CS341: Project in Mining Massive Datasets

Jure Leskovec
Anand Rajaraman
Jeff Ullman

CS341: Project in Data Mining

- **Data mining research project on real data**
  - Teams of **3 students** **(Use Piazza to form teams)**
  - We have room for **10-15 teams**
- **We provide:**
  - **Data**
  - **Computers** (Amazon EC2, **3k$ per team**)
  - **Mentoring:** Each group will have an assigned mentor that they meet on a weekly basis
- **You provide:**
  - Project proposals
  - Work

CS341: Schedule

- Today (3/14): Info session
  - Instructors introduce datasets, problems, ideas
- Students form groups and project proposals
- Mon 3/25: Project proposals are due
- We evaluate the proposals
- Mon 4/1: Admission results
  - 10 to 15 groups/projects will be admitted
- Tue 4/30, Thu 5/2: Midterm presentations
- Tue 6/4, Thu 6/6: Presentations, poster session

More info: http://cs341.stanford.edu

Projects: Intro

Project types:

- **Data analysis/Modeling project:**
  - Discovers interesting relationships within a significant amount of data
- **Algorithmic project that extends/builds on what we learned in class**
  - Extend/Improve/Speed-up some existing algorithm
  - Define a new problem and solve it

Projects: Proposal has to address

- **(1) What is the problem/question your team is solving?**
  - **Give a brief but precise description or definition of the problem or question**
    - **Examples:**
      - **(a)** Analyze the data to understand why editors are leaving Wikipedia
      - **(b)** Build a social recommender engine for movies
      - **(c)** Design a MapReduce algorithm for finding clusters in graphs
- **(2) What data will you use?**
  - **Why is the data you plan to use appropriate? Does it have the right labels/information?**
  - **It is ok to use your own data** (give detailed description)!
    - **Examples:**
      - **(a)** Wikipedia edit history where every action of every user is recorded
      - **(b)** We **crawled** Yelp and obtained X million reviews from Y million users
      - **(c)** We will use the Altavista web graph on X million nodes.

Projects: Proposal has to address

- **(3) How will you solve the problem? What is your plan of action?**
  - **Describe and think about your approach!**
    - What method, algorithm, technique? How will you scale it up?
    - **Be as specific as you can!**
    - **Examples:**
      - **(a)** We will create edit histories of every article. We will then compare article edit histories and argue that users are leaving since all the "easy/obvious" articles have already been written
      - **(b)** Our hypothesis is that friends have similar tastes. We will include a regularization term to a Latent Factor Rec. Sys. which will encourage neighboring users to have similar parameters
      - **(c)** We will implement a scalable Frequent itemset based approach to identify cluster seeds (complete bipartite subgraphs). In the second pass we will then use a random walk based approach to expand around the seed and extract the clusters

Projects: Proposal has to address

- **(4) How will you evaluate your method?**
  - **How will you measure performance or success of your method? What baselines will you use?**
    - **Examples:**
      - **(a)** Using insights from our analysis we will build a model that will predict how complete is the article (much the article will change in the future). We will evaluate predictive accuracy of the model
      - **(b)** We will measure **RMSE** of our system. As a baseline for comparison will use traditional latent factor recommender
      - **(c)** We will measure resource usage and execution time of our algorithm and compare it to open source algs. Metis and Graclus
- **(5) What do you expect to submit/accomplish by the end of the quarter?**

Projects: Proposals

- **Submit** to cs341-spr1213-staff@lists.stanford.edu
  - **PDF should include**
    - **Project title**
    - **Project narrative addressing the 5 Qs**
    - **Information about team members :**
      - **For each team member:** A 5 line CV/Bio about prior experience, and why are you suitable to take this course
  - **No page limit** (we don't promise to read past page 3)
  - **Due Monday 3/25 5pm Pacific time**
- **We will let you know whether you got in by Monday April 1**
- **More info at: http://cs341.stanford.edu/info.html**

Project ideas / datasets

(1) Movie Time-based metadata

- **Dataset of 10,000 movies**
- **Every scene is annotated:**
  - **Actions:**
    - Attacking, Fighting, Flying, …
  - **Locations:**
    - Airport, Garage, Gym, …
  - **Objects:**
    - Animal, Books, Drink, Drugs, …
  - **Appearance:**
    - Actor, Character, Nickname, Type
  - **Genre of the scene:**
    - Action, Family, History, War,


```
"netflix": {
   "id": "70213514",
   "genres": [
     "Action & Adventure",
     "Action Sci-Fi & Fantasy",
     "Sci-Fi Thrillers",
     "Action Thrillers",
     "Blockbusters"
   ]
 },
 "rotten_tomatoes": {
   "id": "771041731",
   "critics_score": 87,
   "audience_score": 92
 },
 {
       "hitType": "tag",
       "subTrack": "Vehicle",
       "startTime": 2682.9823,
 },
 {
       "hitType": "tag",
       "subTrack": "Driving",
       "track": "Action",
       "startTime": 2685.8498,
 }
```

(1) Movie Time-based metadata

- **Some ideas:**
  - **(1)** How films are similar to each other
  - **(2)** Trending scene types in popular films
  - **(3)** Predicting scene based Genre
  - **(4)** Learn models that automatically classify edges (e.g., adversary, friend, lover) based on the patterns of interaction between people
  - **(5)** A qualitative study of how interaction patterns etc. differ between genres, older films etc.
- **Can be combined with IMDB, Rotten Tomatoes,…**
- **If you are interested send us email and we'll share data for 1 movie with you**

(2) Survey and Poll Data

- **People's opinions and online data**
  - People survey data (from a market research agency)
  - 6B news articles & blogs collected since Aug 2008
  - We also have Twitter data (1 month complete and also 10% over 4 years)
- **How to predict / model people opinions from passively collected online data?**
- **If interested send us email and we can talk more**

(3) Online Media Search Engine

- **Online media search engine**
  - We have a collection of 6B news documents and 300M short textual phrases that appear in them
- **Goal:**
  - Build a search engine that allows for efficient querying and retrieval of these documents and phrases
    - Based on time, named entities, mutation of information
- **If interested send us email and we can talk more**

(4) Large Scale Graph Processing

- **Distributed processing of large graphs**
  - System and algorithms for processing graphs
  - We have some ideas here, **talk to us**
- **Graph anomaly detection**
  - Imagine a communication network
    - Email, phone calls (we have both)
  - Can you spot and identify anomalies in communication patterns
  - **If interested send us email and we can talk more**

(5) Wikipedia

- **Complete edit history of Wikipedia**
  - For **every single edit** there is complete information
    - http://dumps.wikimedia.org/enwiki/latest/ and check the ***meta-history*** files
- **Wikipedia article access counts:**
  - http://dumps.wikimedia.org/other/pagecounts-raw/
- **By examining edits happened we can study**
  - **Lifetime of an articles:**
    - How complete and mature is a particular article?
    - What is evolution of Wikipedia articles
  - **External events and Wiki article access patterns**

(6) Online Product Reviews

- **Ratings and reviews of Beer, Wine, Movies**
  - Plus user data, temporal information
  - http://snap.stanford.edu/data/
- **Questions:**
  - Evolution of user tastes and opinions
  - User contribution to the community
  - Beer is reviewed along **5** distinct dimensions
    - Taste, smell, feel, color and overall

Jeff Ullman

TREC Contest Coming 4/8/13

Automatically recommend new edits

1) Initialize with a target entity and info need
2) Iterate over stream of text items
3) For each, output confidence between 0, 1

Your KBA System

Content Stream

- 462M texts, 40% English
- 4,973 hourly chunks of a 10^5 docs/hour
- News, blogs, forums, and link shortening

Permanent Vs. Transient

- Goal: tell immediately whether a new item is of long-term significance or “one of a kind.”
- Example: A travel guide bought on Amazon should not stimulate them to pitch other travel guides; but Harry Potter-1 justifies an ad for Harry Potter-2.

Permanent Vs. Transient – (2)

- Other possible scenarios:
  1. News events.
  2. Web-page views.
- Key issue: it is generally easy to tell in retrospect, but the objective is to know in a very short time (milliseconds? days?) what the long-term significance is.

Synthetic Data

- Problem: it is often too easy to discover true identities from naïvely anonymized data.
  - *Naïve* = "replace names by numbers."
- Example: Netflix/IMDB databases.
- Example: Identify Facebook people from the numbers of their friends.

Synthetic Data – Goal

- Take a real database, perhaps with identities hidden.
- Generate from it another database with different data that has the same statistics as the original, but none of the connections or details from which the original identities can be discovered.

Synthetic Data – Applications

- Measure the efficiency of algorithms without having to use real data.
- Ideally, in areas like medicine, enable statistically valid answers to questions like “is there a relationship between chocolate and diabetes?” without any possibility of revealing who is diabetic.

Synthetic Data – Conundra

1. What are appropriate statistical tests to apply?
2. How do we generate synthetic data from a given database in order to meet specific tests?