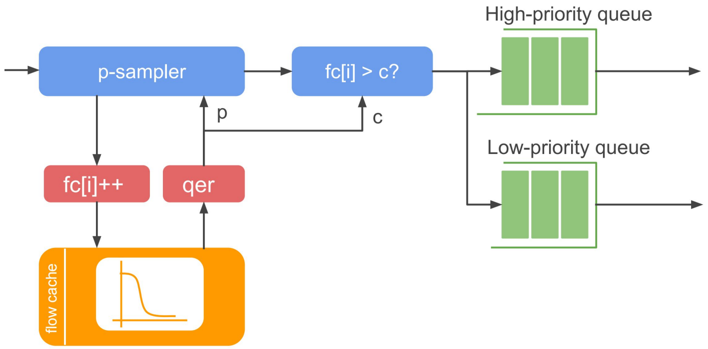
Fast Detection of Elephant Flows with Dirichlet-Categorical Inference

Aditya Gudibanda, Jordi Ros-Giralt, Alan Commike, Richard Lethin
{gudibanda, giralt, commike, lethin}@reservoir.com

Supercomputing Conference: Innovating the Network for Data-Intensive Science (INDIS) Workshop
November 11, 2018

Reservoir Labs
632 Broadway
Suite 803
New York, NY 10012



Roadmap

- Detection Under Partial Information: Mathematical Framework
- The Problem of Elephant Flow Detection
  - Related Work
- Introduction to Dirichlet-Categorical Inference
- Detection of Elephant-Flows via Dirichlet-Categorical Inference
  - Algorithm
  - Theory
  - Results
- Comparison with Static Sampling Methods
- Conclusion

Detection Under Partial Information: Mathematical Framework

- Identifying heavy hitter flows by inspecting all packets is not feasible.
- Packet sampling is introduced to make detection algorithms scalable.
- This leads to the problem of **flow reconstruction** under **partial information**.
- Two sources of uncertainty: (1) packet sampling and (2) inability to predict future flow performance.

Detection Under Partial Information: Mathematical Framework

- Information theoretic framework: what is the **minimum amount of information** (packets) we need to detect heavy hitter flows with a **user specified level of accuracy**?
- Theory of Elephant Flow Detection under Partial Information: IEEE ISNCC 2018 Ros-Giralt et al. Reservoir Labs.

Detection likelihood

Real-time flow cache

Oracle flow cache

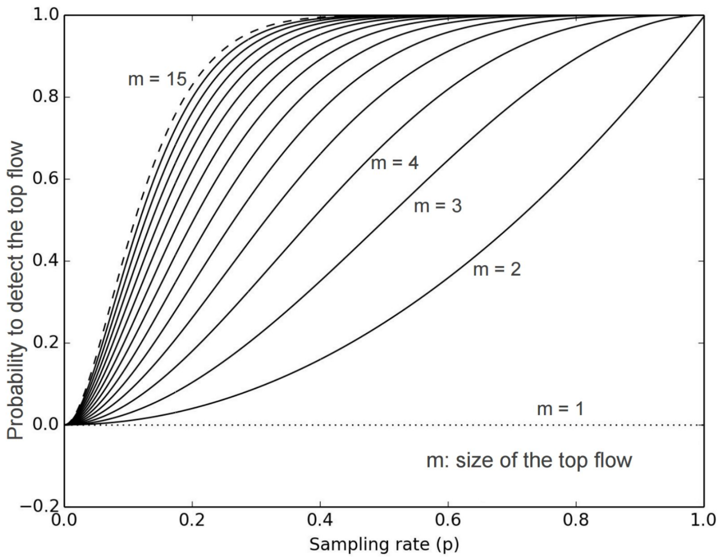
Oracle flow size

$$P(e_\alpha(t) = 0) = P(C_\alpha(t) = F_\alpha) = \sum_{\forall \mathbf{x}' \in Z(t)} \frac{\prod_{\forall i} \binom{\sigma_i}{x'_i}}{\binom{\sum_{\forall i} \sigma_i}{\sum_{\forall i} x_i(t)}}$$

Measured flow size

$$Z_\alpha(t) = \{\mathbf{x}' \in \mathbb{N}^{|F|} \mid \textstyle\sum_{\forall i} x'_i = \sum_{\forall i} x_i(t), \; \mathbf{x}' \leq_p \boldsymbol{\sigma}, \; x'_i > x'_j \; \forall i, j \text{ s.t. } i \leq \alpha, \; j > \alpha\}$$

Zero quantum error region

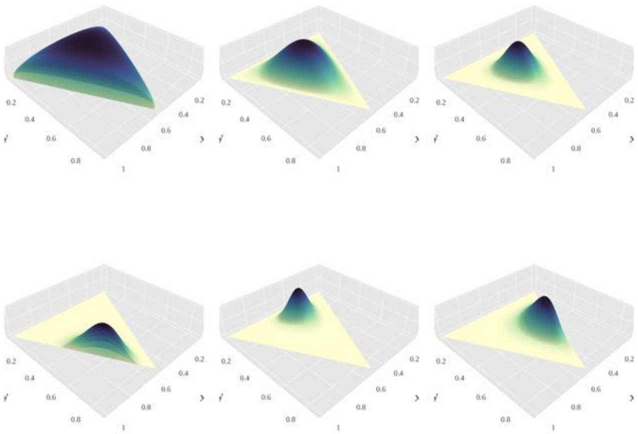


The Problem of Elephant Flow Detection: Related Work

- To the best of our knowledge, all existing solutions require threshold parameters:
  - Zhang 2010: Bayesian single sampling (high-rate flow traffic ratio threshold p^*)
  - Yi 2007: ElephantTrap algorithm (sampling rate p, top talker threshold L)
  - Psounis 2005: SIFT algorithm (sampling rate p).
  - Etc.
- Our contribution:
  - Threshold-free algorithm.
  - User-defined accuracy: first algorithm to accurately compute detection likelihood.
  - Mathematically proven convergence: $O(1/n)$
  - Scalable streaming/delta computations.

Introduction to Dirichlet-Categorical Inference

- Goal is to infer the distribution of flow sizes in a network.
- Normalize flow sizes to one.
- The flow distribution then turns into a categorical distribution.
- Our new goal is to efficiently compute the posterior of the categorical distribution induced by the flow sizes.



Examples of posterior distributions over the space of categorical distributions with three classes

- Dirichlet distribution is the conjugate prior of a categorical likelihood function.
- Posterior distribution is an analytic update of Dirichlet parameters:

Flow sizes

Dirichlet parameters

$$p(\boldsymbol{\sigma}) = Dir(\boldsymbol{\sigma};\boldsymbol{\alpha})$$

Prior

Count size

$$\boldsymbol{n} = [n_1, \ldots, n_\gamma]$$

Observations

$$p(\boldsymbol{\sigma}|\boldsymbol{n}) = Dir(\boldsymbol{\sigma};\boldsymbol{\alpha}+\boldsymbol{n})$$

Posterior

Analytic update

- Yields efficient inference procedure for categorical distributions.

Lemma 3: Lower bound detection likelihood with Bayesian inference. The detection likelihood of a data network at time t in the framework of Bayesian inference of the flow sizes σ_i satisfies:

Quantum error at time t

Objective: to efficiently compute these terms

$$P(e_\epsilon(t) = 0) = P(C_\epsilon(t) = F_\epsilon) \geq \prod_{i \in F_\epsilon, j \in F \setminus F_\epsilon} P(p_i(t) > p_j(t)) \quad (5)$$

- Marginals of Dirichlet distribution are Beta distributions:

Dirichlet distribution → $$Dir(\boldsymbol{x}; \boldsymbol{\alpha}) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

Beta distribution → $$B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}$$

Marginal of Dirichlet distribution → $$P(X_i = x_i) = B(\alpha_i, \alpha_0 - \alpha_i)$$

- Each flow’s marginal posterior distribution is Beta-distributed

- Each such term is an inequality between Beta distributions

$$\prod_{i \in F_\epsilon, j \in F \setminus F_\epsilon} P(p_i(t) > p_j(t))$$

$$P(X_i = x_i) = B(\alpha_i, \alpha_0 - \alpha_i)$$

$$p(\boldsymbol{\sigma}) = Dir(\boldsymbol{\sigma}; \boldsymbol{\alpha})$$

$$P(\sigma_i > \sigma_j) = \int_0^1 \left(\int_x^1 p_i(y)dy \right) p_j(x)dx$$

$$X = B(a, b) \qquad Y = B(c, d)$$

$$\boxed{g(a, b, c, d) = P(X > Y)}$$

- Online Updates given by [Cook2005]

Only first flow seen

Only second flow seen

Neither flow seen

$$g_{1,0}(a,b,c,d) = \frac{h(a,b,c,d)}{a} + \frac{h(a+1,b,c,d)}{d}$$

$$g_{0,1}(a,b,c,d) = -\frac{h(a,b,c,d)}{b} - \frac{h(a,b+1,c,d)}{c}$$

$$g_{0,0}(a,b,c,d) = -\frac{h(a,b,c,d)}{b} + \frac{h(a,b+1,c,d)}{d}$$

Update functions

where:

$$\boxed{h(a,b,c,d) = \frac{B(a+c,b+d)}{B(a,b)B(c,d)}}$$

Second flow: #seen and #unseen

First flow: #seen and #unseen

$$g(a+1,b,c,d+1) = g(a,b,c,d) + g_{1,0}(a,b,c,d)$$

$$g(a,b+1,c+1,d) = g(a,b,c,d) + g_{0,1}(a,b,c,d)$$

$$g(a,b+1,c,d+1) = g(a,b,c,d) + g_{0,0}(a,b,c,d)$$

process_sample()

- Initialize all pairwise probabilities to 0.5
- For every sample received:
    - Update the pairwise probabilities
    - Calculate the detection likelihoods
    - Identify the number of elephant flows and dynamically adjust the sampling rate

Algorithm 1: process_sample()

Initialize $p[\gamma][\gamma]$;
Initialize $a[\gamma]$;
Initialize $b[\gamma]$;
Initialize $d[\gamma - 1]$;
for $1 \leq i \leq \gamma$ **do**
  for $1 \leq j \leq \gamma$ **do**
    $p[i][j] = 0.5$;
begin
  for *sample* s_i **do**
    for $1 \leq i \leq \gamma$ **do**
      for $1 \leq j \leq \gamma$ **do**
        if s_i *is from flow* i *and not flow* j **then**
          $p[i][j] = p[i][j] + g_{1,0}(a[i], b[i], a[j], b[j])$;
        if s_i *is from flow* j *and not flow* i **then**
          $p[i][j] = p[i][j] + g_{0,1}(a[i], b[i], a[j], b[j])$;
        if s_i *is neither from flow* i *nor flow* j **then**
          $p[i][j] = p[i][j] + g_{0,0}(a[i], b[i], a[j], b[j])$;
      if s_i *is from flow* i **then**
        $a[i] = a[i] + 1$;
      else
        $b[i] = b[i] + 1$;
    d = calculate_detection_likelihoods();
    choose_ϵ_and_adjust_sampling_rate();

Dirichlet-Categorical Inference: Algorithm

calculate_detection_likelihoods()

- Compute a vector d which stores in position i the detection likelihood for the top i flows to be the elephant flows

choose_e_and_adjust_sampling_rate()

- Report best_e to be the largest e above the target detection likelihood
- If no e is above the detection likelihood, do not report a best_e and increase the sampling rate
- If an e is found (we have enough information) decrease the sampling rate

```
begin
    Initialize d[|F| − 1];
    for 1 ≤ i < |F| do
        if i = 1 then
            d[i] = 1
            for 2 ≤ j ≤ |F| do
                d[1] = d[1] · p[1][j];
        else
            d[i] = d[i − 1];
            for 1 ≤ j ≤ i − 1 do
                d[i] = d[i] / p[j][i];
            for i + 1 ≤ j ≤ |F| do
                d[i] = d[i] · p[i][j]
```

```
target_dl: target detection likelihood;
begin
    best_ε = −1;
    for 1 ≤ ε ≤ γ − 1 do
        if d[ε] > target_dl then
            best_ε = ε;
    if best_ε = −1 then
        increase_sampling_rate();
    else
        decrease_sampling_rate();
```

Dirichlet-Categorical Inference: Theory

Lemma 4: Computational Complexity. The total cost of the Dirichlet detection algorithm is $O(\gamma^2)$

Lemma 5: Order of Time Convergence. As the number of the packets observed n increases, under the assumption of expected asymptotic behavior, the probability of our algorithm misclassifying any flow goes to 0 at a rate of $O\left(\frac{1}{n}\right)$.

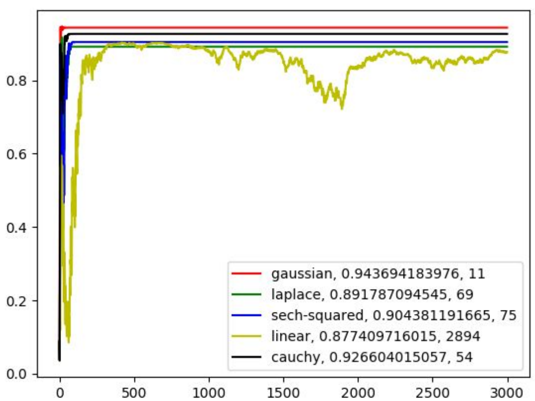
Dirichlet-Categorical Inference: Synthetic Experiments

Gaussian	Laplace	Sech-square	Cauchy	Linear
$\tau_i e^{-\frac{1}{2}i^2}$	$\tau_i e^{-i}$	$\frac{\tau_i e^{-i}}{(1+e^{-i})^2}$	$\frac{\tau_i}{1+i^2}$	$\tau_i(\gamma - i)$

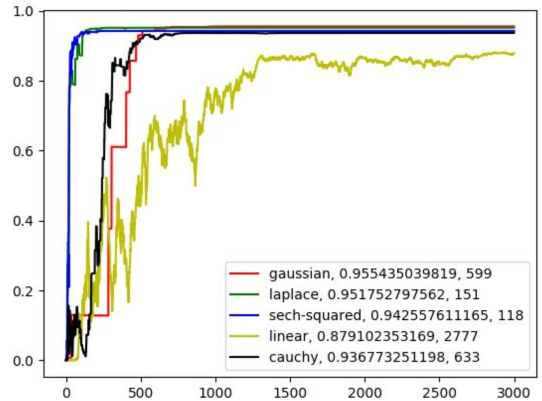
- Created synthetic flow distributions following Gaussian, Laplace, Sech-square, Cauchy, and Linear distributions
- Randomly generated flows from these distributions and measured detection likelihoods based on these samples
- Measured entropy of posterior distribution after each iteration
- Performed experiment for 10, 20, 50, 100, and 200 flows

Dirichlet-Categorical Inference: Synthetic Results

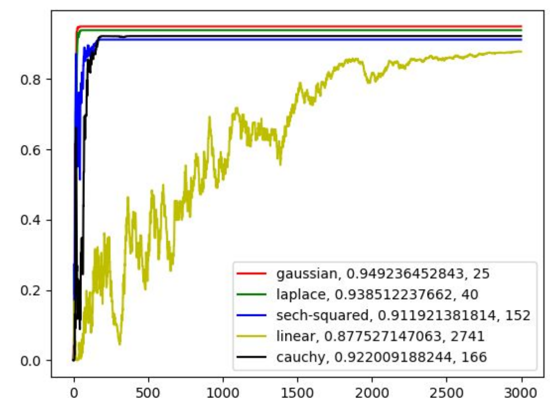
Gaussian: [242; 54; 5; 1; 1; 1; 1; 1; 1; 1]



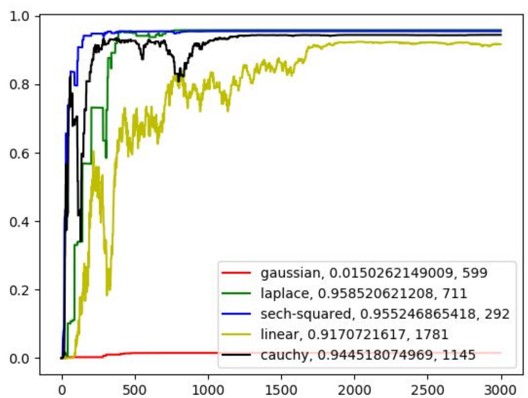
10 flows, detection likelihood for 1 flow



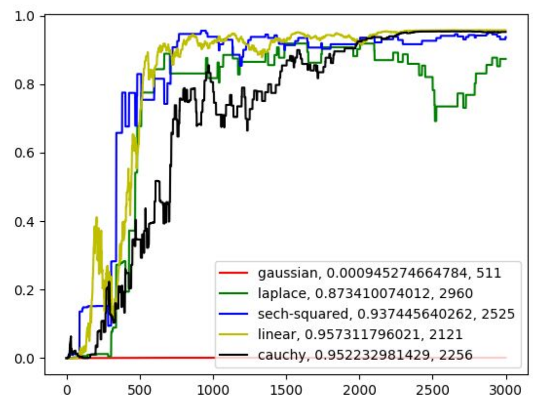
10 flows, detection likelihood for 2 flows



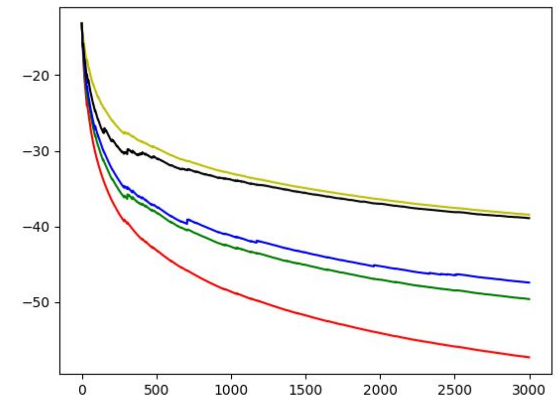
10 flows, detection likelihood for 3 flows



10 flows, detection likelihood for 4 flows



10 flows, detection likelihood for 5 flows

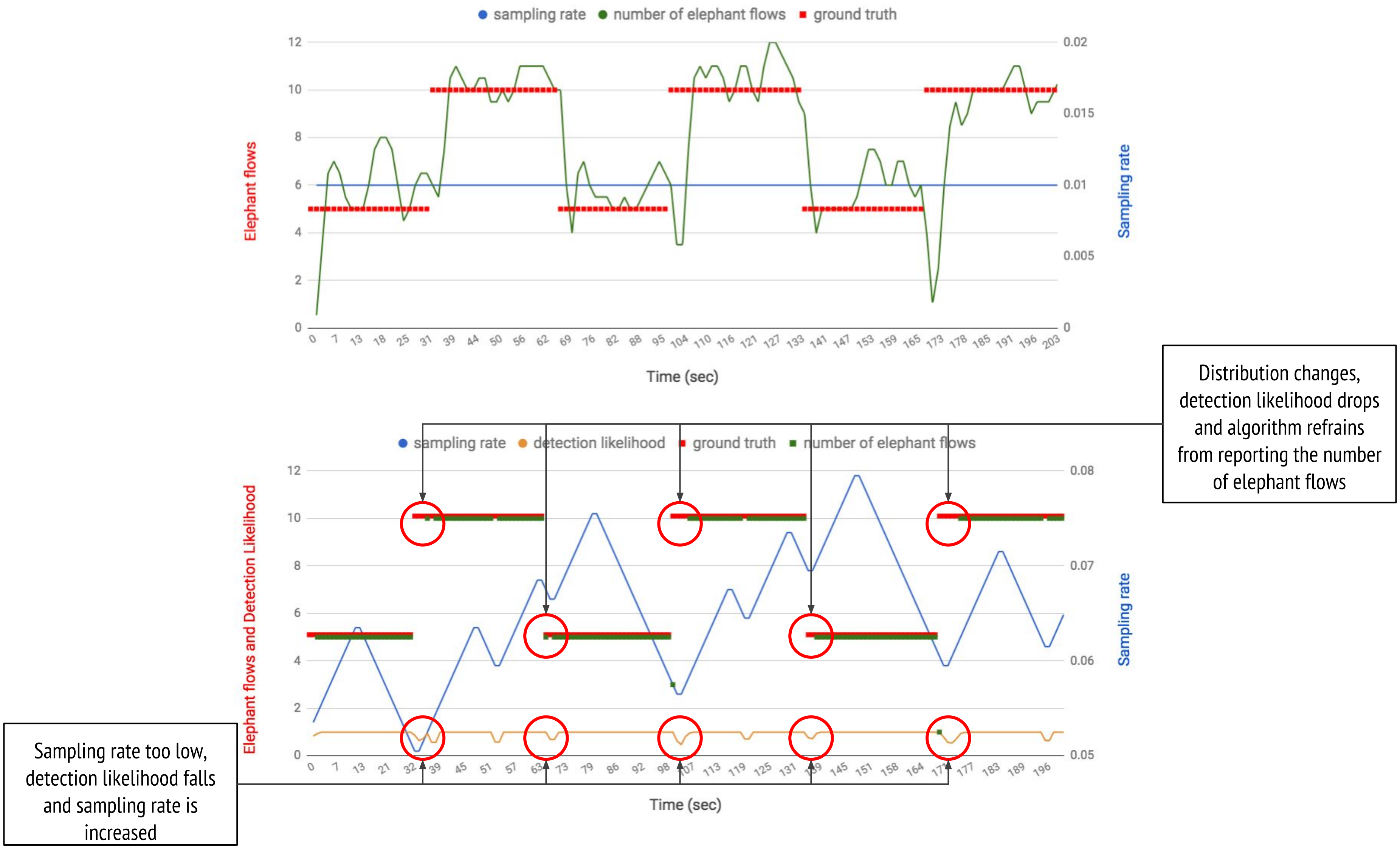


10 flows, Entropy curves for each distribution

Dirichlet-Categorical Inference: SDN Experiments

- Created an SDN testbed using Open vSwitch for network engineering and Linux KVM for virtualization of hosts on our network
- Simulated network had two nodes and one virtual switch between them
- Packets were sampled on the switch using sFlow
  - Modified sFlow to perform dynamic sampling
- Server hosting the network had 2 Xeon E5-2670 CPUs for a total of 32 cores and 64GB of RAM.
- Traffic alternated at 30s intervals between disjoint distributions with 5 elephant flows and 10 elephant flows
- Traffic on link was on average on the order of 100Mb/s
- Comparison between static sampling method with sampling rate of 0.01 [Psounis], and Dirichlet method with dynamic sampling

Dirichlet-Categorical Inference: SDN Results



Comparison with Static Sampling Rate

- Quantum error: the number of flows that an algorithm misclassifies divided by the true number of elephant flows
- Quantum error rate (QER): average quantum error over all time points:
  - Static sampling: 0.19844
  - Dirichlet method: 0.00893
  - Improvement by a factor of 22
- In addition, detection likelihood is a standalone module that can be placed on top of any existing static sampling rate method.
- Detection likelihood is complementary, not competitive.

Conclusion

- We have presented a new algorithm which identifies elephant flows within the Bayesian inference framework.
- Our algorithm is threshold-free, uses dynamic sampling, has proven convergence, and high classification accuracy.
- Dynamic sampling is based on detection likelihood, a quantification of uncertainty in our algorithm.
- Capable of automatically finding the cut-off sampling rate.
- Forthcoming work will integrate this algorithm in real datacenter and wide-area SDN environments.

Thank You

Reservoir Labs

632 Broadway
Suite 803
New York, NY 10012

812 SW Washington St.
Suite 1200
Portland, OR 97205

References

J. D. Cook, “Exact Calculation of Beta Inequalities,” UT MD Anderson Cancer Center Department of Biostatistics, Houston, Texas, Tech. Rep., 2005. [Online]. Available: https://www.johndcook.com/UTMDABTR-005-05.pdf

K. Psounis, A. Ghosh, B. Prabhakar, and G. Wang, “SIFT: A simple algorithm for tracking elephant flows, and taking advantage of power laws,” 2005.

L. Yi, W. Mei, B. Prabhakar, and F. Bonomi, “ElephantTrap: A low cost device for identifying large flows,” in Proceedings - 15th Annual IEEE Symposium on High-Performance Interconnects, HOT Interconnects, 2007, pp. 99–105.

Y. Zhang, B. Fang, and Y. Zhang, “Identifying high-rate flows based on Bayesian single sampling,” in ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings, vol. 1, 2010.

Additional Backup Slides for Q&A

Dirichlet-Categorical Inference: Implementation

- Flow counts are stored in a cache of size gamma
- Flow housekeeping
  - Flows which have not been seen within a certain period of time are removed from the cache
- Unique count optimization
  - Pairwise probability updates are same for flows with the same observation counts
  - Flow counts follow a power law distribution [Zhang2010]
  - Drastically reduces the amount of redundant computation, as flow counts are concentrated in a few small numbers
- Ghost Flow Protocol
  - Always keep a ghost flow in the pairwise probability table with a count size of 1
  - When a new flow enters, copy the ghost flow probabilities
  - Amortise the cost of adding a new flow

Dirichlet-Categorical Inference: QER Supplementary Material

- Measured the QER of Dirichlet Method over all time points, including time points when Dirichlet refrained from producing output
  - Static sampling: 0.19844
  - Dirichlet: 0.03072625698
  - Improvement by a factor of 6.5