Spring 2021 CS 247 Project Guidance

The course project aims to encourage students to explore more about data mining. For this project, feel free to use any programming language you are comfortable with.

1. Important Dates

- Group Formation Deadline: April 10 (Saturday), 11:59 pm
- Project Proposal Deadline: April 29 (Thursday), 11:59 pm
- Project Presentation:
  - Session 1: June 1 (Tuesday), 10 am - 11:50 am
  - Session 2: June 3 (Thursday), 10 am - 11:50 am
- Project Report Deadline: June 3 (Thursday), 11:59 pm

2. Grading Scheme and Submission

Project accounts for 35% of the total score, and it consists of proposal, presentation, and report. For each part, please submit the listed files. **All the submissions should be made on CCLE**. Please note that, **one group, one submission.** Also, **Late Submission is Not Allowed**.

- **Proposal** (5%): please submit the proposal report (PDF form)
- **Presentation** (10%): please submit your slide (PDF form)
- **Report** (20%): please submit your final report (PDF form) and code (include a link to access your code in the abstract of your final report, the code should be runnable and should include a clear readme file)

Please note that, students in the same team **MAY NOT** get the same score. Every student should fill in a peer evaluation form at the end of this quarter. If your team members point out your lack of contribution and you can not prove your own contributions, we may reduce your score.

3. Group Formation

- You are expected to form a group of 3-4 students and collaborate on your course project.
- You need to decide how to distribute the workload by yourselves, please make sure each member has enough contributions, **NO FREE RIDERS!**
- Please fill in the following sign-upform before **April 10 (Saturday), 11:59 pm**, you would NOT be able to edit this form after that:

  https://docs.google.com/spreadsheets/d/1r5W4VMIzQDM4Hjwii0Q1CtkCpeXb7djZ3J-oF9natKA/edit?usp=sharing
- How to find a teammate:
  - https://piazza.com/class/kmtlykd2ut76x4?cid=11
  - Check the sign-up form, email the group leader whose team has available position

- Join our KDD Cup team (See section 4.1 for details)
- Please email TA (wyw10804@cs.ucla.edu) before **April 11 (Sunday) 11:59 pm** if you still can not find a group. TA would randomly team-up students who do not have a group by April 12 (Monday).

4. Choose A Topic

The project is **open ended**, you can explore any topic related to CS247 contents and should be related to **text**, **graph**, or **recommender system**. The project can be either a research work or an application work. We provide 2 potential source of ideas in the following, you are also encouraged to come up with new project topics.

4.1 Source 1: Data Science Competitions

- Kaggle:

  One way to find ideas for your projects is by exploring the topics of various data science competitions in Kaggle(a platform holding data science competitions) : https://www.kaggle.com/competitions.

- KDD Cup:
  - Please refer to: https://ogb.stanford.edu/kddcup2021/ to get detailed information about this competition.
  - Our lab has a group working on this year's KDD Cup competition:
    - If you are interested, please email Ziniu Hu (bull@cs.ucla.edu) before **April 7 (Wednesday), 11:59 pm**. In the email, please:
      - use this title: CS247 Student Interested in KDDCUP - Your Name
      - briefly introduce yourself (1-2 sentences)
      - briefly introduce your previous experience that might be helpful for this project (<=5 sentences)
      - attach your Curriculum Vitae to the email

    Ziniu will get back to you before April 8 (Thursday). Please note that, we only have limited headcounts, so please apply soon if your are interested. We will post a Piazza notice when the positions are filled.

4.2 Source 2: Extension of Existing Papers

- Another way to find ideas for your projects is by exploring the latest methods for various classic graph-related tasks. You can follow the following procedures:
  - **Step 1**: find a classic task you are interested in.
    - It can be node classification, graph classification, link prediction, recommender systems, etc.
  - **Step 2**: find some benchmark datasets.
    - We recommend small-scale datasets, which are relatively easier to handle. We would not provide computation resources (such as GPU/CPU).

- **Step 3:** implement one recent model published in top-notch conferences
    - Check the following conferences. To find the related papers, you can search for the key words such as "graph", "recommend", etc.
        - **Machine Learning Conferences**:
            - NeurIPS 2020: https://papers.nips.cc/paper/2020
            - ICML 2020: http://proceedings.mlr.press/v119/
            - IJCAI 2020: https://static.ijcai.org/2020-accepted_papers.html
            - ICLR 2020: https://openreview.net/group?id=ICLR.cc/2020/Conference
            - AAAI 2020:
                - Application: https://www.aaai.org/Library/AAAI/aaai20contents-issue01.php#3
                - Knowledge Representation and Reasoning: https://www.aaai.org/Library/AAAI/aaai20contents-issue03.php#14
                - Machine Learning: https://www.aaai.org/Library/AAAI/aaai20contents-issue04.php#15
        - **Data Mining Conferences**:
            - KDD 2020: https://www.kdd.org/kdd2020/proceedings/
            - WSDM 2020: https://www.wsdm-conference.org/2020/accepted-papers.php
            - CIKM 2020: https://www.cikm2020.org/accepted-papers/accepted-research-papers/
            - WWW 2020: https://dl.acm.org/doi/proceedings/10.1145/3366423
        - **Natural Language Processing Conferences**:
            - ACL 2020: https://acl2020.org/program/accepted/
            - EMNPL 2020: https://2020.emnlp.org/papers/main
    - **Step 4:** tune the model, compare their performance to the other baseline models, discuss its pros and cons, and try to improve the model.
        - you don't have to implement every baseline models,
            - many baselines have public implementations, and you can download and run them
            - as long as you can make sure the experimental settings atr identical, you can use the results of the baseline models reported in literatues
- You may find this site helpful: https://paperswithcode.com/. You can search the task you are interested in, then you would be able to find some datasets and some classic/state-of-the-art methods (including link to their paper and code). Some examples:
    - Node Classification: https://paperswithcode.com/task/node-classification
    - Graph Classification: https://paperswithcode.com/task/graph-classification
    - Link Prediction: https://paperswithcode.com/task/link-prediction
    - Recommender Systems: https://paperswithcode.com/task/recommendation-systems

4.3 Sample Projects from Last Year

- Sample 1: Graph Adversarial Attack and Defense
    - Inspired by last year's KDD competition, focus on attack side
    - Use Graph Convolutional Network (GCN) as target model, propose and apply some attack strategies and discuss possible defense strategies
- Sample 2: Long-Tail Recommendation System
    - Self-defined problem
    - Propose solutions to long-tail recommendation system problem
        - Long-tail problem: for new items in the system, they have very few interactions so it would be hard to be discovered and be recommended to users.
- Sample 3: Novel COVID-19 Drug Discovery Using Deep Learning And Knowledge Graph Representations for Drug-Target Interaction
    - Self-defined problem
    - Propose framework for drug discovery, the framework integrates deep learning and knowledge graph representation techniques
- Sample 4: Differentiable Architecture Search for Graph Neural Network
    - Self-defined problem
    - Propose an approach to automatically design the graph neural networks' architecture

5. Project Proposal (5%)

5.1 Length and Format

- An **up to 3 pages** project proposal (exclude references) is due on April 29 (Thursday).
- You are required to follow the **"Formatting Instructions for CS247 Project Proposal and Project Report (Spring 2021)"** (uploaded to CCLE).
    - in this Latex template, the `main.tex` is a template for the proposal/report, it includes some commands that you can use. The `reference.bib` contains a template for reference entries. You only need to modify these two files.

5.2 Contents Requirement

- The project proposal should include the following parts, please carefully think about the listed questions:
    - Problem Statement
        - what is your problem formulation?
        - why is this problem important and interesting?
        - what are the challenges in this problem?
    - A comprehensive literature review
        - what are the typical methods to solve your problem?
        - what are the state-of-the-art methods to solve your problem?
        - for those methods, what are their pros and cons?

        - if you plan to use some complex techniques/methods in your own method, please also discuss these techniques/methods.
    - Work plan
        - what are the data you plan to use?
        - which evaluation metric do you plan to use?
        - what are the baselines you plan to compare against?
    - Potential schedule
    - Division of work (plan)
    - Reference

5.3 Grading Criteria

- Writing of your report: 1%
- Problem statement: 1%
- Literature review: 1%
- Work plan: 1%
- A readonable schedule and work division: 1%

6. Project Presentation (10%)

6.1 Participation

- Every group should make a presentation for their project and should take questions from audience (i.e. presentation + QA).
- **Every regular student is expected to attend in the presentation**, please email instructor and TA in advance if you are not able to attend the presentation. **MSOL students are encouraged to attend the presentation but not mendatory.**
    - For MSOL group (i.e. all group members are MSOL students), you can upload a video of your presentation and do the QA in the form of a CCLE assignment.
    - For regular group (i.e. all group members are regular students), you are required to give live presentation and do live QA.
    - For hybrid group (i.e. some group members are MSOL students while others are regular students), regular students can give live presentation and do live QA in representation of your whole group.
        - Suggestion: If the MSOL students can not join the live presentation and live QA, they are suggested to contribute more to the report writing or other aspects of your project (so that the workload is balanced).

6.2 Length and Order

- The length of the presentation depends on the number of groups (10--20min) and will be announced at least one week before your presentation.
- We will send out a doodle poll to set up the order of the final presentation at least one week before your presentation.

6.3 Contents Requirement

- The final presentation is required to include:
    - A clear introduction
    - A comprehensive literature review
    - A precise explanation of your algorithm/model/data pipeline
    - Solid and thorough experimental evaluation and comparisons with existing approaches
    - A clear conclusion

6.5 Grading Criteria

- Clarity of your slides: 3%
- Clarity of your presentation: 4%
- Handling questions from audience: 3%

7. Project Report (20%)

7.1 Length and Format

- An **up to 8 pages** project report (exclude reference) is due on June 3 (Thursday).
- Please follow the same formatting instructions as in the project proposal.

7.2 Contents Requirement

- You can reuse your project proposal as part of your final report.
- The final report is required to include:
    - A succint but clear abstract, please include the link to access you code at the end of abstract.
    - A clear introduction
    - A comprehensive literature review
    - A precise explanation of your algorithm/model/data pipeline
    - Solid and thorough experimental evaluation and comparisons with existing approaches
    - *[**optional**] A discussion of your work's limitation and how your work can be further extended*
    - A clear conclusion
    - Division of work
    - References

7.3 Grading Criteria

- Writing of your final report: 2%
    - Is your report well-organized?
    - Are your main ideas clearly delivered?
- Novelty of your project: 2%
    - Are there any new techniques/methods developed in your project?
- Clarity of Introduction: 3%

    - Is the problem well-motivated?
    - Is the problem well-formulated?
    - Are the challenges well-discussed?
- Literature review: 3%
    - Are the related works well-summarized and well-discussed?
- Method: 4%
    - Is your method clearly introduced?
    - Is your method solid?
    - Is your method effective?
- Evaluation: 4%
    - Is your experiment setting clear and reasonable?
    - Do you compare with proper baselines?
    - Are your experimental results well-discussed?
- Clarity of Code: 2%
    - Is your code bug-free?
    - Is your code easy to understand and use?
    - Does your code include a clear readme file which tells people how to use your code?
- **Please note: If you do not submit your code, there would be a -10% penalty.**