Graph Processing

COS 518: Advanced Computer Systems
Lecture 12
Mike Freedman
[Content adapted from K. Jamieson and J. Gonzalez]

Graphs are Everywhere

Social Network
Collaborative Filtering

Probabilistic Analysis

Text Analysis

Concrete Examples

Label Propagation
Page Rank

Label Propagation Algorithm

- Social Arithmetic:
  - 50% What I list on my profile
  - 40% Sue Ann Likes
  - + 10% Carlos Like
  - I Like: 60% Cameras, 40% Biking

- Recurrence Algorithm:
  \[ Likes[i] = \sum_{j \in \text{Friends}(i)} W_{ij} \times Likes[j] \]
  - iterate until convergence

- Parallelism:
  - Compute all \( Likes[i] \) in parallel
**PageRank Algorithm**

- PageRank of $u$ is dependent on PR of all pages linking to $u$, divided by the number of links from each of these pages.
- Recurrence Algorithm:
  \[ PR[u] = \sum_{v \in B_u} PR[v] / L[v] \]
  - iterate until convergence
- Parallelism:
  - Compute all $PR[u]$ in parallel

**Properties of Graph Parallel Algorithms**

- Dependency Graph
- Factored Computation
- Iterative Computation

**Map-Reduce for Data-Parallel ML**

- Excellent for **large data-parallel** tasks!

**Problem: Data Dependencies**

- MapReduce *doesn’t efficiently express* data dependencies
  - User **must code** substantial data transformations
  - Costly **data replication**

**MapReduce?**

- Feature Extraction
- Algorithm Tuning
- Basic Data Processing
- Label Propagation
- Belief Propagation
- PageRank
- Neural Networks
- Tensor Factorization
- Deep Belief Networks
- Lasso
- Kernel Methods
Iterative Algorithms

- MR doesn't efficiently express iterative algorithms:

MapAbuse: Iterative MapReduce

- Only a subset of data needs computation:

MapAbuse: Iterative MapReduce

- System is not optimized for iteration:

ML Tasks Beyond Data-Parallelism

- Data-Parallel

Map Reduce

GraphLab

- Graph-Parallel

Graphical Models
Gibbs Sampling
Belief Propagation
Variational Opt.

Semi-Supervised Learning
Label Propagation
CoEM

Collaborative Filtering
PageRank
Tensor Factorization

Triangle Counting

Feature Extraction
Cross Validation
Computing Sufficient Statistics
This week’s lectures

• Graph processing
  – Why relationships, sampling, and iterations often use in graph processing not well fit by MapReduce
  – How to take a graph-centric processing perspective

• Machine learning
  – These are solving one type of ML algorithm
  – What other systems are needed, particularly given heavy focus on iterative algorithms

Today’s readings

• Streaming is about unbounded data sets, not particular execution engines
• Streaming is in fact a strict superset of batch, Lambda Architecture destined for retirement
• Needs of good streaming systems: correctness and tools for reasoning about time.
• Differences between event time and processing time, and the challenges they impose

Today’s readings

• What about major data processing approaches for bounded & unbounded data?
• Challenges/needs for unbounded include:
  – time-agnostic, approximation, windowing by processing time, windowing by event time
• Key mechanisms (in Cloud DataFlow)
  – Watermarks: ideal vs. heuristic
  – Triggers
  – Discarding, accumulating, accumulating + retracting