Project Ideas for CS328-2024

Anirban Dasgupta

The following are suggested list of projects. In case you are not picking from these, you have to meet Anirban to explain your idea first.

In all of these you are free to use whatever programming language you want. In some cases, e.g. when using MapReduce, you will have a restricted set of choices e.g. {Java, Python}. In general, sticking to python might be best, and you can use libaries e.g. scipy. Please make a github account where you will need to upload your project plan and the code on a regular basis. The following is just the initial summary of the idea. The exact direction in which the project will go will be determined by you. Please schedule a meeting with Anirban **before midsem break** to get more background and to make some concrete goals and identify some datasets. You will have to do (likely biweekly) updates on i) the exact problem definition ii) the strategies you are trying out, iii) your coding platform and iv) the datasets. The schedule for these updates will be shared in class.

# KDD-cup 2014

The task is to predict which funding requests should have A\* rating. More details and dataset are available at [Kaggle link.](https://www.kaggle.com/c/kdd-cup-2014-predicting-excitement-at-donors-choose)

# Build a Recommendation System

The aim of this project is to build a semi-realistic recommendation system, for books, movies, news articles, IITGN courses, whatever. You should be doing the following— have a way to collect initial set of preferences. You can assume that each user rates, say 10-20 of the presented items in a scale of 1-5. Similarly, it is open whether you want to collect other features for each of the items (e.g.crawl their IMDB page, use the content of the articles etc.)

You can also take in some properties of the user such as age/gender etc. Then for each user who asks, you can suggest a ordered list of 5 recommendations. Each user will then rate the presented suggestions, again the 1-5 scale, and you will evaluate your suggestions.

A specific case for this could be building a recommendation systems for professors to do project under. Given the student’s interest, and given the professors’ research interest (as obtained from the abstract of his/her recent papers), build a recommendation system for 399/499 projects. You can also output a student to advisor matching by taking as input the number of students each faculty is willing to take.

# Pandemic Data Science

A number of data analysts over the world have engaged with the pandemic data in an effort to help out. CoViD data available at number of source. Some are listed here

1. [Kaggle](https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset)
2. [CoViD19-India](https://api.covid19india.org/)
3. [JHU data sources](https://github.com/CSSEGISandData/COVID-19)
4. [Links from AnalyticsIndiaMag](https://analyticsindiamag.com/top-valuable-datasets-for-covid-19-researchers/)

Some possible questions are— identifying specific anomalies in the reported data (there are plenty), can we find correlation between case growth and other health indicators of specific areas (at district or block level for instance), can we predict the growth of the epidemic, etc; if you can get hold of data related to hospital resources, there are important questions to be answered about which areas should be boosting their hospital beds soon. Can you find, in a data-driven manner, why certain regions did much better than others (e.g. Bangalore vs Mumbai or Delhi)– apply causal modeling to identify some potential factors?

# Measuring Scholarly Output

Ranking universities is very much in vogue, for good or for worse. We will explore the scholarly output of, say, the Computer Science departments at various Indian Institutions.CSrankings.org has a nice crowdsourced data of CSE professors and their DBLP pages in their GitHub page. Will be interesting to see whether there are other such resources. Google Scholar can also be crawled for institution based sources, e.g. [this is the page for IIT Bombay faculty.](https://scholar.google.com/citations?view_op=view_org&hl=en&org=15559271020991466530)

# Mining the bitcoin transaction network

The bitcoin transaction network is public, and a part of the data is available in the following url: [http://www.vo.elte.hu/bitcoin/.](http://www.vo.elte.hu/bitcoin/) While the user names are hidden, we wish to see if it is possible to understand the community structure in the bitcoin network and thereby draw any possible inferences, e.g. about flow of funds. A number of community analysis tools are available in packages e.g. SNAP and networkX.

The following work was key in starting the work on analyzing and understanding the structure of the web graph:

[https://www.cis.upenn.edu/ mkearns/teaching/NetworkedLife/broder.pdf,](https://www.cis.upenn.edu/~mkearns/teaching/NetworkedLife/broder.pdf) in hindsight it looks simple. What would be corresponding structure for the bitcoin network?

One other task here could be to understand the data, formalize the questions that could be linked to community detection, and then try out various community detection tools in the analysis. Here are two examples of such empirical study:

[Community Detection and Analysis in the Bitcoin Network.pdf,](http://snap.stanford.edu/class/cs224w-2015/projects_2015/Community_Detection_and_Analysis_in_the_Bitcoin_Network.pdf) and [Empirical analysis of bitcoin network.](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0086197)

Alternately, you could try to find which nodes are “important”. This is typically done by defining measures of network centrality. There are existing measures of centrality: [lecture slides on centrality.](http://www2.unb.ca/~ddu/6634/Lecture_notes/Lecture_4_centrality_measure.pdf) In order to use these in the bitcoin network, you could try to adapt some of them to the temporal setting.

# Defining reputation in signed networks

We often get networks in which edges denote notions of trust or mistrust. Couple of examples are given in [http://snap.stanford.edu/data/soc-sign-bitcoinotc.html](http://snap.stanford.edu/data/soc-sign-bitcoin-otc.html) or [http://snap.stanford.edu/data/soc-RedditHyperlinks.html.](http://snap.stanford.edu/data/soc-RedditHyperlinks.html) Use these to try to formalize and answer the following questions: how do we come up with an unified definition of trustworthiness? Can we define notions of which communities are in conflict with one another? Feel free to get ideas from the papers linked to in the above pages. In terms of a global measure of importance, you could try to adapt the “network centrality” measures to this setting ([lecture slides from a class on centrality)](http://www2.unb.ca/~ddu/6634/Lecture_notes/Lecture_4_centrality_measure.pdf). You could also try to adapt personalized pagerank to signed networks— two sources of personalized pagerank are here— [https://github.com/danieljunhee/Tutorial-on-Personalized-PageRank,](https://github.com/danieljunhee/Tutorial-on-Personalized-PageRank) [https://arxiv.org/abs/1006.2880.](https://arxiv.org/abs/1006.2880) There are more references in the tutorial.

# Using hypothesis testing to identify significant network properties

We will cover the basics of hypothesis testing in our course. For now, please feel free to look up the Wikipedia page ([https://en.wikipedia.org/wiki/Statistical](https://en.wikipedia.org/wiki/Statistical_hypothesis_testing) [hypothesis](https://en.wikipedia.org/wiki/Statistical_hypothesis_testing) [testing)](https://en.wikipedia.org/wiki/Statistical_hypothesis_testing). While the basics are fairly simple, there are a number of interesting questions that arise when we try to draw conclusions using hypothesis testing principles. Few are as follows: i) what do we do when we are trying to use hypothesis testing to find “anomalous subnetworks”, what happens when we need to do lots of different hypothesis testing etc. There are a number of important settings where these questions become important, biological data being one of them. Here are few papers that talk about it. Authors [here](https://www.andrew.cmu.edu/user/lakoglu/pubs/OddBall_cameraready.pdf) propose an algorithm OddBall that uses various properties of the network such as density, weights, eigenvalues, ranks and build rules based on these properties to detect anamolous nodes. The following work is an interesting combination of hypothesis testing and network

mining, applied to biology: [https://www.liebertpub.com/doi/abs/10.1089/cmb.2010.0265.](https://www.liebertpub.com/doi/abs/10.1089/cmb.2010.0265)

Here is another: [on mutated subnetwork discovery.](http://drops.dagstuhl.de/opus/volltexte/2018/9320/pdf/LIPIcs-WABI-2018-18.pdf)

Our aim in this project will be to try to reproduce this, using same or similar dataset. Such datasets may or may not be biological in nature. An example is the following paper which proposes a particular method to distinguish between the case when user-1 is similar to user-2 and hence acting similarly, versus the case when user-1 is influencing user-2 into doing a similar action: [https://www.cs.purdue.edu/homes/neville/papers/lafond-neville-www2010.pdf.](https://www.cs.purdue.edu/homes/neville/papers/lafond-neville-www2010.pdf)

# Sampling to speed up various clustering algorithms

Suppose we are given a large clustering problem to solve. Rather then solve it with all the data points, we can cleverly downsample the dataset so that if we find a clustering of the smaller dataset, we will be able to get a clustering that has cost and quality very similar to that of the original data. Such samples are called “coresets”. Our aim will be to experimentally study such coresets. This paper provides a pretty simple algorithm that should not be too hard to implement:[https://las.inf.ethz.ch/files/bachem18scalable.pdf.](https://las.inf.ethz.ch/files/bachem18scalable.pdf) You could try to compare this against other ways of sampling, including [volume sampling,](https://www.theoryofcomputing.org/articles/v002a012/) [leverage score sampling,](https://www.stat.berkeley.edu/~mmahoney/pubs/coherence-jmlr12.pdf) uniform sampling, norm based sampling etc.

# Towards building a scalable graph stream library

We will study different streaming algorithms when the data represents counts. Similar questions can be asked where the stream represents the set of edges of a graph, and we want to estimate various properties of the graph without storing all the edges in memory. Example problems here include : estimate degree distributions using frequency sketches e.g. CM sketch and Count Sketch, creating a sketch that can be queried for the maximum density subgraph, or for the community structure. While there is quite a bit of theoretical work on graph streaming, much of it is not practical. The goal for this project would be to identify a number of simple sketching algorithms and compare their effectiveness on specific queries related to graph structure.

* <https://arxiv.org/abs/1703.02625>
* [A nice survey](http://people.cs.umass.edu/~mcgregor/papers/13-graphsurvey.pdf)
* [KDD paper](https://www.microsoft.com/en-us/research/wp-content/uploads/2012/08/kdd325-stanton.pdf)

# Learning representations for use as LSH

As discussed in class, while LSH is great way for answering queries about nearest neighbors, it can be improved by making the hash functions data dependent. In this project, we can investigate whether we can setup a learning problem that learns a good representation to compare different images or documents. A number of pre-trained networks are already available. One possible goal could be to take an existing representation for an existing dataset and build a LSH on top of it. We can compare the performance (precision, recall) of different LSH techniques on a dataset.

• Lot of excellent material in this recent survey: [http://cs.nju.edu.cn/lwj/L2H.html,](http://cs.nju.edu.cn/lwj/L2H.html) specifically about deep-hashing.

# Analyzing the Social Media in India

A number of nice datasets have been curated [here, at IIITD.](http://precog.iiitd.edu.in/resources.html) You can try to define some data analysis questions with respect to how politicians react with respect to various events, how politicians influence one another.

Another version of this project can be a real-time one. Collect important Twitter handles – at a certain frequency, scrape their most recent tweets, summarize them, present the trending keywords and hashtags (trends considering this subpopulation only). One important question is to develop classifiers for different types of hate speech, understand how hate speech propagates etc.

# Matrix Factorization in a Distributed/Federated setting

In lot of settings, the data matrix is too large to be able to store on a single machine. Or because of privacy, the different agents do not want to share the data between themselves. Can we try to calculate matrix factorizations while not exchanging the entire dataset? Someone could try out the distributed setup using Apache Spark. Another version of this problem is to consider the federated setting whether you have to make the algorithm private on top of making it distributed, i.e. no agent should be able to infer what datapoints the others have.

* [http://stanford.edu/ rezab/papers/dimsum.pdf](http://stanford.edu/~rezab/papers/dimsum.pdf)
* <https://arxiv.org/abs/1304.3162>
* <http://ai2-s2-pdfs.s3.amazonaws.com/4cb6/b9fda3f45f6747ae07b6def71a5539dd033d.pdf>
* <http://epubs.siam.org/doi/abs/10.1137/1.9781611973730.62>

# Using Sampling and Sketching in a machine learning pipeline

When the number of instances or the number of dimensions are very large, it might make sense to calculate a sketch of the training set and then to learn a classifier over the sketch. We could reduce training time using two different technique (or a combination). First, like the “coresets” introduced in Project 6, we could sample some amount of the E.g. if the number of features is very large, we can use sparse projection/hashing to reduce the dimension and then learn classifiers in the reduced space. The goal for this project would be to take a specific set of classifiers (e.g. SVM) and then experiment with the effect of the different sketching techniques on the classifier accuracy and performance.

* The following papers show that the feature space can be reduced significantly by using a technique called random projection/hashing: the

[hashing-trick paper,](http://alex.smola.org/papers/2009/Weinbergeretal09.pdf) [a popular article,](https://medium.com/value-stream-design/introducing-one-of-the-best-hacks-in-machine-learning-the-hashing-trick-bf6a9c8af18f) and extensions– [https://arxiv.org/abs/1105.4385,](https://arxiv.org/abs/1105.4385) [HashedNet](https://www.cse.wustl.edu/~ychen/HashedNets/) (application to NNs). <http://www.jmlr.org/proceedings/papers/v31/paul13a.pdf>and <https://arxiv.org/abs/1105.4385> which are both applications to SVMs.

On the other hand, if you want to create coresets for classifiers, here is one way to do it.

* [https://arxiv.org/abs/1708.03835.](https://arxiv.org/abs/1708.03835)

# Sparsifying networks while preserving properties

As discussed before, if the graph is too large, we might want to sample the edges and maintain only a subset. Algorithms for community detection or estimating various graph properties, which need to run over the entire data, can now be run only over this subset of edges. There are different sampling strategies known in theory, but most of them have not been evaluated for most of the community detection strategies. The goal of this project would be to take a class of community detection algorithms, as well as a set of graph sampling techniques and then evaluate the performance on given datasets.

* Here is a [sparsification technique that preserve triangles.](http://emis.impa.br/EMIS/journals/JGAA/accepted/2011/TsourakakisKolountzakisMiller2011.15.6.pdf) Counting triangles is often an useful feature in friend recommendation systems. • One way to sparsify networks while preserving community structure: <https://arxiv.org/pdf/1701.07221.pdf>
* [Modularity based clustering](https://www.uni-konstanz.de/mmsp/pubsys/publishedFiles/BrDeGa08.pdf) is a much used method. We can try to see how it is affected by sparsifications.

# Sketching and Sampling based methods for Novel Matrix Factorization

There has been a number of recent results of making SVD/PCA methods more efficient by employing sampling and and random projection type techniques. Often, however, the task of prediction calls for a different matrix factorization, e.g. factoring into non-negative matrices, max-margin matrix factorization, augmenting SVD based methods to handle incomplete entries. The aim of this project is to study empirically the effect of some of the sampling techniques on specific sets of matrix factorization techniques. We will see some of the sketching/sampling techniques in class.

Some of the interesting matrix factorization methods are n max margin matrix factorization (described in [http://ttic.uchicago.edu/∼nati/mmmf/)](http://ttic.uchicago.edu/~nati/mmmf/), ([nonnegative factorization](http://perso.uclouvain.be/paul.vandooren/ThesisHo.pdf) (chapter 2 of [this thesis)](http://perso.uclouvain.be/paul.vandooren/ThesisHo.pdf)), other variants are also available. For most of these matrix factorizations, existing code is available, so no need to implement those afresh. We will only need to implement the sampling methods and the evaluation techniques.

# On Learned Sketches

While we will study sketches that come with provable guarantees, their construction is “data-oblivious”, i.e. does not depend on the properties of the dataset so much. This paper – [On Learned index structures](https://www.cl.cam.ac.uk/~ey204/teaching/ACS/R244_2018_2019/papers/Kraska_SIGMOD_2018.pdf) – shows that a lot can be gained by learning the index structure of a DB from the data. Similar results can hold for sketches. I am describing the case of Bloom filters below, but other sketches can also be picked up for this.

**Learned Bloom filter.** A Bloom filter is a data structure designed to tell you, rapidly and memory-efficiently, whether an element is present in a set. Recently there is a key idea getting popular to do better than standard Bloom filters by learning the set. The idea is to train an oracle which provides the probability that an element is in the set as a binary classification problem, using elements in the set as positive examples, and randomly or selectively generating negative examples. We could then use the oracle to filter out elements that are extremely likely to be in the set and fall back on a standard Bloom filter to catch any false negatives that were not identified by the oracle. This is very new area and it would b a good idea to review small number of works in this area

* [Learned Bloom filters](https://papers.nips.cc/paper/7328-a-model-for-learned-bloom-filters-and-optimizing-by-sandwiching.pdf)
* [Neural bloom filters](http://proceedings.mlr.press/v97/rae19a/rae19a.pdf)
* [Partitioned Learned bloom filters](https://arxiv.org/pdf/2006.03176.pdf)
* [Here is a nice blog to get started](https://kevz.me/learned-bloom-filters/)

# Data Reduction and/or Compression of Neural Networks

Neural Networks are used extensively for quite some time now. However the huge size of neural network models and/or large training time sometimes make their use infeasible. To overcome this problem, the neural networks are compressed i.e. parameters are reduced without significantly affecting performance. Other popular method is to reduce the amount of data required to train the networks. Significant work has been done in these areas, A review of these methods along with new insights will be a useful contribution. Here are a few references • [Baykal et al.](http://www.mit.edu/~baykal/publications/iclr2019.pdf)

* [Sinha et al.](https://arxiv.org/abs/1910.13540)
* [Dubey et al.](https://arxiv.org/abs/1807.09810)

# Graph representation learning

There are a number of methods for graph representation learning e.g. node2vec, DeepWalk, LINE and HARP. Can we develop a scalable method for dynamic graphs? Citation and transaction datasets are examples. Based on the representation, we can try to predict different properties e.g. link formation, how the number of triangles (and other subgraph counts) evolve etc.

# Fairness issues in Data Mining

A very important question is to understand whether a particular data mining / machine learning algorithm is fair. There is a nice tutorial (slides available) in KDD-2019 [available here.](https://sites.google.com/view/kdd19-fairness-tutorial) There are also a lot of nice references in the above page. [ProPublica](https://www.propublica.org/datastore/) has a lot of available datasets that can be used (some datasets are premium, but quite a few are free).

One set of questions involves understanding the bias issues in existing ML products and pipelines and systems. For instance, Google Translate is known to have certain gender biases. One possibility would be to try to analyze the bias in BERT embeddings. [Here is a paper](https://towardsdatascience.com/tackling-gender-bias-in-word-embeddings-c965f4076a10) that looks into it. What seems unexplored is, for instance, looking into specific literature (e.g. sci-fi, Indian literature). It is possible to collect a very large corpus from Project Gutenberg and do this gender/other bias analysis.

# Causal models

[This workshop](https://nips.cc/virtual/2022/workshop/49993) provides a nice set of talks for introduction to causality. Go through selected talks (e.g., the Neural Causal model talk) and implement a neural causal model. You can also find an introduction to causality in [these slides.](https://web.stanford.edu/class/cs224w/notes/Intro_Causality.pdf)