Topics in Deep Learning

CS7.602

Charu Sharma

4 Jan 2024

Pre-requisites

- Basics of machine learning, deep learning, linear algebra:
  - Statistical Methods in AI (SMAI) course at IIIT-H
- Familiarity with python and some deep learning framework.
  - PyTorch or TensorFlow (or similar)
  - (start learning python --> numpy --> pytorch if you are unfamiliar)
- Familiarity with graphs

Class Schedule

- Class lecture: Mon and Thu: 11:40-1:05am (85 minutes)
- Tutorial: Once in two weeks (not yet decided)
- For any queries, ask in the class or on Moodle.
- One instructor
- Two TAs

About

- What are graphs?
- How do we represent data with graphs?
- Why do we need to represent data with graphs?
- Usage of graphs in traditional machine learning and deep learning algorithms
- Usage of graphs for real-world applications
- Lots of applications from various domains
- Deal with all kinds of data in the course

Course Outline

- Introduction to ML for Graphs, Applications
- Basics of Networks and Graphs
- Node and Graph Embeddings
- Node and Graph Classification
- Link Prediction and Relation Prediction
- Clustering and Community Detection
- Graph/Subgraph Matching
- Traditional and Heuristic Methods
- Graph Kernel-based Methods
- Random Walk-based Methods: DeepWalk, Node2vec
- Graph Laplacian and Spectral Methods
- Concept of Graph Neural Networks
- GCN
- GraphSAGE, GIN
- DGCNN (point cloud network)
- Applications of GNNs
- Knowledge Graph Embeddings
- Applications of KG Embedding Methods
- Attention Model: GAT
- Transformers
- Graph Generative models
- Graph RNN

Tutorial Sessions

(subject to small changes)

- Networkx, Plotting graphs, types, and properties
- Examples of graphs in networkx and its properties
- Pytorch Geometric Introduction, DGL
- Graphs to Tensor
- Datasets- SNAP library, network repository
- GNN implementations using PyTorch Geometric
- GCN, GraphSAGE
- GAT, Graph Transformer

Evaluation Criteria

Type of Evaluation	Weightage (in %)
Quiz - 1 or Assignment - 1	10
Quiz - 2	10
Assignment - 2	10
Project (proposal + presentation + report + core work)	60 (10 + 10 + 10 + 30)
Others (class activity, SOTA seminar)	10

Project evaluation

- Form teams of 2-4 (generally 3) members. Larger teams will lead to higher expectations.
- **10 points:** Proposal: 1 page + refs; Write about what you want to do, something that you think is achievable in 2-2.5 months.
- **10 points:** Final report: 2-5 pages including results, figures and tables. Proofs + refs extra. Describe the main contribution and findings. Reference previous work for everything else. Use the NeurIPS paper LaTeX format. 2 pages of text, 3 more pages of figures and tables and references.
- **10 points:** Final presentation (5 slides) OR video (4 minutes) OR poster (1 A0 size)
- **30 points** for core research work. If you do more, this can be used to offset scores in other parts of the project or course.
    - A: (Up to 10 points) Re-implementation of code + main experiment OR re-creation of several experiments using existing code. (Minimum)
    - B: (Up to 10 points) Additional insightful ablations, experiments, analysis, not mentioned in the original paper. (A+B)
    - C: (Up to 15 points) New convincing ideas that (unfortunately) did not work. (A+B+C)
    - D: (Up to 20 points) New working idea, which is publishable in a small conference like ICVGIP. (A+B+C+D)
    - (Up to 30 points) For theoretical advances in graph representation learning.
    - Expect changes to the evaluation in exceptional cases that go beyond some of the areas mentioned above. The goal is to assess overall contributions to the work.

Picking a problem

1. Pick a paper, reproduce the results, implement new ideas. (Novelty in the experiments/algorithm/method/ way of solving)

2. Pick a problem / application, take its recent baseline and do 1. (The problem should be uncommon otherwise there would be many baselines) (Novelty in the experiments/algorithm/method/ way of solving)

3. Introduce a new problem to be solved by graph-based methods. (Novelty in the problem itself)

Suggestion: Pick a less-explored problem/area.
Go to top-tier conference's 2023 paper list, search for keyword "graph".

Late submission policy

- Life gets in the way of assignments and some other deliverables
- We understand, and will give up to 3 days of late submission for any reason
- Use them judiciously
- Submissions beyond the 3 days will be graded at a maximum of 50%
    - Beyond 7 days we will not be able to grade them
- The TAs will create a form to ask permission for late submissions. Please use that to register before the submission deadline.

Plagiarism!❌

- Any kind of plagiarism is **strictly prohibited**.
  - We will check for plagiarism.
  - Using someone else's ideas or work and represent as your own. This goes especially for the project ideas.
  - Copying or allowing to copy for all the quizzes, assignments, and project.
  - Sharing ideas/ giving or receiving help in the project and assignments. Life is collaborative, give fair attribution.
- You are encouraged to discuss and learn from each other, but not during the exams :)
- Copying from ChatGPT is also considered as plagiarism!

References

- Book: Graph Representation Learning by William L. Hamilton
- Research papers
- Other online material will be shared as we go through the course
- Courses
  - CS224W: Machine Learning with Graphs by Stanford
  - Graph Representation Learning by McGill University

About me: Charu

- Assistant Professor, ML Lab
- Research interests: Geometric Machine/ Deep Learning (graphs and point clouds), Topological Data Analysis.
- Have worked on graph matching problem for images (planar + spherical), graph neural networks, knowledge graphs, point cloud representation learning, few-shot learning, self-supervised learning.

Goals of the course

- Learn about machine learning with graphs
- Appreciate and understand models at various levels of abstraction
- Have fun exploring graph learning methods by applying it in a project!

Questions?

Extras

Scribing

- The goal of this activity is to create class notes in groups of ~5 students. After a set of 2-3 lectures a new group will be asked to cover some of the material introduced in the class.
- Deliverable: A Colab notebook that explains the material + includes some explanations using code is desirable.
- Evaluation: Clarity of the notebook in terms of completeness