ECEN 689
Special Topics in Data Science for Communications Networks

Nick Duffield
Department of Electrical & Computer Engineering
Texas A&M University

Lecture 2
Sampling in Network Traffic Measurement
Measuring the ISP Network: data sources

Router Centers

Backbone

Peering

Access

Business

Datacenters

Management

IP Traffic Flow Records
Generated by routers
Flow measurement in routers

- Routers maintain statistics on flows in a **flow table**
  - Each flow: key, #packets, #bytes, first & last packet times, …

- Each packet
  - If no entry for packet key in flow table, instantiate new entry
    #packets(key) = #bytes(key) = 0; first_packet_time(key) = timestamp, …
  - Update flow entry
    #packets(key)++ ; #bytes(key) += bytes; last_packet_time(key) = timestamp, …
Generic flow queries

• Represent flows by
  – (key, packets, bytes, $t_{\text{first}}, t_{\text{last}}, \ldots$)

• Subset sum (volumetric) queries
  – $X(K,T) =$ total #flows or #packets or #bytes in flows with key $\in K$, observed during interval $T$ containing $t_{\text{first}}$

• Flow distribution query
  – What is the distribution of bytes per flow and packet per flow?
    • E.g. What proportion of flows have 1 packet?
  – Restricted to key $\in K$, e.g. flows using TCP STN flag

• Flow graph distribution query:
  – What is the distribution of number of flows per DstIP
  – Restricted to key $\in K$, e.g. flows using particular DstPrt (application)
The need for sampling

- Problem: flow table key lookup at line rate too CPU intensive
- Workaround: packet sampling
  - Select 1 out of N packets on average, some integer parameter N
  - Reduce load for key lookup
- Form flow records as before,
  - from substream of sampled packets only
- How to get approximate answers to queries after sampling?
Independent packet sampling

- Each packet sampled independently with probability \( p \)
  - Also known as Bernoulli sampling
- Statistically simplest form of packet sampling
- Implementation?
  
  ```
  foreach (packet) {
    select(packet) if ( rand(0,1] < p ) ;
  }
  ```

- Or use hash of packet content onto (0,1]
  
  ```
  foreach (packet) {
    select(packet) if ( hash(packet) < p ) ;
  }
  ```

- Question: what are possible statistical pitfalls of hash?
- Computational cost: generate random number for each packet
Better: skip counting

- Find random index of next packet to be sampled
- Let $r \in \{1, 2, \ldots\}$ be random index of next packet to be sampled
- Geometric distribution:
  \[
  \Pr[r = i] = p(1-p)^{i-1} \quad \text{or equivalent CCDF: } \Pr[r > i] = (1-p)^i
  \]
- Generating geometric random variate:
  \[
  i^* = \text{ceil}(\frac{\log u}{\log(1-p)}) \quad \text{where } u \text{ is } \text{Unif}(0,1]
  \]
  \[
  \text{ceil}(x) \text{ is smallest integer } \geq x
  \]
- Check: \{i^* > i\} = \{ (\log u)/(\log(1-p)) > i\} = \{ u < (1-p)^i \}
  \[
  \Pr[i^* > i] = (1-p)^i \text{ as required. (Check details for yourself)}
  \]
Simpler approaches used in NetFlow

• Previous is fine on general purpose computing platform
  – Libraries for log function, random numbers; efficient algorithms

• Simpler approaches
  – Periodic 1 in N sampling
    • Select packet N, 2N, 3N,…
  – Stratified 1 in N sampling
    • Pick 1 packet at random from each block of N successive packet

• Marginal sampling probability of each packet is 1/N for both

• Different joint packet sampling probabilities
  – Also different to independent sampling
  – E.g. different probabilities to sampling two successive packet
  – But packets of any flow are “randomized” by other packets

• More details on NetFlow Sampling
  – http://tinyurl.com/cisco-netflow-samplers
Estimation from packet sampled flow records

- Original flows (key, #packets) = (k,p)
- Generic subset sum query
  - True value $X(K) = \#\text{packets in flows with key } k \in K$
- How to estimate from packet sampled flow records
  - Intuition: sampling yields about $1/N$ of the original packets
  - Multiply number of sampled packets by $N$ to estimate #original packets
- Key idea: can apply the idea to any subset of packets
  - Estimate: $X(K)$ by $X'(K) = N \times \#\text{packets in sampled flows with key } k \in K$
Unbiased estimation

• Consider a single item (e.g. packet)
• Equipped with a size $x$ (e.g. #bytes)
• Sampled with probability $q$
• The **Horvitz-Thompson Estimator** of $x$ is

$$x' = \begin{cases} 
\frac{x}{q} & \text{if the item is selected} \\
0 & \text{otherwise}
\end{cases}$$

• Theorem: $x'$ is **unbiased** estimator of $x$: Expectation $E[x'] = x$
  
  $E[x] = \text{sum of estimate values weighted by their probabilities}$
  
  $= Pr[\text{item sampled}] \times \frac{x}{q} + Pr[\text{item not sampled}] \times 0$
  
  $= q \times \frac{x}{q} + (1-q) \times 0$
  
  $= x$
Subset sum estimation with Horvitz-Thompson

- Set of items with weights \( \{x_1, x_2, \ldots, x_n\} \)
- Individual sampling probabilities \( \{q_1, q_2, \ldots, q_n\} \)
- Subset sum \( X(S) = \sum_{i \in S} x_i \) for any subset \( S \subset \{1,2,\ldots,n\} \)
- Estimate subset sum using sum of constituent estimates
  \[
  X'(S) = \sum_{i \in S} x'_i \quad \text{where} \quad x'_i = \frac{x_i}{q_i} \text{ if } i \text{ selected, } 0 \text{ otherwise}
  \]
- Theorem: \( X'(S) \) is an unbiased estimator of \( X(S) \)
  - Proof uses linearity of expectation
    \[
    E[X'(S)] = E[\sum_{i \in S} x'_i] = \sum_{i \in S} E[x'_i] = \sum_{i \in S} x_i = X(S)
    \]
  - Doesn’t require independent sampling

- More on Horvitz-Thompson: Kolaczyk. Ch. 5.2
- See also Duffield review article
Subset sum estimation for packets (and bytes)

• \(X(K) = \#\{\text{packets in samples flows with key } k \in K\}\)  
  \[= \sum_{i \in S(K)} x_i, \quad x_i = 1; \quad S(K)= \text{packets with key } k \in K\]

• For brevity will mostly write \(X\) for \(X(K)\) and \(S\) for \(S(K)\)

• Packets sampled with probability \(1/N\)

• \(X' = N^* \#\text{packets in sampled flows with key } k \in K\)  
  \[= \sum_{i \in S(K)} x'_i \quad \text{with } x'_i = N \text{ if packet is sampled, } 0 \text{ otherwise}\]

• Recognize \(X'\) as the HT estimator of \(X\): \(E[X'] = X\)

• Can make similar construction for byte estimation
Estimation Accuracy

• Think of independent 1 in N sampling from X packets
• Number of samples has binomial $B(X, 1/N)$ distribution
• Estimate $X'$ is distributed as $N^*B(X, 1/N)$ i.e.

$$\Pr[X'=xN] = \binom{X}{x} N^{-x} (1 - 1/N)^{x-x}$$

• Example:
  - $X = 5,000$,
  - $N = 100$,
  - 1,000 trials of $X'$
Accuracy and Bounds

- HT estimation: weights \( \{x_1, \ldots, x_n\} \) & probabilities \( \{q_1, q_2, \ldots, q_n\} \)
- Initially examine accuracy of subsets sums \( X = X(K) \) via variance:
  \[
  \text{Var}(X') = \mathbb{E}[ (X' - X)^2 ] = \mathbb{E}[ (X')^2 ] - X^2
  \]
- Assume independent sampling for the moment: \( x'_i \) are independent
  \[
  \text{Var}(X') = \text{Var}(\sum_{i \in S} x'_i) = \sum_{i \in S} \text{Var}(x'_i)
  \]
- Compute variance for HT estimate of individuals \( x'_i \):
- Exercise:
  \[
  \text{Var}(x'_i) = x_i^2 (1/p_i - 1) \quad \text{and so} \quad \text{Var}(X') = \sum_{i \in S} x_i^2 (1/p_i - 1)
  \]
Confidence Intervals for HT estimates?

• Common to think of confidence intervals in terms of the normal distribution
  – Because the width of the distribution is characterized by the variance

• But distribution of HT estimator not normal; so either
  1. Find confidence intervals using variance without normal assumption
  2. Exploit simple form of the distribution of the estimator
Towards Variance Based Bounds

• Theorem (Markov inequality)
  – Let Y be non-negative random variable with $E[Y] < \infty$
  – Then $Pr[ Y \geq a ] \leq E[Y] / a$

• Proof:
  
  $E[Y] = E[Y | Y < a] \cdot Pr[ Y < a] + E[Y | Y \geq a] \cdot Pr[ Y \geq a]$
  
  $\geq 0 + a \cdot Pr[ Y \geq a]$

• Use: take $a = n \cdot E[Y]$. Then $Pr[ Y \geq n \cdot E[Y] ] \leq 1/n$
  – Probability Y exceeds its expectation by factor n is less than 1/n

• Very general, but does not depend on (or exploit) variance

• More on bounds: Mitzenmacher & Upfal, Ch. 3
Variance Based Bounds

- Markov Inequality: \( \Pr[ Y \geq a ] \leq E[Y] / a \), for \( Y > 0, E[Y] < \infty \)
- Now let \( Y = (Z - E[Z])^2 \) for a random variable \( Z \), set \( a = b^2 \)
- Apply Markov inequality (assume \( E[(Z - E[Z])^2 = \text{Var}(Z) < \infty) \)

\[
\Pr[ (Z - E[Z])^2 \geq b^2 ] \leq E[ (Z - E[Z])^2 ] / b^2
\]

- In other words (this is called Chebychev’s Inequality)

\[
\Pr[ |Z - E[Z]| \geq b ] \leq \text{Var}[Z] / b^2
\]

- Interpretation: set \( b = n^* \sqrt{\text{Var}[Z]} = n^* \text{StdDev}(Z) \)
  - Probability that \( Z \) further from its expectation than \( n^* \text{StdDev} \) is \( \leq 1/n^2 \)
Application to Packet Sampling

- Estimate packet counts \( x_i = 1 \); uniform sampling prob 1/N

- Chebychev Inequality for \( X' = X'(K) \)
  \[
  \text{Var}(X') = \sum_{i \in S} \text{Var}(x'_i) = X \cdot (N-1)
  \]
  \[
  \text{Prob}( | X' - X | \geq b ) \leq \frac{X \cdot (N-1)}{b^2}
  \]

- Rewriting with \( b = n \cdot X \)
  - a **large deviation** (of \( X'(K) \) proportional to its expectation \( X \))
    \[
    \text{Prob}( | X' - X | \geq n \cdot X ) \leq \frac{(N-1)}{(n^2 \cdot X)}
    \]

- Behavior of bound
  - Increasing in \( N \): less certainty if fewer sampled packets
  - Decreasing in \( X \): more certainty if estimating a larger subset
Exponential bounds

• Recall Markov Inequality:
  \[ \Pr[ Y \geq a ] \leq \frac{E[Y]}{a} \text{ for } Y \geq 0 \text{ with } E[Y] < \infty \]

• Obtained Chebychev using \( Y = (Z-E(Z))^2 \). Try similar.

• Let \( Y = e^{tZ} \) for a random variable \( Z \), and \( t \geq 0 \), and set \( a = e^{tz} \)
  \( \{Z \geq z\} = \{Y \geq e^{tz}\} \) since \( e^{tz} \) is increasing in \( z \) when \( t \geq 0 \)

• Apply Markov Inequality: \( \Pr[Z \geq z] \leq e^{-tz} E[e^{tZ}] \)

• Which value of \( t \) to take?
  – The value of \( t \) that minimizes the upper bound!
    \( \Pr[Z \geq z] \leq \min_{t \geq 0} e^{-tz} E[e^{tZ}] \)

• Next time: evaluate minimum for packet sampling model.