Introduction

Yiming Yang
Carnegie Mellon University
yiming@cs.cmu.edu

Instructor and Teaching Assistants

Faculty
• Yiming Yang
• yiming@cs.cmu.edu
• GHC 6717

TA's (office hours on the syllabus)
• Yuexin Wu
• Guokun Lai
Outline

• Course Contents Overview
• Administrative Details

Big Data Era

Based on some quick search on Google

- **WWW**: 1.8 billion web sites by 2018
- **Wikipedia**: 40 million articles with 27 billion words in 293 languages by Jan. 2019
- **Tweets**: 350,000 per minutes, 500 million per day, 200 Billion per year
- **International Patents** (PCT): 8 million patents by 2017
- **Amazon Product Reviews**: 35 million by 2013

How do we handle such huge and diversified data for massive users?
Topic Coverage

- **High-dimensional Vector Spaces (1 lecture, HW1)**
  - Scalable Indexing, Matrix Calculus Review
- **Link Analysis (3 lectures, HW2)**
  - Analyze interconnected web sites, papers, people, social impacts, etc.
- **Collaborative Filtering (3 lectures, HW3)**
  - Model users’ tastes and predict their preference over items (e.g., shopping products)
- **Matrix Factorization, Stochastic Gradient Decent (3 lectures, HW4)**
  - Unsupervised learning of dimension reduced views of data
- **Web-scale Text Classification (5 lectures, HW5)**
  - Supervised learning of topic assignments over documents
- **Learning to Rank (2 lectures, no HW)**
  - Supervised learning for search engines based on relevance judgments
- **Statistical Significance Tests (3 lectures, HW6)**
  - Evaluation, methods comparison
- **Deep Learning for Text Mimicking and Beyond (4 lectures, HW7)**
  - Latent (semantic) representations of words, documents, topics, etc.
Example: Text Classification

Text mining, also referred to as text data mining, roughly equivalent to text analytics, refers to the process of denoising and extracting meaningful information from text data. Text mining relies on techniques such as natural language processing (NLP) and machine learning methods.

Wikipedia categories form a graph

5.78 million articles in English Wikipedia, with >500,000 categories
Large Classification Taxonomies for Organizational Views of Big Data

- Web pages → over 1M categories in a hierarchy (DMOZ)
- Wikipedia articles → over 500,000 categories in a graph
- Amazon Products → 120,000 categories of videos, books, computers, software, clothing, jewelries, ...
- Medical journal articles → 20,000+ Medical Subject Headings in a 10-level hierarchy
- International Patents → 60,000+ WIPO categories
- ...

Amazon Product Hierarchy

[240 million products across 12,000 categories in Amazon store directory]
### # of categories in benchmark data sets

<table>
<thead>
<tr>
<th>Data Sets</th>
<th># of Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>LargeWiki</td>
<td>614,428</td>
</tr>
<tr>
<td>DMOZ_2011</td>
<td>27,875</td>
</tr>
<tr>
<td>OHSUMED</td>
<td>14,321</td>
</tr>
<tr>
<td>IPC</td>
<td>552</td>
</tr>
<tr>
<td>RCV1</td>
<td>137</td>
</tr>
<tr>
<td>CLEF (image)</td>
<td>87</td>
</tr>
<tr>
<td>20-News</td>
<td>20</td>
</tr>
</tbody>
</table>

---

### Large-Scale Classification Challenges

- **Scalability challenge** – training massive classifiers (one per category)
  - Need fast algorithms
  - Need parallel computing strategies for divide-\&-conquer

- **Modeling challenges**
  - Skewed category distributions where many categories do not have enough training examples
  - Need to leverage the hierarchical or graphical dependency structures to "borrowing" labeled data from richer categories for poorer categories
SVMs Solver: Pegasos

[Shalev-Shwartz et al., 2011]

Recall

\[ \ell_i(w) = \max\left(0, 1 - y_i w^T x_i\right) + \frac{\lambda}{2}\|w\|^2 \]  \hspace{1cm} (10)

\[ \begin{cases} \frac{1}{2}\|w\|^2 & y_i w^T x_i \geq 1 \\ 1 - y_i w^T x_i + \frac{\lambda}{2}\|w\|^2 & y_i w^T x_i < 1 \end{cases} \]  \hspace{1cm} (11)

Therefore

\[ \nabla \ell_i(w) = \begin{cases} \lambda w & y_i w^T x_i \geq 1 \\ \lambda w - y_i x_i & y_i w^T x_i < 1 \end{cases} \]  \hspace{1cm} (12)

---

Empirical Comparisons

SGD v.s. batch solvers\(^3\) on RCV1

<table>
<thead>
<tr>
<th>#Features</th>
<th>#Training examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>47,152</td>
<td>781,265</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time (secs)</th>
<th>Primal Obj</th>
<th>Test Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMO (SVM(^{tight}))</td>
<td>(\approx 16,000)</td>
<td>0.2275</td>
<td>6.02%</td>
</tr>
<tr>
<td>Cutting Plane (SVM(^{perf}))</td>
<td>(\approx 45)</td>
<td>0.2275</td>
<td>6.02%</td>
</tr>
<tr>
<td>SGD</td>
<td>(&lt; 1)</td>
<td>0.2275</td>
<td>6.02%</td>
</tr>
</tbody>
</table>

Where is the magic?  \(\leftarrow\) Lecture 12 and HW4.
Large-Scale Classification Challenges

- **Scalability challenge** – training massive classifiers (one per category)
  - Need fast algorithms
  - Need parallel computing strategies for divide-\&-conquer

- **Modeling challenges**
  - Skewed category distributions where many categories do not have enough training examples
  - Need to leverage the hierarchical or graphical dependency structures to "borrowing" labeled data from richer categories for poorer categories

Joint Optimization of All Classifiers
Leveraging Dependency Structures

- **Hierarchical Dependency**
  - Model parameters of parent and children should not be too far away from each other

\[
\mathbf{w}_n \sim \mathcal{N}(\mathbf{w}_{\pi(n)}, \Sigma)
\]

- **Graphical Dependencies**
  - Model parameters of each node should not be too far away from its nearest neighbors
Joint Optimization of All Classifiers: Parallel Divide-&-Conquer Strategies

- **Hierarchies**
  - Optimize odd and even levels alternately

- **Graphs**: First find a graph vertex coloring, and then
  - Pick a color
  - In parallel, optimize all nodes with that color
  - Repeat with a different color

---

Large-Scale Classification Challenges

- **Scalability challenge** – training massive classifiers (one per category)
  - Need fast algorithms
  - Need parallel computing strategies for divide-&-conquer

- **Modeling challenges**
  - Skewed category distributions where many categories do not have enough training examples
  - Need to leverage the hierarchical or graphical dependency structures to "borrowing" labeled data from richer categories for poorer categories

  **How?** → Lecture 15 on Extreme Scale Text Classification
The Power Law Phenomenon
(Skewed distribution of categories over documents)

\[ y = ax^{-b} \]
\[ \log y = \log a - b \log x \]
\[ b = 1: \text{Zipf's Law} \]

Large Taxonomies → Skewed Category Distribution
(using DMOZ 2010 as an example)

Majority of classes are rare

Distribution of training examples across classes (ODP)
Topic Coverage

- **High-dimensional Vector Spaces (1 lecture, HW1)**
  - Scalable Indexing, Matrix Calculus Review

- **Link Analysis (3 lectures, HW2)**
  - Analyze interconnected web sites, papers, people, social impacts, etc.

- **Collaborative Filtering (3 lectures, HW3)**
  - Model users’ tastes and predict their preference over items (e.g., shopping products)

- **Matrix Factorization, Stochastic Gradient Decent (3 lectures, HW4)**
  - Unsupervised learning of dimension reduced views of data

- **Web-scale Text Classification (5 lectures, HW5)**
  - Supervised learning of topic assignments over documents

- **Learning to Rank (2 lectures, no HW)**
  - Supervised learning for search engines based on relevance judgments

- **Statistical Significance Tests (3 lectures, HW6)**
  - Evaluation, methods comparison

- **Deep Learning for Text Mimicing and Beyond (4 lectures, HW7)**
  - Latent (semantic) representations of words, documents and topics in various tasks

Deep Learning for Text Mimicing

- **Lecture 1: An Introduction to Word Embedding**

- **Lecture 2. Recurrent Neural Networks (RNN)**

- **Lecture 3. Convolutional Neural Networks (CNN)**

- **Lecture 4. Knowledge-graph embedding and architecture search for Deep Learning**

- **Lecture 4. Contextualized Text Representations**
Administrative Details

Okay, that's the content … now for the administrative stuff

Course Materials Online

• Textbooks (available at the bookstore)
  • Primary: Introduction to Information Retrieval (IR), Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze, Cambridge University Press. 2008
  • Reference: Pattern Recognition and Machine Learning (ML), Christopher M. Bishop, Springer 2006

• Course syllabus and support systems via URL’s
  • http://nyc.lti.cs.cmu.edu/classes/11-741/s19/index.html for homework assignments, lecture notes, copies of papers (when necessary) and data with restricted access to .cmu.edu (or via VPN)
  • https://canvas.cmu.edu/courses/9338, Canvas
  • https://canvas.cmu.edu/courses/9338/external_tools/32_Piazza
**Fall 2017 Introduction**

- HW5: A programming assignment for text classification (rating prediction) on a large dataset of Yelp reviews. The implementation of a multi-class logistic regression (softmax) method is required, while existing software (LIBLINEAR) can be used for SVM.
- HW6: A problem-solving set for hands-on exercise with statistical significance tests, including sign test, t-test, proportion test, signed-rank test and rank-sum.
- HW7: A programming assignment for text classification on the same Yelp review dataset with deep learning, including word embedding, convolutional neural net (CNN) and recurrent neural net (RNN) components. Existing software like TensorFlow or Keras can be used.

---

**Course Syllabus (Slides)**

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Lecture</th>
<th>Reading</th>
<th>Homework</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15-Jan</td>
<td>Course overview and introduction</td>
<td></td>
<td>HW6 (link) (submission is not required; receiving feedback if submitted)</td>
</tr>
<tr>
<td>2</td>
<td>17-Jan</td>
<td>High-dimensional vectors and scalable indexing</td>
<td><em>IR: Ch 1.1, 1.2, 6.1, 6.2, 6.3</em></td>
<td>HW1 (link) (Due 1/23 11:59PM)</td>
</tr>
<tr>
<td>3</td>
<td>22-Jan</td>
<td>Link Analysis 1: HITS and PageRank</td>
<td>IR: Ch 21</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>24-Jan</td>
<td>Link Analysis 2: Personalized and Topic-sensitive PageRank</td>
<td><em>IR: Ch 8.1 – 8.4; Handout: WWW2002</em></td>
<td><em>HW2: write-up, resources (Due 2/6 11:59PM)</em></td>
</tr>
</tbody>
</table>

---

**Web Site of Lecture Slides**

**Index of /classes/11-741/s19/Lectures**

- **Name**: Parent Directory
  - **Last modified**: -
  - **Size**: -

*Apache/2.4.7 (Ubuntu) Server at nyc.lti.cs.cmu.edu Port 80*
Course Organization

- **Lectures**
  - With quizzes for every few lectures, focusing on concepts beyond HW
  - You may not get an A if missing a few quizzes (15% of total credits)

- **Hands-on experience (homework)**
  - 6 programming assignments (Python is required)
  - 1 problem solving exercises (significance tests)

- **Weekly readings**
  - Textbook chapters or relevant papers -- highly recommended

- **CPP (Capstone Project Proposal)**
  - Team work (oral presentations, peer-review)

- **No Exams**

Signature of the Course

- **A mixture of important building blocks in MLTM**
  - Instead of narrow foci on one or two areas

- **Global pictures and in-depth understanding**
  - Key concepts, intuitions, mathematical formulation, optimization algorithms

- **Learning by doing (7 HW assignments)**
  - “We felt the ownership after doing the HW assignments”

- **Capstone Project Proposal (CPP) (with 9 topics)**
  - Mental exercise (no implementation), focusing on knowledge transfer, literature review, creative thinking, method & experimental design, oral presentation
  - 3 or more students per group, 1 topic per group, graded by the entire group
Differences among 741, 641 and 441

(More details in the syllabus)

- **11-741**
  1. Count as a PhD-level course (12 units) in LTI
  2. CPP work/peer-review is required; HW 1-8

- **11-641**
  1. Count as a MS-level course (12 units) but not PhD-level in LTI
  2. CPP work/peer-review is required; HW subset (details in syllabus)

- **11-441**
  1. Count as UG course (9 units)
  2. CPP work is not required but peer-review is required; HW subset (details in syllabus)

<table>
<thead>
<tr>
<th></th>
<th>11-741 (PhD Level)</th>
<th>11-641 (MS Level)</th>
<th>11-441 (UG Level)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quiz (Mandatory)</td>
<td>15%</td>
<td>15%</td>
<td>15%</td>
</tr>
<tr>
<td>CPP (Mandatory)</td>
<td>15% Team Presentation + Peer Review</td>
<td>15% Peer Review Only</td>
<td>10% Peer Review Only</td>
</tr>
<tr>
<td>HW1</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>HW2</td>
<td>7%</td>
<td>0.0%</td>
<td>10%</td>
</tr>
<tr>
<td>HW3</td>
<td>10%</td>
<td>12%</td>
<td>14%</td>
</tr>
<tr>
<td>HW4</td>
<td>13%</td>
<td>15%</td>
<td>18%</td>
</tr>
<tr>
<td>HW5</td>
<td>7%</td>
<td>9%</td>
<td>10%</td>
</tr>
<tr>
<td>HW6</td>
<td>13%</td>
<td>15%</td>
<td>18%</td>
</tr>
<tr>
<td>HW7</td>
<td>15%</td>
<td>15%</td>
<td>18%</td>
</tr>
</tbody>
</table>

10 CPP Topics (w/ selected papers for each)

1) Semi-supervised Learning
2) Link Prediction in Citation Networks
3) Collaborative Filtering
4) Knowledgebase Completion
5) Graph Embedding
6) Question Answering and Reading Comprehension
7) Seq2Seq Models
8) Deep Learning for Text Classification
9) Deep Learning for Sentiment Detection

You submit your priority lists and we do the assignment.
Homework Policies

• All homework must be submitted via Blackboard
  o Due by 11:59 pm of the due date

• Late homework:
  o Deduct 10% for each day late.

  Don’t fall behind.

Cheating, Copying, Plagiarism, Etc

• You must be the author of everything that you submit for a grade

• Revising or modifying someone else’s work does not make you the author

• It is okay to discuss homework with other students, share ideas, experience, and lessons learned

• Turn in the signed form (otherwise you will not be graded)

Cheating, Copying, Plagiarism, Etc

Penalties
- Usually failure of the course
- Possibly expulsion from the graduate program

If you are having problems meeting your deadlines
... submit the assignment late, or don’t submit it at all
- Being late or taking a zero just lowers your grade
- Cheating causes you to fail the course