Thoughts on the Past, Present and Future

Sreenivas Rangan Sukumar, PhD

Data Scientist and R&D Staff
Computaional Data Analytics Group
Computational Sciences and Engineering Division
Oak Ridge National Laboratory

Email: sukumarsr@ornl.gov Phone: 865-241-6641

Tomorrow: Experience with Data Parallel Frameworks

Food-for-thought towards the exascale data analysis supercomputer
Today’s Outline

• Scalable Machine Learning
  – Recent Advances and Trends

• State of the Practice
  – Philosophy, Engineering, Process, Paradigms

• Are we there yet?
  – If yes, how so?
  – If not, why not?

• Concluding Future Thoughts

• Offline Debate and Discussion
Given examples of a function \((x, f(x))\), **Predict** function \(f(x)\) for new examples \(x\)

<table>
<thead>
<tr>
<th>(f_1)</th>
<th>(f_2)</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>(f_d)</th>
<th><strong>Labels</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(c_1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(c_1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(c_3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(c_2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(c_k)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(c_k)</td>
</tr>
</tbody>
</table>
Machine Learning in the Big Data Era
Just in case you missed….

<table>
<thead>
<tr>
<th></th>
<th>1990 – 2000s</th>
<th>2010-Present</th>
<th>Insight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assumption</td>
<td>A model exists. Better data will reveal the beautiful model. (Knowing “why” is important)</td>
<td>A model may not exist, but find a model anyway. (“Why” is not as important)</td>
<td>Dilemma: Better data or better algorithms.</td>
</tr>
<tr>
<td>Complexity of data</td>
<td>(N \sim O(10^2)), (d \sim O(10^1)) (e.g. IRIS data) (k \sim O(1))</td>
<td>(N \sim O(10^6)) (d \sim O(10^4)) (e.g. ImageNet) (k \sim O(10^4))</td>
<td>Volume, Velocity, Variety and Veracity have all increased several orders of magnitude.</td>
</tr>
<tr>
<td>Data – Model Relationship</td>
<td>Model abstracts data</td>
<td>Data is the model</td>
<td>Models aggregated data. It is not anymore about the average. It is about every individual data point.</td>
</tr>
<tr>
<td></td>
<td>(\hat{p}(x_i \mid c = c_j) = \frac{1}{\sqrt{2\pi} \sigma_j} \exp \left( \frac{- (x_i - \mu_j)^2}{2\sigma_j^2} \right) )</td>
<td>(f(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{h_i} G \left( \frac{x - x_i}{h_i} \right) )</td>
<td></td>
</tr>
<tr>
<td>Model Parameter Complexity (e.g. Size of Neural Network)</td>
<td>(O(10^3))</td>
<td>(O(10^{10})) (O(10^{8})) to (O(10^{10})) in months.</td>
<td>10-billion parameter network learned to recognize cats from videos.</td>
</tr>
<tr>
<td>Accuracy, Precision, Recall e.g. Face Recognition Visual Scene</td>
<td>(\sim 70%) was accepted Not possible</td>
<td>(\sim 95%) is the norm (\sim 10%) is the best result to date.</td>
<td>Big Data also means Big Expectations.</td>
</tr>
<tr>
<td>Computing Capability Personal Computing High Performance Computing</td>
<td>1 core, 256MB RAM, 8GB disk 1000 cores, 1 teraflops</td>
<td>16 cores, 64 GB RAM, 2TB disk 3 million cores, 34 petaflops</td>
<td>Commercial tools are keeping pace with the PC market and not HPC market.</td>
</tr>
<tr>
<td>Number of Dwarves !</td>
<td>7</td>
<td>13</td>
<td>Big Data Magic: Dwarves are doubling.</td>
</tr>
</tbody>
</table>
Today’s Talk

‘Compute’ is scaling up commensurate the ‘data’. Is machine learning keeping pace with the data and compute scale-up?

- If Yes: How so?
- If Not: Why not?
Scalable Machine Learning: Philosophy

The Lifecycle of Data-Intensive Discovery

- Querying and Retrieval e.g. Google, Databases
- Interrogation
- Association e.g. Robotic Vision
- Modeling, Simulation, & Validation e.g. Climate Change Prediction

Off-the-shelf Parallel Hardware

- Custom ICs e.g. FPGAs, Adapteva, Raspberry Pi
- Customized Processing E.g. Nvidia GPGPUs, YarcData Urika
- Multi-core HPC e.g. (Cray XK, Cray XC, IBM Blue Gene)
- Virtual clusters / Cloud computing e.g. Amazon AWS, SAS (PaaS, + SaaS)
Scalable Machine Learning: Philosophy

The Lifecycle of Data-Intensive Discovery

- Querying and Retrieval
  - e.g. Google, Databases

- Interrogation

- Association
  - Data-fusion
  - e.g. Robotic Vision

- Predictive Modeling
  - e.g. Climate Change Prediction

- Better Data Collection

- Predictive modeling appliance

- Simulation

- Relationship analytics

- Business Intelligence
Scalable Machine Learning: Discovery Process

The Lifecycle of Data-Intensive Discovery

Descriptive Analytics
What happened?

Diagnostic Analytics
Hindsight
Why did it happen?

Predictive Analytics
Insight
What will happen?

Prescriptive Analytics
Foresight
How can we make it happen?

Data-Driven Discovery Process

Querying and Retrieval
e.g. Google, Databases

Interrogation

Descriptive Analytics

Diagnostic Analytics

Predictive Analytics

Prescriptive Analytics

Concept adapted from Gartner’s Webinar on Big Data

Data-fused

Association
e.g. Robotic Vision

Modeling, Simulation, & Validation
e.g. Climate Change Prediction

The Lifecycle of Data-Intensive Discovery
Better Data Collection

Concept adapted from Gartner’s Webinar on Big Data
Scalable Machine Learning: System Engineering

Data-Driven Discovery Process

- **Descriptive Analytics**: What happened?
  - Hindsight

- **Diagnostic Analytics**: Why did it happen?
  - Insight

- **Predictive Analytics**: What will happen?
  - Foresight

- **Prescriptive Analytics**: How can we make it happen?
  - How can we make it happen?

- **Staging for Predictive Modeling**
  - Extract, Transform, Load
  - Data Pre-processing
  - Feature Engineering

- **Predictive Modeling**
  - Rule-base extraction
  - Pairwise-similarity (Distance Computation)
  - Model-parameter estimation
  - Cross validation

- **Inference/ Model Deployment**
  - Data is model? Model is data?
  - Adaptive model? Reinforcement?

Concept adapted from Gartner’s Webinar on Big Data
Scalable Machine Learning: Production

• Staging for Predictive Modeling
  – Extract, Transform, Load
  – Data Pre-processing
  – Feature Engineering

• Predictive Modeling
  – Rule-base extraction
  – Pairwise-similarity (Distance Computation)
  – Model-parameter estimation

• Inference/ Model Deployment
  – Cross-validation
  – Data is model ? Model is data ?
  – Adaptive model ? Reinforcement ?

Disk Intensive
File processing and repeated retrieval best done in massively parallel file systems or databases

Disk, Memory and Compute Intensive
Typically computing an aggregate measure, vector product, a kernel function etc.

Memory + Compute Intensive
Real-time requirements
Scalable Machine Learning: Bleeding Edge

The Berkeley Data Analysis Stack

In-database processing

In-memory processing

Storage scale-up

This is tremendous progress....
But...

Is machine learning keeping pace with the data and compute scale-up?

• If Yes: How so?
• If Not: Why not?
The 5 Challenges of Scalable Machine Learning

Given examples of a function \((x, f(x))\), **Predict** function \(f(x)\) for new examples \(x\)

**Dimensionality (Variety)**

<table>
<thead>
<tr>
<th>(f_1)</th>
<th>(f_2)</th>
<th>(\ldots)</th>
<th>(\ldots)</th>
<th>(\ldots)</th>
<th>(f_d)</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c_1)</td>
<td>(c_1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c_1)</td>
<td>(c_1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c_3)</td>
<td>(\ldots)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\ldots)</td>
<td>(\ldots)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c_2)</td>
<td>(\ldots)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c_k)</td>
<td>(\ldots)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(c_k)</td>
<td>(\ldots)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Science of Data**
(Quality, Structure, Content, Signal-to-noise)

**Data Science**
(Infrastructure, Hardware, Software, Algorithms)

**The Big Data Problem**
(Volume, Velocity, Veracity)

**Multi-class scaling**
(Number, Hierarchies)
### Challenge #1: Data Science

<table>
<thead>
<tr>
<th>Systems</th>
<th>Data</th>
<th>Compute</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure</td>
<td>Management</td>
<td>Structure</td>
<td>HPC</td>
</tr>
<tr>
<td>Design</td>
<td>• Quality</td>
<td>• Matrix/ Table</td>
<td>• TITAN</td>
</tr>
<tr>
<td>Operations</td>
<td>• Privacy</td>
<td>• Text, Image, Video</td>
<td>• CADES</td>
</tr>
<tr>
<td>Management</td>
<td>• Provenance</td>
<td>• Graphs</td>
<td>• Cloud</td>
</tr>
<tr>
<td>• Governance</td>
<td>• Sequences</td>
<td>• Urika</td>
<td>• SQL</td>
</tr>
<tr>
<td>Architecture</td>
<td></td>
<td>• Spatiotemporal</td>
<td>• Hadoop</td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td>• Schema</td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Databases</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NoSQL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graph</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Performance of “algorithm” dependent on architecture.
  - Most data scientists/algorithm specialists are used to in-memory tools such as R, MATLAB etc.
  - Existing cloud-based solutions are designed for high performance storage and not high-performance compute or in-memory operations.
  - Steep learning curve towards programming “new” innovative algorithms. Too many options without guiding benchmarks.
Challenge #2: Science of Data

• Data-science is not the same as “science of data”
  – Is the process of understanding characteristics of data before applying/designing a machine-learning algorithm.

  - SNR > 1
  - SNR >>> 1
  - SNR << 1
  - SNR >> 1

• Data characterization – (Avoid using machine learning as a black box)
  • Signal-noise-ratio, bound on noise
  • i.i.d sampling assumptions
  • stationarity, randomness, ergodicity, periodicity
  • Generating models behind data
Challenge #3: The N-d-k problem

• The Big Data Problem
  – The future is unstructured.
    • Text, images, videos, sequences

• Algorithms and infrastructure expected to handle Big Data – i.e., increasing N, d and k.
  – Feature engineering and requires automation.
    • Self-feature extracting methodologies encouraged.
  – Traditional (painstaking) pipeline of SMEs creating features from the data will fail or transform into a collaborative-parallel effort.
  – Increasing $N$ does not imply increasing information content. (Samples can still be good if not better than all of the data statistically.)
  – There can be hierarchies within the N-d-k dimensions.
Challenge #4: The N-d-k problem (d)

• Traditional algorithms assume \( N \gg d \) and \( d > k \)
  – Most tools available today scale well for increasing \( N \).

  - [Donoho, 2000] – The curse and blessings of dimensionality
  - Methods are emerging: Multi-task learning, Spectral Hashing etc.

<table>
<thead>
<tr>
<th></th>
<th>single</th>
<th>multi</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWLR</td>
<td>( O(mn^2 + n^3) )</td>
<td>( O(mn^2 + n^3) )</td>
</tr>
<tr>
<td>LR</td>
<td>( O(mn^2 + n^3) )</td>
<td>( O(mn^2 + n^3) )</td>
</tr>
<tr>
<td>NB</td>
<td>( O(mn + nc) )</td>
<td>( O(mn + nc) )</td>
</tr>
<tr>
<td>NN</td>
<td>( O(mn + nc) )</td>
<td>( O(mn + nc) )</td>
</tr>
<tr>
<td>GDA</td>
<td>( O(mn^2 + n^3) )</td>
<td>( O(mn^2 + n^3) )</td>
</tr>
<tr>
<td>PCA</td>
<td>( O(mn^2 + n^3) )</td>
<td>( O(mn^2 + n^3) )</td>
</tr>
<tr>
<td>ICA</td>
<td>( O(mn^2 + n^3) )</td>
<td>( O(mn^2 + n^3) )</td>
</tr>
<tr>
<td>k-means</td>
<td>( O(mn^2 + n^3) )</td>
<td>( O(mn^2 + n^3) )</td>
</tr>
<tr>
<td>EM</td>
<td>( O(mn^2 + n^3) )</td>
<td>( O(mn^2 + n^3) )</td>
</tr>
<tr>
<td>SVM</td>
<td>( O(mn^2 + n^3) )</td>
<td>( O(mn^2 + n^3) )</td>
</tr>
</tbody>
</table>

Time-complexity analysis

Data characteristics

[Chu et al., NIPS 2007]
Challenge #5: The N-d-k problem (k)


• What happens when K= K + 1 ? (adding a new class)
  • Engineered features may not be good enough.
  • Trained model has to relearn from the entire feature set without guarantees on accuracy.
Concluding Thoughts

Dimensions of Big Data

- **Volume**
  - Hadoop, MPP, Spider
  - Archival, Reports, Discovery
- **Velocity**
  - Streaming, Batch
- **Variety**
  - SQL, NoSQL, Graph

Analytical Requirements

- **Programming**
  - MapReduce, MPI, Threads
- **Data-Parallel**
  - Task-parallel
- **Complexity of Algorithms**
  - Linear, Iterative, \( > O(N^2) \)
- **Compute on Data**
  - Retrieval, Machine Learning
- **Speed of Execution**
  - Real-time, Feasibility

What can be scaled up?

- Compute
  - Storage
  - Memory
  - Cores

What aspect of data that needs scale up?

- Volume
- Velocity
- Variety

What aspect of algorithm that needs scale up?

- Programming
- Data-Parallel
- Complexity of Algorithms
- Compute on Data
- Speed of Execution

I/O ?

- Network ?

• Future
  - We need benchmarks before we make big investments. (Fox et al., 2014)
Concluding Thoughts

• Storage/Memory and Memory/Compute Ratios that are critical for machine learning are smaller than Storage/Compute Ratio.

• Associative memory and cognitively-inspired architectures may prove better than the Von-Neuman “store-fetch-execute paradigm”.
  – May be time to redesign from scratch.

• The machine learning algorithms that scale all use either data-parallelism or the “dwarves of parallel computing in some form”.
  – Encouraging because – gives us an intuition to build custom “hardware” for learning algorithms.

• We have done well so far by treating – “Analysis as a retrieval problem” – We can do better.
Thank You

• Questions ?