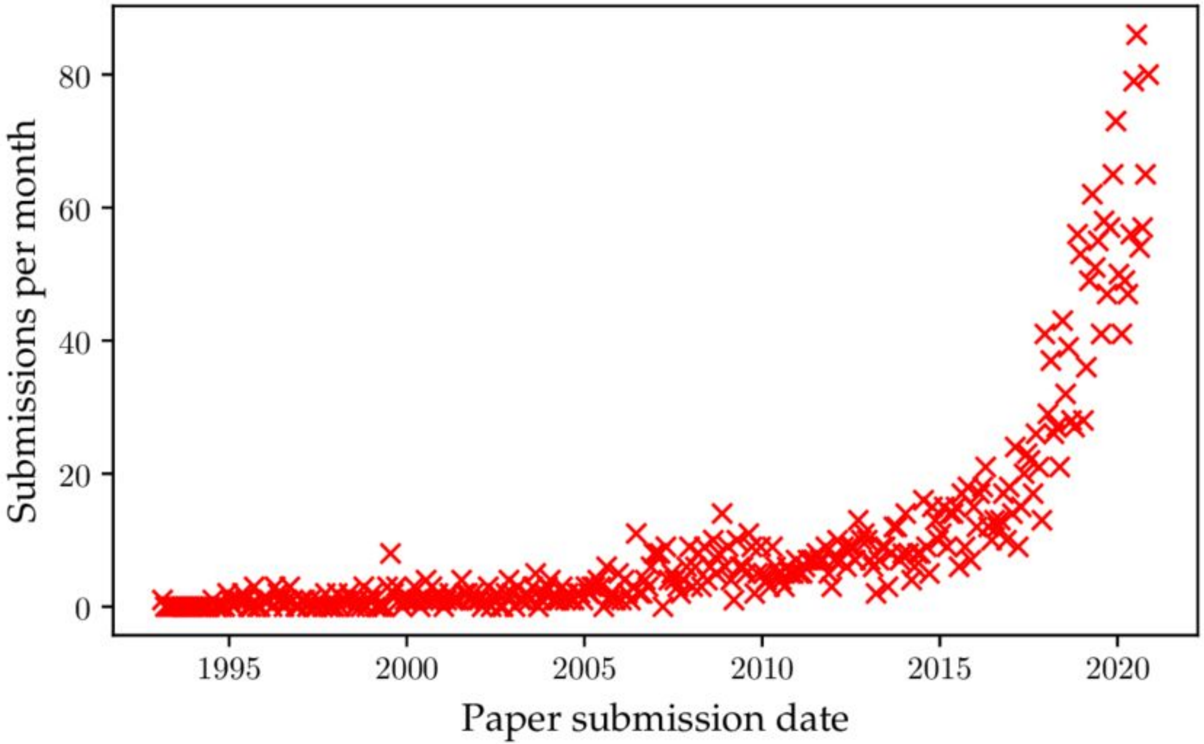
Recommending Academic Papers with Graph Neural Networks

Bowen Li

Motivation


NUMBER of PEER-REVIEWED AI PUBLICATIONS, 2000-19
Source: Elsevier/Scopus, 2020 | Chart: 2021 AI Index Report
Number of Peer-Reviewed AI Publications (in Thousands)
120
100
80
60
40
20
0
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019

Motivation



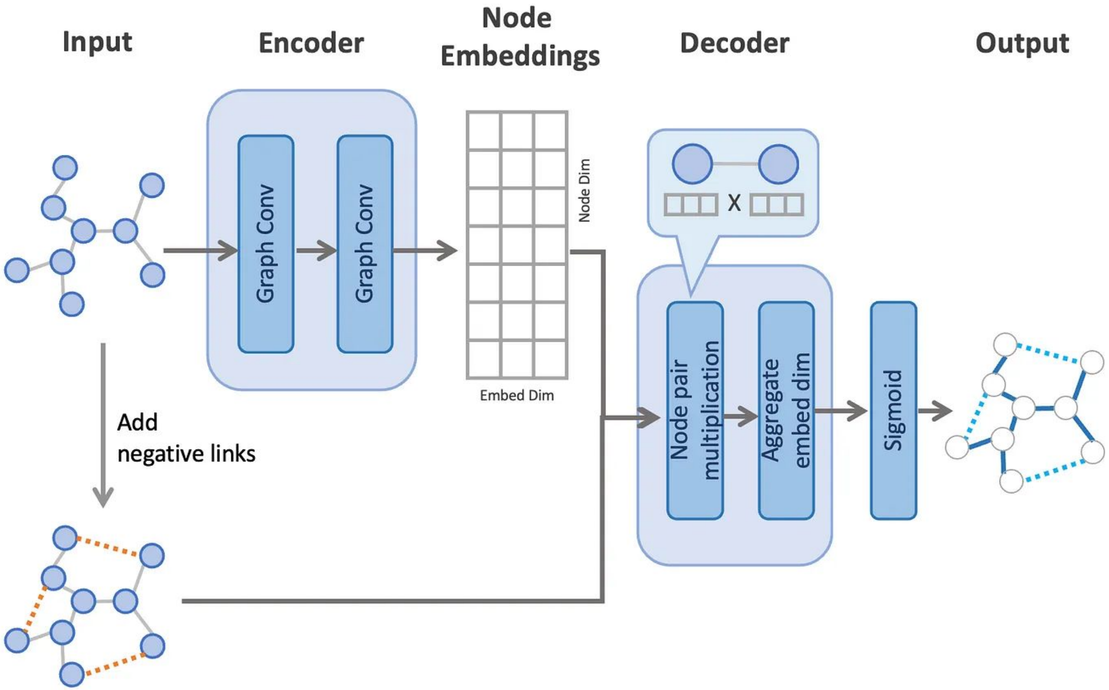
Source: https://www.researchgate.net/publication/365209417_Astronomia_ex_machina_a_history_primer_and_outlook_on_neural_networks_in_astronomy

Problem Statement

Given a paper, can we leverage graph neural networks (GNNs) to provide recommendations to helpful prior or future readings based on a citation network?

Link Prediction

Predict missing edges in a graph:



Source: https://towardsdatascience.com/graph-neural-networks-with-pyg-on-node-classification-link-prediction-and-anomaly-detection-14aa38fe1275

Dataset: CORA-ML

- Directed: Citing paper points to cited paper

- 2995 papers
  - Each paper represented by 2879-dimensional embedding
  - Papers in the topic of machine learning

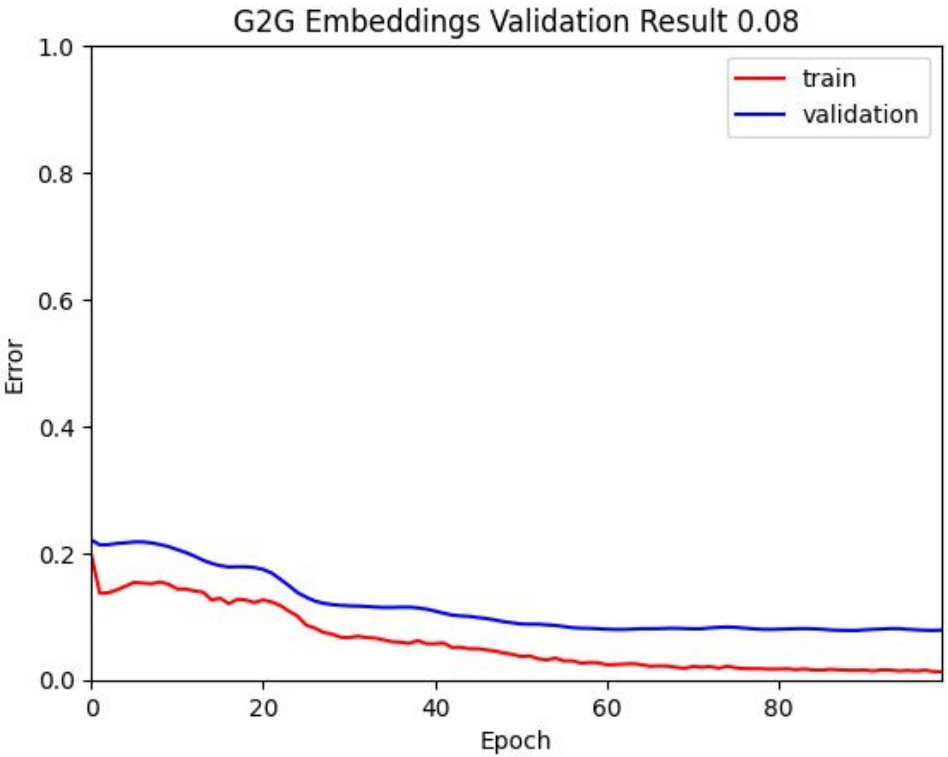
- 8416 citation relationships

- https://github.com/abojchevski/graph2gauss/tree/master

Model & Training Parameters

- Graph Encoder:
  - Conv (2879, 128)
  - ReLU
  - Conv (128, 64)
- Decoder:
  - Cosine similarity
- Split: 85% Train, 5% Validation, 10% Test
- Optimizer: BCEWithLogits
- Learning rate: 0.001
- Epochs: 100

Results



Test ROC-AUC Score: 0.929

Qualitative Evaluations

A COMPRESSION ALGORITHM FOR PROBABILITY TRANSITION MATRICES*

WILLIAM M. SPEARS†

Abstract. This paper describes a compression algorithm for probability transition matrices. The compressed matrix is itself a probability transition matrix. In general the compression is not error free, but the error appears to be small even for high levels of compression.

Key words. probability transition matrix, transient behavior, compression, lumping, aggregation

Cites

Using Markov Chains to Analyze GAFOs

Kenneth A. De Jong, William M. Spears, Diana F. Gordon

Analyzing GAs Using Markov Models with Semantically Ordered and Lumped States *

William M. Spears
Code 5510 - AI Center
Naval Research Laboratory
Washington, DC 20375-5337
spears@aic.nrl.navy.mil

Kenneth A. De Jong
Computer Science Department
George Mason University
Fairfax, VA 22030
kdejong@gmu.edu

An Overview of Evolutionary Computation †

William M. Spears ‡
Kenneth A. De Jong
Thomas Bäck
David B. Fogel
Hugo de Garis

Abstract. Evolutionary computation uses computational models of evolutionary processes as key elements in the design and implementation of computer-based problem solving systems. In this paper we provide an overview of evolutionary computation, and describe several evolutionary algorithms that are currently of interest. Important similarities and differences are noted, which lead to a discussion of important issues that need to be resolved, and items for future research.

Qualitative Evaluations

A COMPRESSION ALGORITHM FOR PROBABILITY TRANSITION MATRICES*

WILLIAM M. SPEARS†

Abstract. This paper describes a compression algorithm for probability transition matrices. The compressed matrix is itself a probability transition matrix. In general the compression is not error free, but the error appears to be small even for high levels of compression.

Key words. probability transition matrix, transient behavior, compression, lumping, aggregation

Predicted

Multiassociative Memory

John F. Kolen
Jordan B. Pollack

5 citations

Evolving Optimal Neural Networks Using Genetic Algorithms with Occam's Razor*

Byoung-Tak Zhang
Heinz Mühlenbein

112 citations

Discussion

- Although it can point to somewhat relevant papers, more information is needed for the model to be able to recommend pedagogically

- Limitations
  - Small dataset size
    - Average node degree of 2.8
  - Unknown initial node embedding method
  - A “good” recommendation is ambiguous
  - Restriction to academic publications makes more fundamental expository pieces a rarity

Next Steps

- Collect and/or test on larger dataset
- Use different embedding methods and incorporate content embeddings into prediction
- Inductive Learning to deal with unseen node embeddings