A Data Streaming Algorithm for Estimating Entropies of OD Flows

Ashwin Lall, Mitsu Ogihara
University of Rochester

Oliver Spatscheck, Jia Wang
AT&T Research

Chuck Zhao, Jim Xu
Georgia Tech

October 25th, 2007

Entropy: Definition

- Given n flows of sizes $a_1, \ldots, a_n$. Let $s \equiv \sum_i a_i$. The empirical entropy is defined as

$$H \equiv -\sum_i \frac{a_i}{s} \log\left(\frac{a_i}{s}\right) = \log(s) - \frac{1}{s}\sum_i a_i \log(a_i).$$

- E.g. A stream of packets (anonymized): 1, 2, 1, 3, 2, 1, 3, 1

$$s = 8$$

$$\sum_i a_i \log(a_i) = 4\log 4 + 2\log 2 + 2\log 2 = 12$$

$$H = \log(s) - \frac{1}{s}\sum_i a_i \log(a_i) = \log 8 - \frac{12}{8} = 1.5$$

OD Flow Entropy

- An OD flow is all the traffic between an ingress point and an egress point.
- Problem statement: measure the empirical entropies of all OD flows in an ISP network.
- We need to measure
  - the entropy norm $\sum_i a_i \log(a_i)$,
  - the volume of the OD flow (i.e., a traffic matrix element).

Motivation

- Entropy is a measurement of the diversity of the traffic
- Anomaly detection (profiling behavior; [Lakhina et al., 2005])
- DDoS attacks may not be detectable as simple volume changes, and may significantly increase traffic entropy across the whole network
- Entropy of OD flows may reveal more information than entropy of traffic at ingress routers

Solution Strategy

- Data streaming is concerned with processing a long stream of data items in one pass using a small working memory (called a sketch) in order to answer a class of queries regarding the stream.

- An (ϵ, δ)-approximation algorithm for θ is one that returns an estimate $\hat{\theta}$ with relative error more than ϵ with probability at most δ. That is $\Pr[|\hat{\theta} - \theta| \geq \epsilon\theta] \leq \delta$.

- We want a solution with Intersection Measurable Property (IMP), which none of the existing entropy estimation algorithms has.

- [Indyk, 2006] stable distributions algorithm for L_p norm estimation has IMP. L_p norm of n flows of sizes $a_1, \ldots, a_n$ is defined as $(\sum_i a_i^p)^{1/p}$.

Outline

- Indyk's L_p norm algorithm for single stream
- Using L_p norm to approximate entropy norm
- Extending Indyk's L_p norm algorithm to OD flows (IMP)
- Enhancing accuracy of Indyk's L_p norm algorithm

Indyk's L_p Norm Algorithm: Big Picture

- Each packet causes increments of stable distribution values to an array of counters
- At end of epoch, L_p norm can be inferred from the counter values
- More counters, more accurate the estimate.

Stable Distributions

- The p.d.f. of stable distribution $S(p), p \in (0, 2]$ is the Fourier transform of $e^{-|t|^p}$.

- $S(1)$ is the standard Cauchy distribution. $S(2)$ is the Gaussian distribution with mean 0 and standard deviation 2.

- p-stable property: for any constants $a_1, \ldots, a_n$ and random variables $X, X_1, \ldots, X_n$ with distribution $S(p)$

$$a_1 X_1 + \ldots + a_n X_n \quad \sim_d \quad \left(|a_1|^p + \ldots + |a_n|^p\right)^{1/p} X$$

Indyk's L_p Norm Algorithm: Details

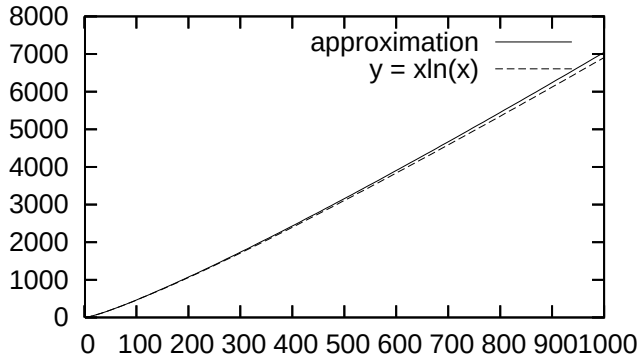
- For each potential flow i, draw a random number X_i from the p-stable distribution.
- For each packet in flow i, increment a real-valued counter by X_i.
- At end of epoch, the value of the counter will be $\sum a_i X_i$, which is distributed as $(\sum a_i^p)^{1/p} X$, where X is of distribution $S(p)$.
- To extract the quantity $(\sum a_i^p)^{1/p}$, do this independently for many counters, take the median of their absolute values, and divide by the median of $|X|$.

Using L_p Norm to Approximate Entropy Norm

- We can approximate $x \ln x$ by linear combinations of x^p for x on a fixed interval $[0, N]$ within relative error ϵ:

$$x \ln x \approx \frac{1}{2\alpha}(x^{1+\alpha} - x^{1-\alpha}), \text{ where } \alpha = \frac{\sqrt{\frac{6\epsilon}{1+6\epsilon}}}{\ln N}$$

  e.g. $N = 1000, \epsilon = 0.026, \alpha = 0.05$



- Proof: By Taylor expansion of $x^\alpha = e^{\alpha \ln x}$

Using L_p norm to approximate Entropy Norm

- Therefore we can use $L_{1+\alpha}$ and $L_{1-\alpha}$ norms to estimate the entropy norm

$$x \ln x \approx \frac{1}{2\alpha}(x^{1+\alpha} - x^{1-\alpha})$$

$$\sum a_i \ln a_i \approx \frac{1}{2\alpha}\left(\sum a_i^{1+\alpha} - \sum a_i^{1-\alpha}\right)$$

- In parallel, we have an elephant-detection module that handles (with high probability) all the flows of size greater than N.

Estimating L_p Norm of OD Flows

- Indyk's algorithm has Intersection Measurable Property (IMP).

$$\begin{aligned} \vec{O} &= \text{origin } L_p \text{ sketch} \\ \vec{D} &= \text{destination } L_p \text{ sketch} \\ \vec{O} - \vec{D} &= \text{component-wise subtraction of the sketches} \end{aligned}$$

- The OD flow L_p norm estimator is $\left(\frac{\Lambda(\vec{O})^p + \Lambda(\vec{D})^p - \Lambda(\vec{O}-\vec{D})^p}{2}\right)^{1/p}$, where Λ is the operator to extract L_p norm from sketch.

- $L_{1+\alpha}$ and $L_{1-\alpha}$ norm estimations of OD flows give us entropy estimation of OD flows.

Estimating L_p Norm of OD Flows

$$\begin{aligned}
\Lambda(\vec{O})^p &\approx |a_1|^p + \ldots + |a_k|^p + |b_1|^p + \ldots + |b_l|^p \\
\Lambda(\vec{D})^p &\approx |a_1|^p + \ldots + |a_k|^p + |c_1|^p + \ldots + |c_m|^p \\
\Lambda(\vec{O} - \vec{D})^p &\approx |b_1|^p + \ldots + |b_l|^p + |-c_1|^p + \ldots + |-c_m|^p \\
&= |b_1|^p + \ldots + |b_l|^p + |c_1|^p + \ldots + |c_m|^p
\end{aligned}$$

Hence,

$$\frac{\Lambda(\vec{O})^p + \Lambda(\vec{D})^p - \Lambda(\vec{O} - \vec{D})^p}{2} \approx |a_1|^p + \ldots + |a_k|^p.$$

Estimating L_1 Norm of OD Flows (Traffic Matrix)

- We can apply IMP to L_1 sketches
- Or we can utilize $L_{1+\alpha}$ and $L_{1-\alpha}$ norm estimations to avoid the overhead of L_1 sketch.

$$\begin{aligned} x &\approx \frac{1}{2}(x^{1+\alpha} + x^{1-\alpha}) \\ L_1(\vec{a}) = \sum a_i &\approx \frac{1}{2}\left(\sum a_i^{1+\alpha} + \sum a_i^{1-\alpha}\right) \end{aligned}$$

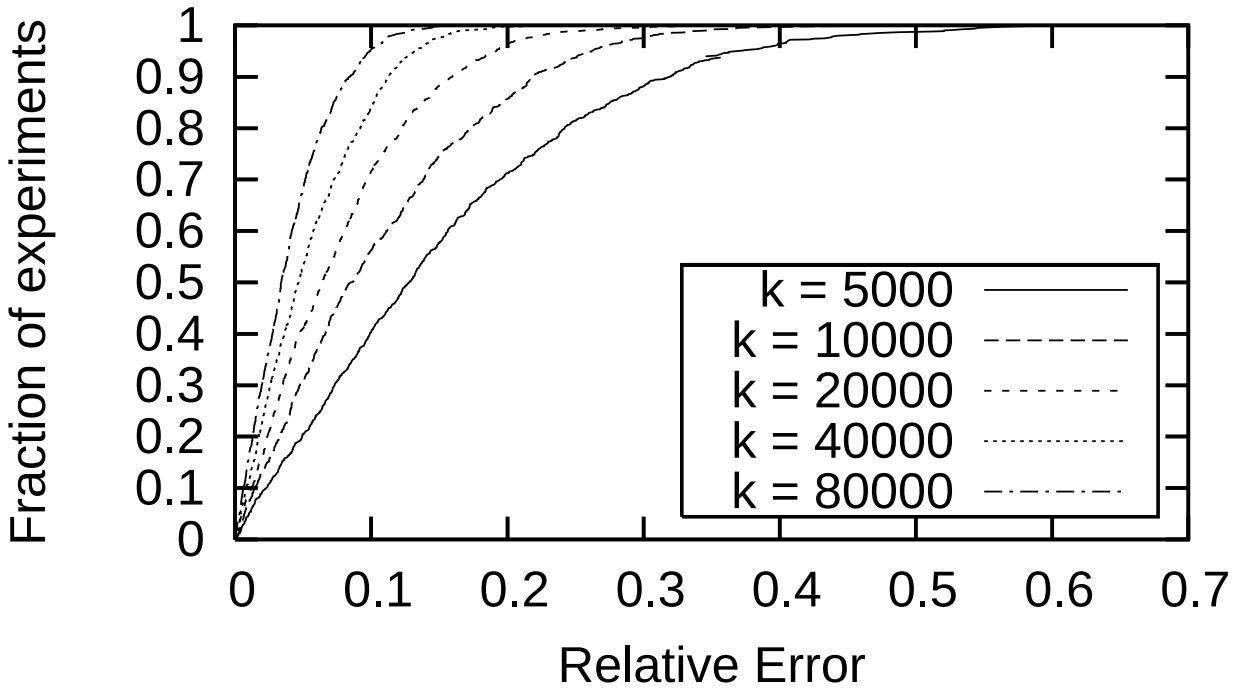
Modifications to Indyk's Sketch

- Note that for every packet we have to perform hundreds or thousands of updates per packet (infeasible at line speeds).
- Solution:
  - Hash packets into many (thousands of) buckets.
  - Apply Indyk's algorithm to each bucket with a small number (tens) of counters.
  - Combine the L_p norm of all buckets.
- The overall relative error is much smaller than the relative error of each bucket. (Think of Central Limit Theorem.)
- We also use large lookup tables for the stable distribution RV's.

Experiment Setup

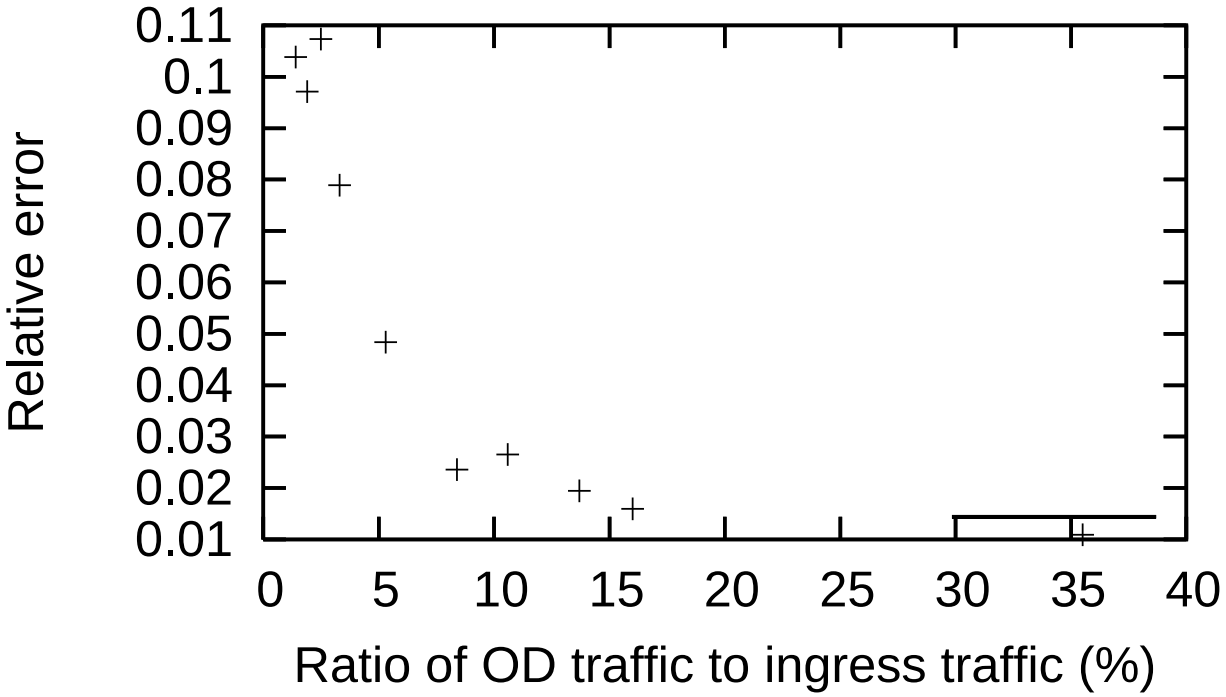
- We collect full packet trace on a 1Gbit/s ingress link of a large Tier-1 ISP.
- We use routing table dump to figure out the flows going to different egress link.
- At egress link we add dummy flows to simulate the full egress trace.
- We run our algorithm on the ingress trace and egress trace many times, and compare with the actual OD flow entropy.
- Typical parameters: $50,000$ buckets, 20 counters, $\alpha = 0.05$, elephant detection sampling rate 0.001.

Experiment Results



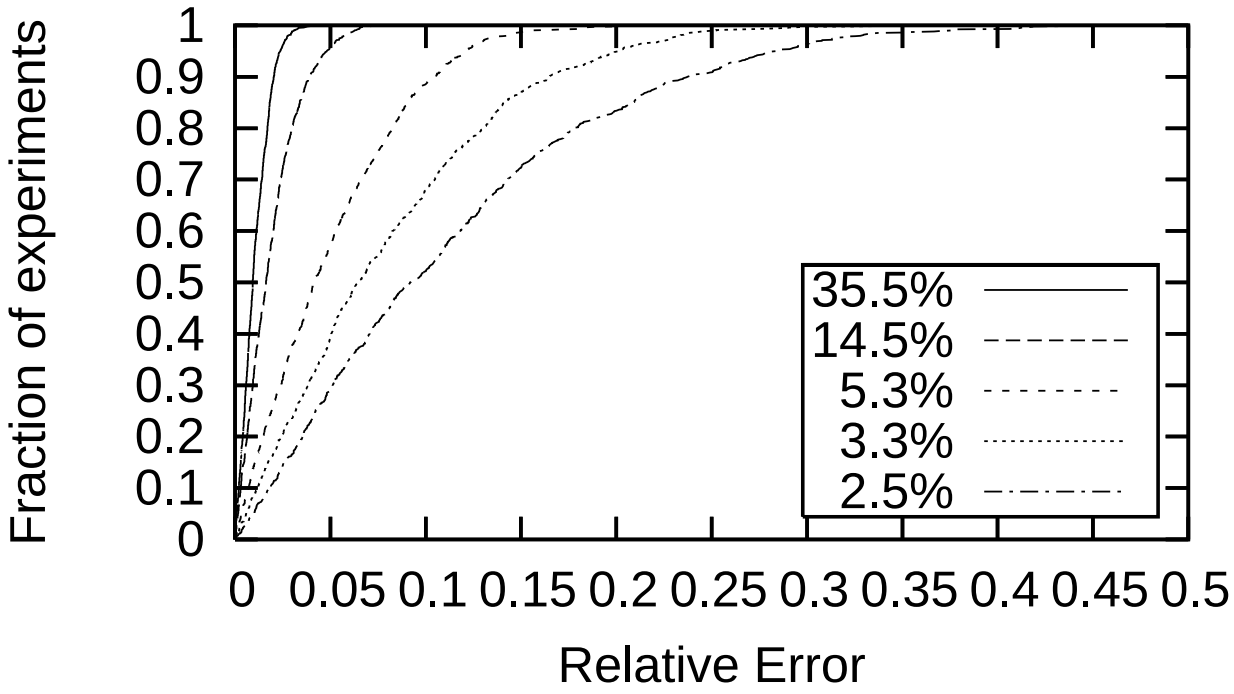
Varying numbers of buckets

Experiment Results



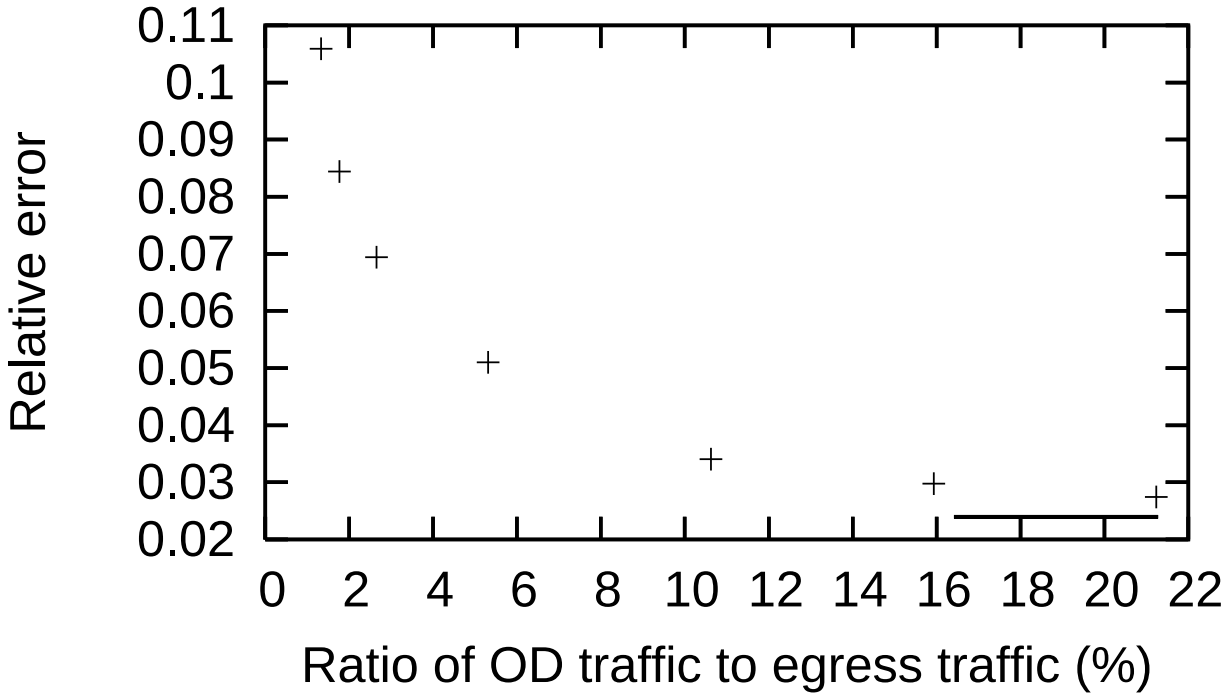
Varying fraction of traffic from ingress

Experiment Results



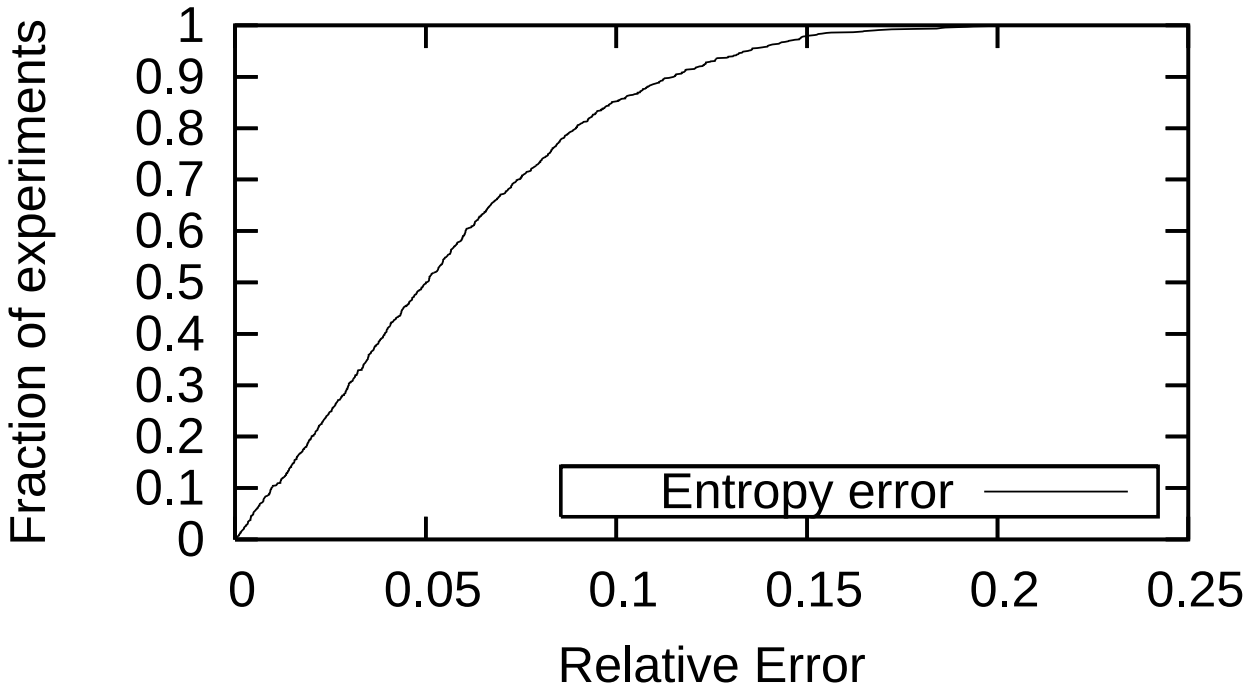
Varying fraction of traffic from ingress

Experiment Results



Varying fraction of traffic from egress

Experiment Results



Error distribution for actual entropy

Summary

- Approximation of the entropy norm using the L_p norms. (A new algorithm to estimate entropy of a single stream.)
- L_p norms for OD flows. (A new algorithm for traffic matrix since it is the L_1 norm of OD flows.)
- The first algorithm to estimate entropy of OD flows by combining the above two.
- Improvement to Indyk's sketch structure.

Thank you!

Questions?

References

[Indyk, 2006] Indyk, P. (2006). Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. ACM*, 53(3):307–323.

[Lakhina et al., 2005] Lakhina, A., Crovella, M., and Diot, C. (2005). Mining anomalies using traffic feature distributions. In *SIGCOMM*.