ECS289: **Scalable** Machine Learning

Cho-Jui Hsieh
UC Davis

Sept 24, 2015
Course Information

- Website: www.stat.ucdavis.edu/~chohsieh/ECS289G_scalableML.html
- My office: Mathematical Sciences Building (MSB) 4232
- Office hours: by appointment (email)
- My email: chohsieh@ucdavis.edu, cjhsieh@cs.utexas.edu
- This is a 4-unit course
Goals:
- Understand the challenges in large-scale machine learning.
- Understand state-of-the-art approaches for addressing these challenges.
- Identify interesting open questions.

Course Structure:
- Pick some important machine learning problems
  (classification, regression, recommender system, ...)
- Introduce the model
- Discuss the computational challenges
- How do people scale to large datasets?

Prerequisites:
- Basic knowledge in linear algebra (matrix multiplication, inversion, ...)
- Basic knowledge in programming (C/MATLAB) for the final project.
Grading Policy

- Class participation (10%)
- 1 assignment and 1 presentation (30%)
- Midterm exam (20%)
- Final project (40%)
Final Project

Topics include:

- Develop new algorithms or improve existing algorithms
- Implement parallel machine learning algorithms and test on large datasets
- Apply machine learning to some application
- Compare existing algorithms.
- ...

Schedule:

- Final project proposal presentation **10/20**
- Final project presentation **12/1, 12/3**
- Final project paper due **TBD**
Syllabus

- Supervised Learning: Classification and Regression
- Optimization for Machine Learning
- Matrix Completion
- Semi-supervised Learning
- Ranking
- Neural Networks
What is Machine Learning?

**Training Data**
(documents, images, ...

**Test Data**
(documents, images, ...

- Train and test data are usually assumed to be iid sample from the same distribution
Training

- Linear SVM/regression: Linear hyperplane
- Kernel SVM/regression: Nonlinear hyperplane
- Decision tree, random forest
- Nearest Neighbor
- ...
Learn a model that best explains the observed data as well as generalizes to unseen data.

Scalability Issues:
- Time & space complexity of the (Training) Learning Algorithm
- Size of the Model
- Time complexity of Prediction (for real-time applications)
A simple example

- K-nearest neighbor classification
- Model size: storing all the training samples
  - 1 billion samples, each requires 1 KBytes space
  ⇒ 1000G memory
- Prediction time: Find the nearest training sample
  - 1 billion samples, each distance evaluation requires 1 micro second
  ⇒ 1000 secs per prediction
Topics in this course

- Classification
- Regression
- Matrix Completion (Recommender systems)
- Ranking
- Semi-supervised learning
Machine Learning Problems: Classification

**Image classification**
- Kit fox
- Croquettes
- Airplane
- Frog

**Hand-written digit recognition**
- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

**Spam filters**
- Gmail
  - Inbox (8,439)
  - Starred
  - Important
  - Sent Mail
  - Drafts
  - Notes
  - Less
  - Chats
  - All Mail
  - Spam (298)
  - Trash
Binary Classification

- **Input:** training samples \( \{x_1, x_2, \ldots, x_n\} \) and labels \( \{y_1, y_2, \ldots, y_n\} \)
  - \( x_i \): \( d \)-dimensional vector
  - \( y_i \): +1 or -1
- **Output:** A decision function \( f \) such that
  \[
  f(x_i) > 0 \text{ if } y_i = 1, \quad f(x_i) < 0 \text{ if } y_i = -1
  \]

**Diagram:**
- **Training:**
  - \( x_1, x_2, x_3, x_4 \)
  - \( y_1, y_2, y_3, y_4 \)
- **Feature extraction:**
  - \( x_{\text{test}} \)
  - \( y_{\text{test}} \)
Feature generation for documents

- Bag of words features for documents:

<table>
<thead>
<tr>
<th>Term</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>(international)</td>
<td>2</td>
</tr>
<tr>
<td>(conference)</td>
<td>2</td>
</tr>
<tr>
<td>(machine)</td>
<td>2</td>
</tr>
<tr>
<td>(train)</td>
<td>0</td>
</tr>
<tr>
<td>(learning)</td>
<td>2</td>
</tr>
<tr>
<td>(leading)</td>
<td>1</td>
</tr>
<tr>
<td>(totoro)</td>
<td>0</td>
</tr>
</tbody>
</table>

The International Conference on Machine Learning is the leading international academic conference in machine learning,

number of features $= \text{number of potential words} \approx 10,000$
Feature generation for documents

- Bag of $n$-gram features ($n = 2$):

  The International Conference on Machine Learning is the leading international academic conference in machine learning,

| (international) | 2 |
| (conference)    | 2 |
| (machine)       | 2 |
| (train)         | 0 |
| (learning)      | 2 |
| (leading)       | 1 |
| (totoro)        | 0 |
| (international conference) | 1 |
| (machine learning) | 2 |
| (leading international) | 1 |
| (totoro tiger)   | 0 |
| (tiger woods)    | 0 |
| (international academic) | 1 |

10,000 words $\Rightarrow$ 10,000^2 potential features
Classification

Class 2: -1

\[
\frac{2}{\|w\|}
\]

Class 1: 1

\[
w^T x = \{-1, 0, 1\}
\]

> 1 million dimensional space, > 1 billion training points
Scalability challenges

- Large number of features
- Large number of samples
- Data cannot fit into memory

  Splice-site: 10 million samples, 11 million features, $>1T$ memory

Current solutions:
- Intellectually swap between memory and disk
- Online algorithms
- Parallel algorithms on distributed systems
- Other idea?
Challenges: large number of categories

- Multi-label (or multi-class) classification with large number of labels
- Image classification — $\geq 10000$ labels
- Recommending tags for articles: millions of labels (tags)
Challenges: large number of categories

- Consider a problem with 1 million labels.
- Traditional approach: reduce to binary problems.
- Training: 1 million binary classification problems.
  
  Need **694 days** if each binary problem can be solved in 1 minute
- Model size: 1 million models.
  
  Need **1 TB** if each model requires 1MB.
- Prediction one testing data: **1 million** binary prediction
  
  Need **1000 secs** if each binary prediction needs $10^{-3}$ secs.

<table>
<thead>
<tr>
<th>Cat</th>
<th>Dog</th>
<th>......</th>
<th>Laptop</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Cat" /></td>
<td><img src="image" alt="Dog" /></td>
<td><img src="image" alt="null" /></td>
<td><img src="image" alt="Laptop" /></td>
</tr>
</tbody>
</table>
Machine Learning Problems: Regression

Line fitting

\[ y \]

\[ \hat{y}_t \]

\[ x \]

Polynomial curve fitting

(Figures from Dhillon et al)
We have $p > 20,000$ stocks

- $X_{i,t}$: the price of stock $i$ at time $t$; $x_t = [X_{1,t}, \ldots, X_{p,t}]$
- Find a function $f$ such that

$$x_{t+1} \approx f(x_t, x_{t-1}, \ldots, x_{t-L})$$

$pL$ input variables, $p$ output variables
### Netflix Problem

#### Rating Matrix

<table>
<thead>
<tr>
<th>Users</th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie 3</th>
<th>Movie 4</th>
<th>Movie 5</th>
<th>Movie 6</th>
<th>Movie 7</th>
<th>Movie 8</th>
<th>Movie 9</th>
<th>Movie 10</th>
<th>Movie 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hsiang-Fu</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cha-Jui</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Si Si</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indrjit</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kai-Yang</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Donghyuk</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ajeet</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Koje</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Raghunath</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joseph</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Figure from Dhillon et al)
Collaborative Filtering

Customers Who Bought This Item Also Bought

- A Rulebook for Arguments
  - Anthony Weston
  - Paperback
  - $9.45

- The Hidden Brain: How Our Unconscious Minds... (Figure from Dhillon et al)
  - Shankar Vedantam
  - Paperback
  - $13.45

- Outliers: The Story of Success: A BookCaps...
  - BockCaps
  - Paperback
  - $4.46

- A Dictionary of Sociology
  - John Scott
  - Paperback
  - $7.00

(Figure from Dhillon et al)
Machine Learning Problems: Recommender Systems

Latent Factor Model

\[ W \]

\[ H^T \]

(Figure from Dhillon et al)
## Latent Factor Model

$$H^T$$

$$W$$

(Figure from Dhillon et al)
Latent Factor Model

(Figure from Dhillon et al)
Recommender Systems: challenges

- Size of the matrix:
  
  billions of users, billions of items, >100 billions of observations
  
  Memory to store ratings: > 1200 GBytes

- How to incorporate Side information?
  
  User/Item profiles
  
  Temporal information, click sequence

- Prediction time:
  
  Recommend top-k items to a user:
  
  Need to compute a row of a matrix: $O(mk)$ time
  
  $m > 1,000,000,000$, $k > 500$: need $> 100$ seconds

  Recommend items to all users: 100 billion seconds $\approx 3170$ years
Ranking

- Ranking players by pair comparison
  
  Given $n$ items and a subset of pair comparisons, what’s the ranking for each player?

  Examples: Chess tournaments, . . .

  What’s the ranking?
Ranking players by group comparison

Given $n$ items and a subset of group comparisons, what’s the ranking for each player?

What’s the ranking?
Ranking

- Ranking players by group comparison

  Given \( n \) items and a subset of group comparisons, what’s the ranking for each player?

  How to form the best group?

  Examples: Halo, LOL, Heroes of the storm player ranking, . . .
Ranking: challenges

- Sample complexity: how many comparisons do we need?
- Scalability: how to compute the ranking for huge datasets?
- Side information: how to incorporate features?
Semi-supervised Learning

- Given both labeled and unlabeled data
- Is unlabeled data useful?
Semi-supervised Learning

- Two approaches:
  - Graph-based algorithm (label propagation)
  - Graph-based regularization

- Scalability: need to construct an $n \times n$ graph
  
  $n$: total of labeled and unlabeled samples

  What if $n > 1$ million?

- Extensions: can we apply similar idea to other learning algorithms?
  
  Matrix completion? Ranking?
Scalability: Need

- Information gathered today is often in terabytes, petabytes and exabytes
- About 2.5 exabytes ($= 2.5 \times 10^{18}$) bytes of data added each day
- Almost 90% of the world’s data today was generated during the past two years!
- Wal-Mart collects more than 2.5 petabytes of data every hour from its customer transactions
- Twitter generates 7TB/day and Facebook 10TB/day
How to address these scalability challenges?

Algorithmic level:
- Faster (optimization) algorithms
- Approximation algorithms — expensive to compute exact solutions
- Parallel algorithms
  - Multi-core optimization algorithms: when data can store in a single machine
  - Distributed algorithms: when data cannot fit in memory of a single machine

Modern architecture:
- High-throughput computational clusters and fault tolerance
- Tools and technologies to leverage computational resources — such as Hadoop, Spark
- Parallel programming paradigms, software and enabling easy adoption
Coming up

- Read the class website
- Next class: linear regression problems

Questions?