Sampling and Inference Problems for Big Data in the Internet and Beyond

Nick Duffield
Rutgers University – DIMACS
http://nickduffield.net/work
Structure of Large ISP Networks

- City-level Router Centers
- Backbone Links
- Peering with other ISPs
- Network Management & Administration
- Service and Datacenters
- Access Networks: Wireless, DSL, IPTV
- Downstream ISP and business customers
Measuring the ISP Network: Data Sources

- **Status Reports**: Device failures and transitions
- **Protocol Monitoring**: Routers, Wireless handovers
- **Loss & Latency**: Roundtrip to edge
- **Link Traffic Rates**: Aggregated per router interface
- **Customer Care Logs**: Reactive indicators of network performance

- **Router Centers**
- **Backbone**
- **Datacenters**
- **Management**
- **Traffic Matrices**: Flow records from routers
- **Active probing**
Three challenges for ISP data analysis

• Scale: some datasets are enormous
  – IP Flow Records: 100s TB per day
  – Cellular network: device handovers, physical layer

• Incompleteness:
  – Need link performance; but only measure aggregate path performance
  – Less detailed traffic measurement out in access networks
  – Less instrumentation lower down the protocol stack

• Complexity:
  – Interactions between hardware, protocols, traffic, applications, customers
  – Which factors best determine customer experience?
  – Noisy or missing data
  – Sparseness: individual events leave data footprints over many subsystems
Outline

• Big Data in Communications Networks
  – Data scale: sampling traffic measurements
  – Data incompleteness: inference and tomography
  – Data complexity: machine learning the factors of network performance

• Extending the big data research agenda
  – Data scale: sampling massive graphs
  – Data incompleteness: urban informatics
  – Data complexity: learning cellular pathways, healthcare analytics

• Outlook and Summary
Data Scale: Summarization and Sampling
Data Scale: Why Summarize (ISP) Big Data?

- When transmission bandwidth for measurements is limited
  - Not such a big issue in ISPs with in-band collection
- When raw accumulation is not feasible
  - High rate streaming data
  - Maintain historical summaries for baselining, time series analysis
- To facilitate fast queries
  - When infeasible to run exploratory queries over full data
- As part of hierarchical query infrastructure:
  - Maintain full data over limited duration window
  - Drill down into full data through one or more layers of summarization

Argue that sampling is a flexible method to accomplish this
Sampling as a Mediator of Constraints

Data Characteristics
(Heavy Tails, Correlations)

Sampling

Resource Constraints
(Bandwidth, Storage, CPU)

Query Requirements
(Ad Hoc, Accuracy, Aggregates, Speed)
Traffic Measurement in the ISP Network

Traffic Matrices
Flow records from routers

Router Centers
Backbone
Access
Business
Datacenters
Management
**Massive Dataset: Flow Records**

- **IP Flow**: set of packets with common key observed close in time
- **Flow Key**: IP src/dst address, TCP/UDP ports, ToS,…
- **Flow Records**:
  - Protocol level summaries of flows, compiled and exported by routers
  - Flow key, packet and byte counts, first/last packet time, some router state
  - Realizations: Cisco Netflow, IETF Standards
- **Scale**: 100’s TeraBytes flow records daily are generated in a large ISP
- **Used to manage network over range of timescales**:
  - Capacity planning (months),…., detecting network attacks (seconds)
- **Analysis tasks**
  - Easy: timeseries of predetermined aggregates (e.g. address prefixes)
  - Hard: fast queries over exploratory selectors, history, communications subgraphs
Abstraction: Keyed Data Streams

• Data Model: objects are keyed weights
  – Objects (x,k): Weight x; key k
    • Example 1: x = flow bytes, k = flow key (common endpoints of packets)
    • Example 2: k = account ID, x = account update

• Stream of keyed weights
  – \{(x_i, k_i): i = 1,2,\ldots,n\}

• Generic query: subset sums
  – \(X(S) = \sum_{i \in S} x_i\) \(S \subseteq \{1,2,\ldots,n\}\) i.e. total weight of index subset \(S\)
  – Typically \(S = S(K) = \{i: k_i \in K\}\) : objects with keys in \(K\)
    • Example 1: \(X(S(K))\) = total bytes to given IP dest address / UDP port
    • Example 2: \(X(S(K))\) = change in balances of one or more accounts

• Aim:
  – Compute fixed size summary of stream that can be used to estimate arbitrary subset sums with known error bounds
How to Sample and Estimate

- **Horvitz-Thompson Estimation:**
  - Object of size \( x_i \) sampled with probability \( p_i \)
  - Unbiased estimate \( x'_i = x_i / p_i \) (if sampled), 0 if not sampled: \( E[x'_i] = x_i \)

- **Linearity:**
  - Estimate of subset sum = sum of matching estimates
  - Subset sum \( X(S) = \sum_{i \in S} x_i \) is estimated by \( X'(S) = \sum_{i \in S} x'_i \)

- **Accuracy**
  - Exponential Bounds:
    - \( \Pr[ |X'(S) - X(S)| > \delta X(S)] \leq \text{Exp}[-g(\delta)X(S)] \)
    - Confidence intervals: \( X(S) \in [X^-(\varepsilon), X^+(\varepsilon)] \) with probability \( 1 - \varepsilon \)

- **Futureproof**
  - Don’t need to know queries at time of sampling
    - “Where/where did that suspicious UDP port first become so active?”
    - “Which is the most active IP address within than anomalous subnet?”
  - Retrospective estimate: subset sum over relevant keyset
  - Contrast: can’t drill down into aggregates
Matching Data to Analysis with Sampling

• Generic problem 1: Counting objects: weight $x_i = 1$
  – Uniform sampling with probability $p$ works fine
    • Estimated subset count $X'(S) = \#\{\text{samples in } S\} / p$
    • Relative Variance ($X'(S)$) = $(1/p - 1)/X(S)$
      – given $p$, get any desired accuracy for large enough $S$

• Generic problem 2: $x_i$ in Pareto distribution, a.k.a. 80-20 law
  – Small proportion of objects possess a large proportion of total weight
  – How to best to sample objects to accurately estimate weight?
  – Strawmen
    • Uniform sampling?
      – likely to omit heavy objects $\Rightarrow$ big hit on accuracy
      – making selection set $S$ large doesn’t help
    • Select $m$ largest objects?
      – biased & smaller objects systematically ignored
Cost Optimization

- Independent sampling from n objects with weights \( \{x_1, \ldots, x_n\} \)
- Goal: find the “best” sampling probabilities \( \{p_1, \ldots, p_n\} \)
- Horvitz-Thompson: unbiased estimation of each \( x_i \) by
  \[
  x'_i = \begin{cases} 
  \frac{x_i}{p_i} & \text{if weight selected} \\
  0 & \text{otherwise}
  \end{cases}
  \]

- Two costs
  1. Estimation Variance: \( \text{Var}(x'_i) = x_i^2 \left(\frac{1}{p_i} - 1\right) \)
  2. Expected Sample Size: \( \sum_i p_i \)

- Minimize Linear Combination Cost: \( \sum_i \left(x_i^2 \left(\frac{1}{p_i} - 1\right) + z^2 p_i\right) \)
  - \( z \) expresses relative importance of small sample vs. small variance
Minimal Cost Sampling: IPPS

- **IPPS**: Inclusion Probability Proportional to Size
- **Minimize**
  - $\text{Cost } \sum_i \left( x_i^2 \left( \frac{1}{p_i} - 1 \right) + z^2 p_i \right)$ subject to $1 \geq p_i \geq 0$
- **Solution**
  - $p_i = p_z(x_i) = \min\{1, x_i / z\}$
    - small objects ($x_i < z$) selected with probability proportional to size
    - large objects ($x_i \geq z$) selected with probability 1
  - Call $z$ the “sampling threshold”
  - Unbiased estimator $x_i/p_i = \max\{x_i, z\}$

- Perhaps reminiscent of importance sampling, but not the same:
  - make no assumptions concerning distribution of the $x$

[Duffield, Lund, Thorup, IEEE ToIT, 2004]
IPPS Stream Sampling into Reservoir of Size $m$

- Each arriving item:
  - Provisionally include item in reservoir
  - If $m+1$ items, discard 1 item randomly
    - Recalculate threshold $z$ to sample $m$ items on average: $z$ solves $\sum p_z(x_i) = m$
    - Discard item $i$ with probability $q_i = 1 - p_z(x_i)$
    - Adjust $m$ surviving $x_i$ with Horvitz-Thompson $x'_i = x_i / p_i = \text{max}\{x_i, z\}$

- Efficient Implementation:
  - Computational cost $O(\log m)$ per item, amortized cost $O(\log \log m)$

Example: $m=9$

- Implemented in ISP measurement infrastructure today

Variations on the Theme of Cost

1. **Fixed Size IPPS Sampling**
   - Problem: Independent sampling => variable sample size
   - Solution: fixed size sampled of flow records (e.g. per hour, per router interface)
     

2. **Structure Aware Sampling**
   - Problem: many queries concern subset sums over IP address prefixes
   - Solution: improve estimation accuracy by cost-based localizing sampling to prefixes
     
     [Cohen, Cormode, Duffield; PVLDB 2011]

3. **Fair Sampling**
   - Problem: sampling based estimates less accurate for small customers
   - Solution: adaptively share sampling budget over different customer’s flows
     
     [Duffield; Sigmetrics 2012]

4. **Stable Sampling**
   - Problem: churn in membership of sample, implicit cost
   - Solution: include explicit sample changeout cost to penalize churn
     
     [Cohen, Cormode, Duffield; 2013]
Estimation Accuracy in Practice

- Estimate any subset sum comprising at least some fraction $f$ of weight
- Suppose: sample size $m$
- Analysis: typical estimation error $\varepsilon$ (relative standard deviation) obeys

$$\varepsilon \leq \frac{1}{\sqrt{fm}}$$

**Estimate fraction** $f = 0.1\%$ with typical relative error $12\%$

- $2^{16} = \text{same storage needed for aggregates over 16 bit address prefixes}$
  - But sampling gives more flexibility to estimate traffic within aggregates
Heavy Hitters: Exact vs. Aggregate vs. Sampled

- Sampling does not tell you where the interesting features are
  - But does speed up the ability to find them with existing tools
- Example: Heavy Hitter Detection
  - Setting: Flow records reporting 10GB/s traffic stream
  - Aim: find Heavy Hitters = IP prefixes comprising ≥ 0.1% of traffic
  - Response time needed: 5 minute
- Compare:
  - Exact: 10GB/s x 5 minutes yields upwards of 300M flow records
  - 64k aggregates over 16 bit prefixes: no deeper drill-down possible
  - Sampled: 64k flow records: any aggregate ≥ 0.1% accurate to 10%
Data Incompleteness: Inference and Tomography
Traffic Measurement in the ISP Network

- **Router Centers**
- **Backbone**
- **Access**
- **Datacenters**
- **Management**
- **Business**

**Traffic Matrices**
Flow records from routers

**Link Traffic Rates**
Aggregated per router interface
Incomplete Data: Traffic Matrix Tomography

• What we wanted:
  – traffic matrix $M_{ij} = \{\text{traffic rates } \text{src}_i \rightarrow \text{dst}_j \}$
  – compute from flow records
    • measured in network core, but not at edge

• What we had:
  – link aggregate rates $L_{xy} = \{\text{traffic rate on each network link } (x,y) \}$
  – measured ubiquitously by routers:
    • “SNMP statistics”, 5 minute granularity

• Linear system $L = A \cdot M$:
  – $A$ = routing matrix = incidence of src-dst paths over links
    • Link rate = sum of traffic on src-dst paths routed over that link

• Invert?
  – Under-constrained linear system

• Use Vardi’s network tomography?
  – Computation not scalable to large networks
Incomplete Data: Traffic Matrix Tomography

- **Underconstrained linear system** $L = A \cdot M$
  - Find constraints (data or model)
- **Gravity model** $M^{grav}$ for $M$
  - Product form $M^{grav}(src \rightarrow dst) \propto L(src \rightarrow) \cdot L(\rightarrow dst)$
  - Does NOT satisfy constraints $L = A\cdot M$ in general
- **Tomogravity = Tomography + Gravity**
  - $M^* = \arg\min_M \|M - M^{grav}\|^2$ subject to $L = A\cdot M$
- **Typically accurate to with 10% of true $M$**
  - Used in ISPs, Cisco

[Zhang, Roughan, Duffield, Greenberg; Sigmetrics 2003]
ACM Sigmetrics Test of Time Award 2013
Data Complexity:
Machine Learning the Factors of Performance
Data Complexity: Learning & Customer Experience

- Customer experience determined by performance of many systems
  - Learn performance features that best determine customer experience
  - Ground truth: customer care calls, technician resolutions
    - Noise: imperfect classification of problems, attribution of problem time/location
  - Machine Learning: boosting combines many weakly predictive features

- Benefits:
  - Better customer experience: fix problems before they result in call
  - Reduce management costs: fewer calls, suggest resolutions to technicians

[Jin, Duffield, + others: CoNext 2010; W-MUST 2011, CellNet 2012, Infocom 2013]

- Many other applications of machine learning
  - Learning signatures of malicious traffic
    [Duffield, Haffner, Krishnamurthy, Ringberg : Infocom 2009]
  - Learning type of application that generated traffic from collective traffic patterns
    [Jin, Duffield, Haffner, Sen; TKDD 2012]
Emerging Research & Big Data Challenges

• Data Scale:
  – Massive graph sampling and cybersecurity

• Data Incompleteness:
  – Data fusion and inference in urban informatics

• Data Complexity:
  – Machine learning cellular pathways
Data Scale:
Massive Graph Sampling
Massive Graph Sampling

- **Graph Service Providers**
  - Search providers: web graphs
  - Online social networks
    - Facebook: $\sim 10^9$ users (nodes), $\sim 10^{12}$ links
  - ISPs: communications graphs
    - From flow records: node = src or dst IP, linked if traffic flows between them

- **Graph service provider perspective**
  - Already have all the data, but how to use it?

- **Want a general purpose graph sample design that can:**
  - Quickly answer exploratory queries, compactly archive snapshots
  - Amplify the capacity of graph parallel storage systems / processing models
    - MapReduce, Pregel, Giraph, GraphX
  - Sample design that matches data characteristics to query needs
    - Sampling links, nodes, subgraphs, labels, with the appropriate weights
Communications graphs and attack detection

- Node = IP address
- Directed edge = flow from source node to destination node

• Hard to detect against background
• Known attacks:
  - Signature matching based on partial graphs, flow features, timing
• Unknown attacks:
  - exploratory & retrospective analysis
  - preserve accuracy if sampling?
• Current work: subgraph sampling
Subgraph estimation: triangles and counting

- **Streaming Graph Updates**
  - Links in communications graphs, OSN updates, ...
- **Hot topic: sample-based triangle counting**
  - Triangles: simplest non-trivial representation of node clustering
    - Regard as prototype for more complex subgraphs of interest
  - State of the art: stream sample algorithms, specific for triangle counting
    - Jha et.al. SIGKDD 2103; Pawan et.al. VLDB 2013
- **Graph Sample and Hold** [Ahmed, Duffield, Neville, Kompella, 2014]
  - General framework for subgraph counting; e.g. triangle counting
  - Similar accuracy to previous state of art, but using smaller storage

![Horvitz-Thompson count estimator](image)
Data Incompleteness:
Fusion and inference for urban informatics
Urban Informatics

- Data-driven analysis of economic activity, human behavior, mobility patterns, resource consumption,…
  - Predictive models of connections between these
  - Improve urban services; provide incentives to optimize resource usage
  - Range of timescales: from urban planning to emergency management

- Data sources
  - Transport & service usage logs, sensors, location, census, emergency service logs

- Challenges
  - Data fusion, inference, privacy, organizational diversity, political churn

- Example: infer traffic matrix for transport planning
  - Infer from rider entry logs?
    • underconstrained inverse problem
  - Supplement with constraints from data to solve
    • ticket sales / billing information
    • online timetable queries
    • cellular / wi-fi mobility data
    • seasonal / geographical background
Data Complexity: Machine learning cellular signaling pathways
Bioinformatics

• Cell signaling pathways
  – Understand systems biology behind cancer, autoimmunity, diabetes
  – Big data role:
    • automate integration and correlation of vast literature on individual pathways
    • discover pathway interactions, disease causes, potential disruptive therapies
  – Challenges
    • Unstructured data, guide data-driven discovery with domain knowledge

• Genomics
  – Genetic sequencing become cheaper: < $1k
  – Big data roles
    • Manage data volumes (efficient storage, query, computation)
    • Scaleable machine learning applications, e.g. functional genomics
  – Challenges
    • Big data per individual (~1TB), few individuals (1,000s, smaller if specific)
    • Data-driven discovery needs guidance from domain knowledge
Healthcare Analytics

- Clinical Data Analytics
  - Electronic Health Records
- Rich source of data illuminating connections
  - Demographic, lifestyle, diagnosis, treatments, outcomes
- Data analytics
  - Use ML to develop predictive models; incorporate domain knowledge
  - Cause of illness; spread of disease; efficacy of treatments; cost-benefit
    - Candidates for preventive care
    - Understand and prepare for seasonal variations
    - Longitudinal studies

- Challenges
  - Unstructured data (text, imaging); Data privacy
Outlook

• Big Data in Communications Networks
  – Acquisition, analysis, applications of large and complex operational datasets
  – Network state, usage, performance, configuration, customer care calls
  – Mathematically grounded algorithms for network measurement & analysis in ISPs

• Disciplines
  – Applied probability, statistics, machine learning, algorithms, networking

• Emerging applications: common problems, methods, disciplines
  – Massive graph sampling: communications graph and cybersecurity
  – Urban informatics: fusion of multiple views of urban resource usage
  – Bioinformatics: small number of huge datasets: analytics with domain knowledge
  – Healthcare analytics: discover relations between lifestyle, treatments and outcomes
  – Data analytics in agriculture: environmental sensing, relating crop treatment to yields
  – Power grid: sensor array data, low dimensional event leaves wide traces
  – Privacy: cost-utility trade-off for network flows of information
  – Financial data analytics: sentiment analysis and relation to markets prices

• Many of the same challenges: scale, incompleteness, complexity
Thank you!