Large Scale Network Modeling and Prediction using GraphLab: a new framework for parallel machine learning

Yocheng Low, Joseph Gonzalez, Aapo Kyrola, Danny Bickson, Carlos Guestrin, Joseph M. Hellerstein
Previous activities

- Machine learning tools within MANETS for estimation and placement
- Robust distributed inference in MANETS
- Handling large scale data using multicore/distributed machines
Current activities

- Large scale network modeling and predictions

- The GraphLab parallel/distributed ML framework
  - An efficient tool for computing our network models and predictions
Modeling Large Scale Networks
Why is network modeling useful?

- Characterize **normal** network behavior
- Identify **anomalies** / security threats
- **Predict** future behavior based on history
- Optimize available resources
Challenges of network modeling

- Huge amounts of data

- Daily stats collected from the PlanetLab network using PlanetFlow:
  - 662 PlanetLab nodes spread over the world
  - 19,096,954,897 packets were transmitted
  - 10,410,216,514,054 bytes were transmitted
  - 24,012,123 unique IP addresses observed
Heavy tailed traffic

Bandwidth/port number distribution is heavy tailed

Bandwidth distribution is heavy tailed: 1% of the top flows are 19% of the total traffic
Challenges of Network Modeling

- Adversarial activities

![Graph comparing DARPA 1989 training data with attack data](image)
Our solution

1) Model network using **probabilistic graphical models**

2) Develop the theory to support inference with **heavy tailed** distributions

3) Devise **distributed** ML algorithms for inference of very large scale data

4) **Build an efficient framework** for implementing the algorithms on large scale data
Theoretical Contributions

- Novel linear-stable model
  - A linear graphical model with heavy-tailed stable distribution

In a communication network, the model defines a linear-stable channel
Equivalent to, the developed model can be used for modeling linear channel with additive stable noise.

Useful in information theoretic problems like: multiuser detection, beamforming, MIMO decoding, equalization etc.

Developed tools for computing closed-form ML and MMSE detectors.
Inference in linear-stable model

- Closed-form solution for exact inference
  - Inference computed in the Fourier domain
- Iterative algorithms for efficient approximate inference
Modeling network flows

- Find a low dimensional explanation for the data
- Treat flows as entries in a sparse tensor \((\text{src} \times \text{dest} \times \text{time})\)

Application: given a source destination pair, predict amount of traffic transmitted

Model is used both for prediction and anomaly detection
The problem

- For 10 PlanetLab days, tensor size is about $50,000,000 \times 50,000,000 \times 240 \sim 2^{50}$.
- How can we predict for such a large data?
- How can we scale for even larger data?

**Solution:** use GraphLab!
GraphLab: a new framework for parallel machine learning
Parallel Programming is Hard

- Designing **efficient** Parallel Algorithms is **hard**
  - Race conditions and deadlocks
  - Parallel memory bottlenecks
  - Architecture specific concurrency
  - Difficult to debug

Graduate students **repeatedly** address the same parallel design challenges

Avoid these problems by using **high-level abstractions.**
Map-Reduce good for data-parallel problems. But, most ML problems are more complex.

Data-Parallel

- Feature Extraction
- Cross Validation

Complex Parallel Structure

- Kernel Methods
- Belief Propagation
- Tensor Factorization
- Deep Belief Networks
- Neural Networks
- SVM
- Sampling

Our target!
ML: Common Properties

1) Sparse Data Dependencies
   • Sparse Primal SVM
   • Tensor/Matrix Factorization

2) Local Computations
   • Sampling
   • Belief Propagation

3) Iterative Updates
   • Expectation Maximization
   • Optimization
GraphLab is the **Solution**

- Simplifies the design of parallel programs:
  - Allows for high level algorithm specification
  - Abstract away hardware/platform issues
  - Automatic data synchronization
  - Allows for rapid deployment of iterative algorithms on big datasets
GraphLab

Data Graph

Shared Data Table

Flexible scheduling

GraphLab Model

Update Functions and Scopes
Data Graph

A Graph with data associated with every vertex and edge.

Flow prediction problem:
Nodes are machines
Packet flows are edges (in time)
The task: predict unobserved edges
**Update Functions** are operations which are applied on a node and transform the data in the scope of the node.

Flow prediction problem: Update function perform a Monte Carlo Sample:
- Read samples from neighboring nodes
- Compute conditional probabilities
- Sample the current node
GraphLab Implementation

- Implemented on C++, tested on Linux/MacOS
- Open source multicore version available for download on http://graphlab.ml.cmu.edu
  - Reported in UAI 10’
- Distributed version to be released soon
  - Under submission
- Tested using a large set of applications:
  - Interior point methods
  - Belief propagation
  - Lasso
  - CoEM
  - Video cosegmentation
  - Among others…
Experimental Results

- Solve network traffic modeling problem using GraphLab distributed ML engine.

- PlanetFlow data
  - 90% of the flows are used as training data.
  - 10% of the flows are test data.

- Performance metric: RMSE (square root of mean square error.)

- Data size 250GB compressed (only 10 days!)
Experimental results

- Quality of predictions
  - Normalized flow magnitude using log scale 1-5
  - Prediction accuracy: RMSE 0.3
  - Quite a surprising result considering the fact we have 24,000,000 x 24,000,000 x 24 potential flows a day!
  - For comparison, movie rating prediction on Netflix data: RMSE 0.88
Experimental results

- Benefit of using Graphlab:
  - For “small” dataset of 100,000,000 observed flows:
    - Matlab took 8 hours,
    - while GraphLab 30 minutes (with 8 cores)

- Almost linear speedup
- Communication overhead does not grow linearly
Utilized ML techniques for improved network modeling
- Handle heavy tailed distributions
- Handle very large scale data

Designed and implemented the GraphLab framework
- Allows researchers specify high level algorithms avoiding distributed programming pitfalls
- Applied GraphLab to large scale network modeling problems
Future work

- Refine developed ML techniques for modeling and prediction on other datasets
  - MANETs
  - Security related logs

- Extend GraphLab framework to other settings
  - GPUs
  - Mobile devices?
Our research is based on the fundamental principles of active protocol monitoring for both performance, stability and adversary handling, of employing communication channel diversity for robust end-to-end operation in the face of failures and deliberate attacks, and of exploiting cross-layer interaction for predicting the effects of performance changes caused by layer-specific failures and attacks on end-to-end MANET operation. We employ design and analysis techniques found in network theory, statistics, game theory, cryptography, economics and sociology, and system theory to develop, design and analyze models, tools, and mathematical representations for predicting performance and prescribing resilient, secure MANETs.