Machine Learning for Data Management Systems

Introduction

Amol Deshpande
Jan 26, 2023

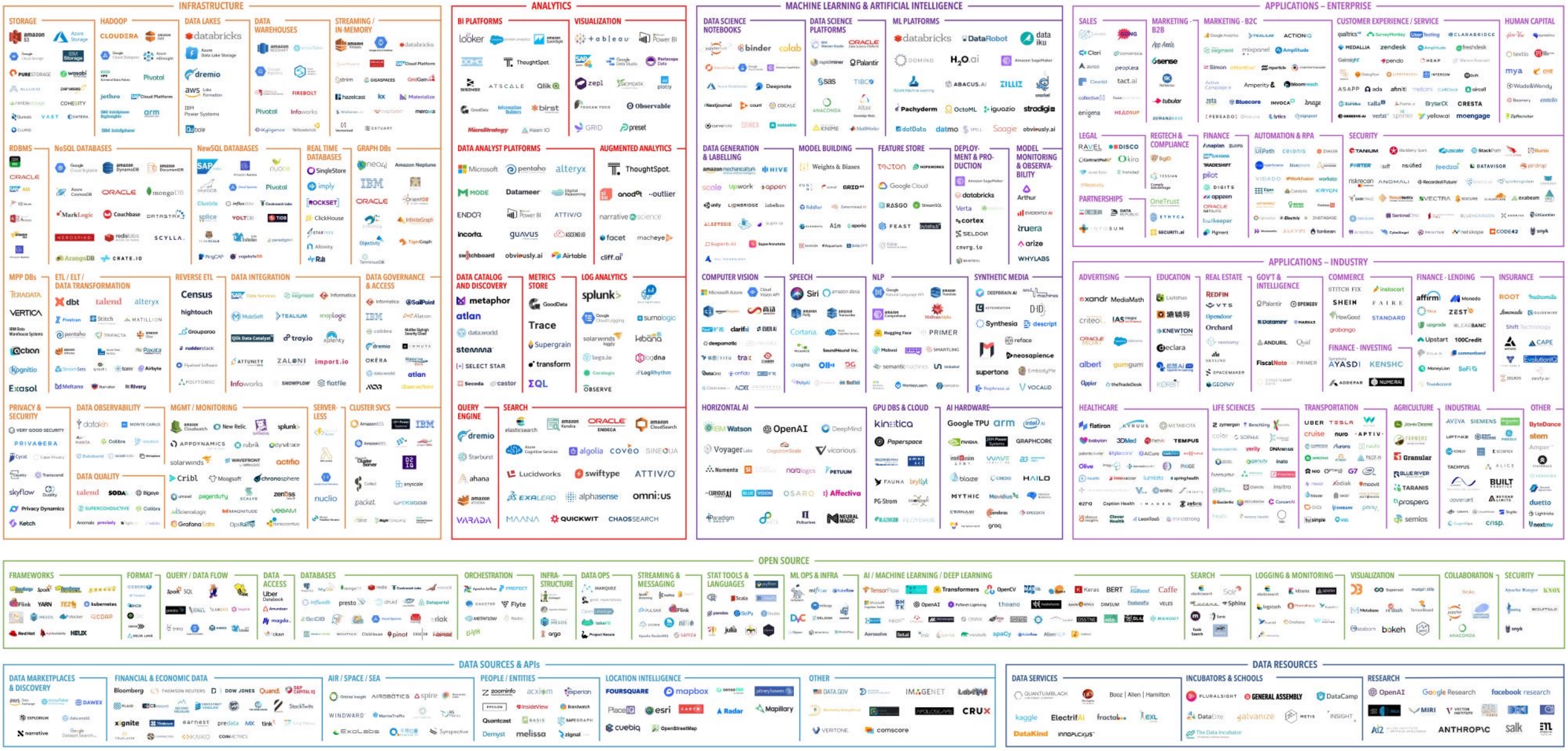
Outline

- Motivation
- Course Goals and Focus Areas
- Intros
- Overview of Data Management Systems
- History of Database Automation

Promise of Big Data

- Explosion of data, in pretty much every domain
  - Sensing devices and sensor networks that can monitor everything 24/7 from temperature to pollution to vital signs
  - Increasingly sophisticated smart phones
  - Internet, social networks make it easy to publish data
  - Scientific experiments and simulations → astronomical data volumes
  - Genome/health data
  - Internet of Things, Smart wearables
  - …

MACHINE LEARNING, ARTIFICIAL INTELLIGENCE, AND DATA (MAD) LANDSCAPE 2021



mattturck.com/data2021

FIRSTMARK
EARLY STAGE VENTURE CAPITAL

Backbone = Data Management Systems

- Need better and bigger data management systems to support these use cases
  - To handle the much much larger datasets (Volume)
  - To respond quickly to new data (Velocity)
  - To manage and query a wide variety of complex data types (Variety)
  - To properly reason about robustness and other issues (Veracity)

The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety and Veracity**.

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015
4.4 MILLION IT JOBS
will be created globally to support big data, with 1.9 million in the United States

Volume
SCALE OF DATA

40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by 2020, an increase of 300 times from 2005

It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day

6 BILLION PEOPLE
have cell phones

Most companies in the U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored

Variety
DIFFERENT FORMS OF DATA

As of 2011, the global size of data in healthcare was estimated to be
150 EXABYTES
[161 BILLION GIGABYTES]

30 BILLION PIECES OF CONTENT
are shared on Facebook every month

By 2014, it's anticipated there will be
420 MILLION WEARABLE, WIRELESS HEALTH MONITORS

4 BILLION+ HOURS OF VIDEO
are watched on YouTube each month

400 MILLION TWEETS
are sent per day by about 200 million monthly active users

Velocity
ANALYSIS OF STREAMING DATA

The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session

Modern cars have close to
100 SENSORS
that monitor items such as fuel level and tire pressure

By 2016, it is projected there will be
18.9 BILLION NETWORK CONNECTIONS
– almost 2.5 connections per person on earth

Veracity
UNCERTAINTY OF DATA

1 IN 3 BUSINESS LEADERS
don't trust the information they use to make decisions

in one survey were unsure of how much of their data was inaccurate

Poor data quality costs the US economy around
$3.1 TRILLION A YEAR

Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

IBM

Modern Data Management Systems

- Much more complex than in the past
  - Deployed over 100's or 1000's of servers (or more)
    - With rapidly changing configurations
    - Often deployed through virtualization on clouds
  - Many combinations of hardware technologies
    - CPU/GPUS, Complex Cache Hierarchies, Direct Remote Memory Access, Fast interconnects, …
    - Likely a mix of hardware with different characteristics in a single setup
    - Frequent changes, upgrades, etc.

Modern Data Management Systems

- Much more complex than in the past
  - Support many different types of data
    - JSON, Video, Timeseries, Audio, Text, Geospatial, …
  - More complex query languages with many features
    - More operations, user-defined functions…
    - New query languages (e.g., Apache Spark, MongoDB QL)
  - Recently built-in support for ML training and inference
  - Corresponding increase in the software complexity
    - New types of join operators and indexes

Modern Data Management Systems

- Many more DMS today than in the past
  - Relational (SQL-based) database systems
  - Stream processing systems (focusing on streaming data)
  - Special-purpose data warehousing systems (most start from some RDBMS)
  - Batch analysis frameworks (like Hadoop, Pregel, Spark, …)
    - Typically, data stored in distributed file systems
  - Key-value stores (like HBase, Cassandra, Redis, …)
    - Basically, persistent distributed hash tables
  - Semi-structured/Document data stores (for XML/JSON query processing)
  - Graph databases
  - Data lakes (e.g., scientific data, machine learning data)

1.Tuning Data Management Systems

- Data management systems have many “knobs” (tuning params)
  - max #connections, shared memory, cache size, when to garbage collect, how often to run statistics, how to allocate memory across components, commit parameters, …
    - PostgreSQL has about 170 knobs -- a small fraction with significant impact
  - Which materialized views to maintain, what indexes to use, what compression schemes (in data warehouses), window sizes (in streaming systems), what keys to use for partitioning data, how to partition, how many machines to use for query processing, …
    - Most have significant impact on performance

1.Tuning Data Management Systems

- In the past, most decisions made by "DBAs"
- Much harder to "tune" or "configure" modern systems
  - Too many variables and too many combinations → Hard for humans to reason about
    - Lot of trial and error required, not feasible at the data volumes
  - Too many different systems → Hard to build up the experience
  - Environment variables changing too rapidly

- Motivation 1: Build autonomous data management systems using ML

2. "Learned" Components

- Many complex trade-offs when making design decisions in a database system
  - Different indexes better for different environments/different workloads
  - Same for storage layouts and other design decisions

- Motivation 2: Could we use modern ML techniques to design new self-adapting components, that can learn from the data/workload and automatically do the right thing for the given data/workload?

3.Workload Forecasting

- Better understanding of the future workload can help with planning through…
  - Allocating additional resources proactively rather than reactively
  - Exploiting different tradeoffs (e.g., using less memory per task if many tasks expected)
- Caveat: There must be patterns to be learned from
- Motivation 3: Incorporate forecasting algorithms to improve overall performance

4. Intra-query Adaptivity

- For complex queries/analysis tasks, things can change significantly during execution of a single query/task
  - Data characteristics may be very different than expected
  - Resources may fluctuate significantly during execution

- Motivation 4: Use ML techniques to adapt during the execution of a single query/task

5.Hard Planning/Optimization Problems

- Quite a few NP-Hard planning/optimization problems being solved in systems
  - (Query optimization) choosing a "query plan" given a complex query
  - Partitioning strategies in distributed systems, etc.
- Often need to be solved in presence of significant "uncertainties"
- Motivation 5: Could use of ML techniques provide different solutions to such problems? If yes, why?
  - Recent work on how deep learning could be used to "partially" solve hard combinatorial problems

6.User Interfaces/Interactions

- Natural language interfaces to querying (e.g., through conversion to SQL)
- Inferring user intent and responding accordingly with the right data/graphs
- Reducing the time to design schemas and build end-to-end applications

- Motivation 6: Using LLMs (large language models) and other such technologies to improve these facets

7.Miscellaneous

- Capturing correlations in the data for better estimation of query sizes (for optimization or approximate query processing)
- Synthetic data generation (e.g., to preserve privacy) through use of generative models
- Better dataset discovery and correlation in data lakes
- …

Summary

- Modern data management systems are too complex to manage
- Many ways to incorporate ML techniques
  - to improve performance through forecasting and adapting
  - to reduce friction in user interactions
  - to obtain better optimization algorithms in face of uncertainty
  - …
- Many other places where ML comes up in data management
- Also, much work on using database techniques to improve ML

Outline

- Motivation
- **Course Goals and Focus Areas**
- Intros
- Overview of Data Management Systems
- History of Database Automation

Course Goals

- Overarching goal: How to rearchitect modern data management systems to utilize advances in ML especially deep learning
  - Evolutionary (e.g., better forecasting), or
  - Revolutionary (entirely change how indexing or QO is done)
- More specifically:
  - Study the recent work on applying ML to data management systems
  - Reason about whether the use of ML is appropriate and why prior techniques can't be adapted
  - Think through the failure scenarios
  - Understand fundamental reasons (if any) why ML-based approach is superior
  - Explore other places where ML could help (especially LLMs)
  - Simplify data management systems through use of ML

Topics

- Learned Indexes and Storage Layouts
  - Improve performance of search and storage organization through learning
- Query Processing
  - Adaptive query operators, as well as adaptive query processing
- Query Optimization
  - Better estimations through capturing correlations, better search algorithms
- Natural Language to SQL
- Workload forecasting and resource management
- *May adjust as the semester goes on*

Approach

- Read 1-2 papers per class, mostly from database/systems conferences
  - I will try to provide the relevant background on the DB side
  - Coverage of ML techniques as required for the papers
    - May take breaks in between to cover some of the ML background in more depth
- Discuss each paper in the class
  - with approx. 45-minute presentation by one of you (will circulate sign up sheet)
  - Primary aim to discuss the papers deeply
  - Secondary goal to cover the broader topic of the paper but hard to do given how new the work is
- A few classes dedicated to broader discussions

Grading

- Paper Readings + class participation, etc. (20%)
  - Submissions through Gradescope (not Slack)
  - Due 11am of the day of the class
- Written Assignments/Final -- all individual (50%)
  - Spread throughout the semester -- will cover additional papers
  - Survey assignment: One written assignment will be doing a literature survey on one of the relevant topics and summarizing the recent work in that topic
  - One assignment on proposing a new idea in this space
- Research Project (30%)
  - Group research project

Other Logistics

- All submissions (paper critiques, assignments, project deliverables) through Gradescope
  - Will share all submitted critiques after the deadline
- Slack to be used for announcements/discussions
- Will try to move (at least some of) the classes to Iribe

Outline

- Motivation
- Course Goals and Focus Areas
- **Intros**
- Overview of Data Management Systems
- History of Database Automation

Next Steps…

- Sign up for Slack and Gradescope (not set up yet)
- Look out for the sign-up sheet for class presentations (starting the week after next)
- Readings for next week (more background)

- We will start with "Architecture of a Traditional Database System" in the next class