DSC 102
Systems for Scalable Analytics

Winter 2020

Arun Kumar
About Myself

2009: Bachelors in CSE from IIT Madras, India

Summers: 110F!

2009—16: MS and PhD in CS from UW-Madison
PhD thesis area: Data systems for ML workloads

Winters: −40F!

2016-: Asst. Prof. at UC San Diego CSE
2019-: + Asst. Prof. at UC San Diego HDSI

Ahh! 😊
My Current Research

New abstractions, algorithms, and software systems to “democratize” ML-based data analytics from a data management/systems standpoint

Democratization = System Efficiency + Human Efficiency
(Lower resource costs) (Higher productivity)

Practical and scalable data systems for ML analytics
Inspired by relational database systems principles
Exploit insights from learning theory and optimization theory
My Current Research

Research Approach:

Abstract key steps + Formalize computation + Automate grunt work + Optimize execution

https://adalabucsd.github.io/
What is this course about? Why take it?
1. IBM’s Watson wins Jeopardy!
How did Watson achieve that?
Watson devoured LOTS of data!
2. “Structured” data with search results

Pradeep Khosla - UC San Diego Office of the Chancellor - University ...  
Pradeep K. Khosla became UC San Diego's eighth Chancellor on August 1, 2012. As UC San Diego's chief executive officer, he leads a campus with more than ...  

Pradeep K. Khosla - UC San Diego Office of the Chancellor  
Chancellor, University of California San Diego. Pradeep K. Khosla, an internationally renowned electrical and computer engineer, is the eighth Chancellor of the ...  

Pradeep Khosla - Wikipedia  
Pradeep K. Khosla is an academic computer scientist and university administrator. He is the current chancellor of the University of California, San Diego. He was ...  

Pradeep Khosla | LinkedIn  
Greater San Diego Area - Chancellor, UC San Diego - Avigilon  
View Pradeep Khosla's professional profile on LinkedIn. LinkedIn is the world's largest business network, helping professionals like Pradeep Khosla discover ...  

Robotics Institute: Pradeep Khosla  
www.ri.cmu.edu › people
How does Google know that?
Google also devours LOTS of data!

Knowledge Vault* fuses all these signals together

- Data from web
  - Unstructured text
  - Semi-structured DOM trees
  - Structured WebTables
- “Prior” data from FB

Details in a paper submitted to WWW’14 (Dong et al)
3. Amazon’s “spot-on” recommendations

![Amazon screenshot showing recommended products and star ratings.](image-url)
How does Amazon know that?
You guessed it! LOTS and LOTS of data!
And innumerable “traditional” applications
Scalable software systems for data management and analytics are the cornerstone of many digital applications, both modern and traditional.
When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren’t seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, “It was like arriving at a conference reception and realizing you don’t know anyone. So you just stand in the corner sipping your drink—
Data data everywhere,  
All the wallets did shrink!  
Data data everywhere,  
Nor any moment to think?
DSC 102 will get you thinking about the fundamentals of scalable analytics systems

1. “Systems”: What resources does a computer have? How to store and compute efficiently over large data? What is cloud computing?
2. “Scalability”: How to scale and parallelize data-intensive computations?
3. Scalable Systems for “Analytics”:
   3.1. Source: Data acquisition & preparation for ML
   3.2. Build: Dataflow & Deep Learning systems
   3.3. Deploying ML models
4. Hands-on experience with tools for scalable analytics
The Lifecycle of ML-based Analytics

Data acquisition
Data preparation

Feature Engineering
Training & Inference
Model Selection

Model Serving
Monitoring

Data Scientist/ML Engineer

Source
Build
Deploy

ML/AI + Data Systems Infrastructure

python
TensorFlow
pyTorch
DASK
Spark
AWS
Learning Outcomes of this course

- Understand the basic systems principles of the memory hierarchy, scalable data access, parallelism paradigms, cloud computing, and containerization.
- Identify the abstract data access patterns of, and opportunities for parallelism in, data processing and ML algorithms.
- Reason critically about practical tradeoffs between accuracy, efficiency, scalability, usability, and total cost.
- Learn the basics of dataflow (“Big Data”) programming with HDFS, MapReduce, and Spark.
- Gain exposure to deep learning inference on unstructured data with TensorFlow and Keras.
- Apply SQL, dataflow programming, and DL inference for end-to-end pipelines for data preparation, feature engineering, and model selection on large-scale heterogeneous datasets.
What this course is NOT about

❖ NOT a course on databases, relational model, or SQL
  ❖ Take DSC 100 instead (pre-requisite!)
❖ NOT a course on how to use DBMSs or SQL for DB-backed applications (indexing, JDBC, triggers, etc.)
  ❖ Take CSE 132B instead
❖ NOT a training module for how to use Spark
❖ NOT a course on internal details of RDBMSs/Spark
  ❖ Take CSE 132C instead
❖ NOT a course on ML or data mining algorithmics; instead, we focus on ML systems
Q: What is a Machine Learning (ML) System?

❖ A data processing system (aka data system) for mathematically advanced data analysis operations (inferential or predictive), i.e., beyond just SQL aggregates
  ❖ Statistical analysis; ML, deep learning (DL); data mining (domain-specific applied ML + feature eng.)
  ❖ *High-level APIs* for expressing statistical/ML/DL computations over large datasets
Background: ML 101

Generalized Linear Models (GLMs); from statistics

Bayesian Networks; inspired by causal reasoning

Decision Tree-based: CART, Random Forest, Gradient-Boosted Trees (GBT), etc.; inspired by symbolic logic

Support Vector Machines (SVMs); inspired by psychology

Artificial Neural Networks (ANNs): Multi-Layer Perceptrons (MLPs), Convolutional NNs (CNNs), Recurrent NNs (RNNs), Transformers, etc.; inspired by brain neuroscience
Data Systems Concerns in ML

Key concerns in ML:
- Accuracy
- Runtime efficiency (sometimes)

Additional key *practical* concerns in ML Systems:
- Scalability (and efficiency at scale)
- Usability
- Manageability
- Developability

Q: How do “ML Systems” relate to ML?

ML Systems : ML :: Computer Systems : TCS

Q: What if the dataset is larger than single-node RAM?

Q: How are the features and models configured?

Q: How does it fit within production systems and workflows?

Q: How to simplify the implementation of such systems?

Long-standing concerns in the DB systems world!

Can often trade off accuracy a bit to gain on the rest!
## Conceptual System Stack Analogy

<table>
<thead>
<tr>
<th>Theory</th>
<th>Relational DB Systems</th>
<th>ML Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>First-Order Logic</td>
<td>Learning Theory</td>
</tr>
<tr>
<td></td>
<td>Complexity Theory</td>
<td>Optimization Theory</td>
</tr>
<tr>
<td>Program Formalism</td>
<td>Relational Algebra</td>
<td>Matrix Algebra</td>
</tr>
<tr>
<td>Program Specification</td>
<td>Declarative Query Language</td>
<td>Gradient Descent</td>
</tr>
<tr>
<td>Program Modification</td>
<td>Query Optimization</td>
<td>TensorFlow? R? Scikit-learn?</td>
</tr>
<tr>
<td>Execution Primitives</td>
<td>Parallel Relational Operator Dataflows</td>
<td>Depends on ML Algorithm</td>
</tr>
<tr>
<td>Hardware</td>
<td>CPU, GPU, FPGA, NVM, RDMA, etc.</td>
<td></td>
</tr>
</tbody>
</table>
Categorizing ML Systems

❖ Orthogonal Dimensions of Categorization:

1. **Scalability**: In-memory libraries vs Scalable ML system (works on larger-than-memory datasets)

2. **Target Workloads**: General ML library vs Decision tree-oriented vs Deep learning, etc.

3. **Implementation Reuse**: Layered on top of scalable data system vs Custom from-scratch framework
Major Existing ML Systems

General ML libraries:

- In-memory:
  - Torch
  - R
  - WEKA
  - SAS
  - DASK

- Disk-based files:
  - MADlib

- Layered on RDBMS/Spark:
  - Spark
  - MLlib

Cloud-native:

- Azure Machine Learning
- Amazon SageMaker

“AutoML” platforms:

- DataRobot
- H2O.ai

Decision tree-oriented:

- XGBoost
- LightGBM

Deep learning-oriented:

- TensorFlow
- PyTorch
Q: Suppose you are given ad click-through prediction models A, B, C, and D with accuracies of 95%, 85%, 90%, and 85%, respectively. Which one will you pick?

Q: What about now?

- Real-world data scientists must grapple with multi-dimensional Pareto surfaces: accuracy, monetary cost, training time, scalability, inference latency, tool availability, interpretability, fairness, etc.

- Multi-objective optimization criteria set by application needs / business policies.
And now for the (boring) logistics …
Prerequisites

❖ **DSC 100** (or equivalent) is necessary
❖ Transitivity **DSC 80**; basics of ML is necessary
❖ Proficiency in Python programming
❖ For all other cases, email the instructor with proper justification; a waiver can be considered

http://cseweb.ucsd.edu/~arunkk/dsc102_winter20/
Lectures: TueThu 12:30-1:50pm, PCYNH 106

*Attending ALL lectures is mandatory!*

Instructor: Arun Kumar; arunkk@eng.ucsd.edu
Office hours: Thu 2-3pm, 3218 CSE (EBU3b)
TAs: Supun Nakandala, Vraj Shah, and Yuhao Zhang
Discussions: Fri 8:00-8:50am, CENTR 115
Piazza: https://piazza.com/class/k4x69eft94v65n

*Bring your iClicker to every lecture!*

http://cseweb.ucsd.edu/~arunkk/dsc102_winter20/
Grading

❖ Midterm Exam: 20%
  Date: Thu, Feb 6; in-class (12:30-1:50pm)

❖ 3 Programming Assignments: 35% (10% + 15% + 10%)
  ❖ No late days! Plan your work well ahead.

❖ 5 Surprise Quizzes (in-class iClicker): 5%

❖ Final Exam: 40% (cumulative)
  Date: Tue, Mar 17; 11:30am-2:30pm; Room TBD

http://cseweb.ucsd.edu/~arunkk/dsc102_winter20/
Hybrid of relative and absolute; grade is **better** of the two

<table>
<thead>
<tr>
<th>Grade</th>
<th>Relative Bin (Use strictest)</th>
<th>Absolute Cutoff (&gt;=)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>Highest 5%</td>
<td>95</td>
</tr>
<tr>
<td>A</td>
<td>Next 10% (5-15)</td>
<td>90</td>
</tr>
<tr>
<td>A-</td>
<td>Next 15% (15-30)</td>
<td>85</td>
</tr>
<tr>
<td>B+</td>
<td>Next 15% (30-45)</td>
<td>80</td>
</tr>
<tr>
<td>B</td>
<td>Next 15% (45-60)</td>
<td>75</td>
</tr>
<tr>
<td>B-</td>
<td>Next 15% (60-75)</td>
<td>70</td>
</tr>
<tr>
<td>C+</td>
<td>Next 5% (75-80)</td>
<td>65</td>
</tr>
<tr>
<td>C</td>
<td>Next 5% (80-85)</td>
<td>60</td>
</tr>
<tr>
<td>C-</td>
<td>Next 5% (85-90)</td>
<td>55</td>
</tr>
<tr>
<td>D</td>
<td>Next 5% (90-95)</td>
<td>50</td>
</tr>
<tr>
<td>F</td>
<td>Lowest 5%</td>
<td>&lt; 50</td>
</tr>
</tbody>
</table>

**Example:** Score 82 but 33%ile; Rel.: B-; Abs.: B+; so, B+
## Tentative Course Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>Basics of Computer Org. and Operating Systems</td>
</tr>
<tr>
<td>1-2</td>
<td>Basics of Cloud Computing</td>
</tr>
<tr>
<td>2-3</td>
<td>Scalable Data Access; Parallelism Paradigms</td>
</tr>
<tr>
<td>4</td>
<td>Data Preparation for ML</td>
</tr>
<tr>
<td>4</td>
<td>Guest Lecture by Alkis Polyzotis (Google Brain) on Thu, Jan 30</td>
</tr>
<tr>
<td>5</td>
<td>Review; <strong>Midterm Exam 1 on Thu, Feb 6</strong></td>
</tr>
<tr>
<td>6-7</td>
<td>Dataflow Systems for Data Preparation and ML</td>
</tr>
<tr>
<td>8</td>
<td>Deep Learning Systems</td>
</tr>
<tr>
<td>9</td>
<td>ML Deployment</td>
</tr>
<tr>
<td>9</td>
<td>Guest Lecture by Manasi Vartak (Verta.AI) on Thu, Mar 5</td>
</tr>
<tr>
<td>10</td>
<td>Optional: Open Research Questions; Review</td>
</tr>
<tr>
<td>11</td>
<td><strong>Final Exam on Tue, Mar 17</strong></td>
</tr>
</tbody>
</table>

---

**Attending ALL lectures (incl. guest lectures) is mandatory!**
<table>
<thead>
<tr>
<th>Date</th>
<th>Agenda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fri, Jan 17</td>
<td>PA 1 released</td>
</tr>
<tr>
<td>Fri, Jan 17</td>
<td>Discussion on PA 1 by Vraj Shah</td>
</tr>
<tr>
<td>Mon, Feb 3</td>
<td>PA 1 due</td>
</tr>
<tr>
<td>Thu, Feb 6</td>
<td>(Midterm Exam)</td>
</tr>
<tr>
<td>Fri, Feb 7</td>
<td>PA 2 released</td>
</tr>
<tr>
<td>Fri, Feb 7</td>
<td>Discussion on PA 2 by Yuhao Zhang</td>
</tr>
<tr>
<td>Mon, Feb 24</td>
<td>PA 2 due</td>
</tr>
<tr>
<td>Tue, Feb 25</td>
<td>PA 3 released</td>
</tr>
<tr>
<td>Fri, Feb 28</td>
<td>Discussion on PA 3 by Supun Nakandala</td>
</tr>
<tr>
<td>Wed, Mar 11</td>
<td>PA 3 due</td>
</tr>
<tr>
<td>Tue, Mar 17</td>
<td>(Final Exam)</td>
</tr>
</tbody>
</table>

**No late days—plan your work upfront!**
**Do not miss the Discussion slot talks by the TAs.**
Suggested Textbooks

Aka “CompOrg Book”  Aka “Comet Book”  Aka “Cow Book”

Aka “Spark Book”  Aka “MLSys Book”

(Free PDFs available online; also check out our library)
Why so many textbooks?!

1. Computer systems are about carefully layering *levels of abstraction*!

   - Hardware
   - Low-level systems software
   - Operating Systems
   - Higher-level relational dataflows
   - ML-oriented dataflows & lifecycle
   - Data Management in Machine Learning Systems
   - More general scalable dataflows
   - Spark: The Definitive Guide

2. Analytics/ML Systems is a recent/emerging area of research

3. Also, DSC 102 is the first UG course of its kind in the world! 🙂
General Dos and Do NOTs

Do:
❖ Raise your hand before asking questions during lectures
❖ Participate in class discussions; also on Piazza, if you like
❖ Use “DSC102:” as subject prefix for all emails to me/TAs

Do NOT:
❖ Plagiarize or share PA code/solutions with your peers
❖ Cheat on your exams or PAs; I will notify the university!
❖ Harass, cut off, or be disrespectful to your peers or TAs
❖ Use email as primary communication mechanism for doubts/questions instead of Office Hours
❖ Record or quote the instructor’s anecdotes out of class! 😊