Data handling in PyG (part 2)

Giovanni Pellegrini[1,2,3]

SML[1] Lab, University of Trento, Italy
TIM[2]
EIT DIGITAL[3]

01 Last tutorial recap

Pytorch Geometric most common classes for
Data handling and manipulation (**torch_geometric.data**):

- **Data**
- **Dataset (InMemoryDataset)**
- **ClusterData , ClusterLoader**
- **Batch**
- **NeighborSampler**
- **DataLoader**

01 Last tutorial recap

Pytorch Geometric most common classes for
Data handling and manipulation (**torch_geometric.data**):

- **Data**
- **Dataset (InMemoryDataset)**
- **ClusterData , ClusterLoader**
- **Batch**
- **NeighborSampler**
- **DataLoader**

Available datasets in PyG (**torch_geometric.datasets**)

- **Planetoid**
- **TUDataset**
- **Cora**
- ...and many others

01 Last tutorial recap

Pytorch Geometric most common classes for
Data handling and manipulation (**torch_geometric.data**):

- **Data**
- **Dataset (InMemoryDataset)**
- **ClusterData , ClusterLoader**
- **Batch**
- **NeighborSampler**
- **DataLoader**

Available datasets in PyG (**torch_geometric.datasets**)

- **Planetoid**
- **TUDataset**
- **Cora**
- ...and many others

01 Last tutorial recap

Pytorch Geometric most common classes for
Data handling and manipulation (**torch_geometric.data**):

- **Data**
- **Dataset (InMemoryDataset)**
- **ClusterData , ClusterLoader**
- **Batch**
- **NeighborSampler**
- **DataLoader**

Available datasets in PyG (**torch_geometric.datasets**)

- **Planetoid**
- **TUDataset**
- **Cora**
- ...and many others

Especially recent benchmarks!

TABLE OF CONTENTS

02 Load a new dataset

We are going to load a dataset from scratch, implementing it as an **InMemoryDataset**.

The dataset of choice is called **FRANKENSTEIN**, a mix of graphs representing molecules whose vertices are **MNIST** images (instead of atom symbols).

The dataset is available at the networkrepository site, there are plenty of graph datasets available for free!

Let's switch to the jupyter notebook...

03 Graphs' benchmarks

In 2020, two main works on graph benchmarks were released:

- Open Graph Benchmark[1]
- Benchmarking GNNs[2]

[1]Hu, Weihua, et al. "Open graph benchmark: Datasets for machine learning on graphs." *arXiv preprint arXiv:2005.00687* (2020).

[2]Dwivedi, Vijay Prakash, et al. "Benchmarking graph neural networks." *arXiv preprint arXiv:2003.00982* (2020).

03 Graphs' benchmarks

In 2020, two main works on graph benchmarks were released:

- Open Graph Benchmark[1]
- Benchmarking GNNs[2]

There are several advantages in using benchmarks:

- The repository provides a collection of datasets
- Standardized train/validation/test split
- Leaderboards

This helps reproducibility and comparison between different methods.
Both frameworks provide datasets and code infrastructure to run the models.

[1]Hu, Weihua, et al. "Open graph benchmark: Datasets for machine learning on graphs." *arXiv preprint arXiv:2005.00687* (2020).

[2]Dwivedi, Vijay Prakash, et al. "Benchmarking graph neural networks." *arXiv preprint arXiv:2003.00982* (2020).

03 Open Graph Benchmark

The datasets are chosen based on three main (orthogonal) aspects:

- Scale
- Tasks
- Domains

03 Open Graph Benchmark

The datasets are chosen based on three main (orthogonal) aspects:

- Scale
- Tasks
- Domains

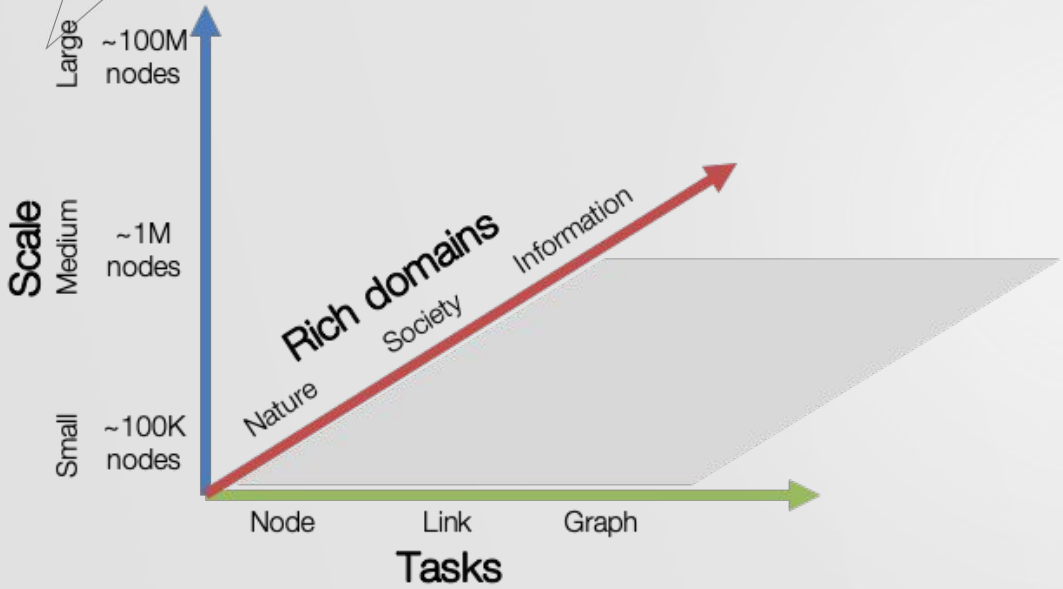

Image taken from the OGB site

03 Open Graph Benchmark

The datasets are chosen based on three main (orthogonal) aspects:

- Scale
- Tasks
- Domains

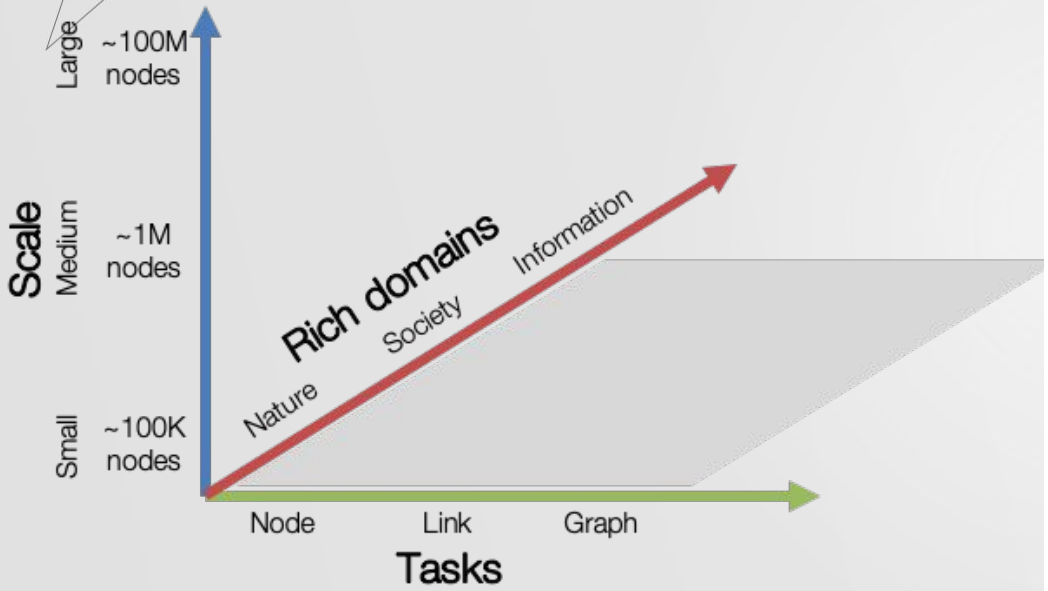

Image taken from the OGB site

Task	Node property prediction ogbn-		
Domain	Nature	Society	Information
Small		arxiv	
Medium	proteins	products	mag
Large		papers100M	

Task	Link property prediction ogbl-		
Domain	Nature	Society	Information
Small	ddi	collab	biokg
Medium	ppa	citation2	wikikg2
Large			

Task	Graph property prediction ogbg-		
Domain	Nature	Society	Information
Small	molhiv		
Medium	molpcba / ppa		code2
Large			

Table taken from the publication

04 Benchmarking GNNs

- Focused on small and medium-scale datasets (6K to 7M nodes)
- Node, edge and graph classification; graph regression
- 8 datasets of from different domains
- Solid code base for comparing GNNs
  using Pytorch and DGL (Deep Graph Library)

Domain & Construction	Dataset	#Graphs	#Nodes	Total #Nodes	Task
Chemistry: Real-world molecular graphs	ZINC	12K	9-37	277,864	Graph Regression
Mathematical Modelling: Artificial graphs generated from Stochastic Block Models	PATTERN CLUSTER	14K 12K	44-188 41-190	1,664,491 1,406,436	Node Classification
Computer Vision: Graphs constructed with SLIC super-pixels of images	MNIST CIFAR10	70K 60K	40-75 85-150	4,939,668 7,058,005	Graph Classification
Combinatorial Optimization: Uniformly generated artificial Euclidean graphs	TSP	12K	50-500	3,309,140	Edge Classification
Social Networks: Real-world citation graph	COLLAB	1	235,868	235,868	Edge Classification
Circular Skip Links: Isomorphic graphs with same degree	CSL	150	41	6,150	Graph Classification

Table taken from the publication