Towards continual learning for networking algorithms

Keith Winstein
(with Francis Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, and Philip Levis)

Assistant Professor of Computer Science
Assistant Professor of Electrical Engineering (by courtesy)
Stanford University
High level takeaways

- Networking presents unique challenges for machine learning.

- We don’t know how to emulate the Internet very accurately.
  - **Training** algorithms in emulation: *disappointing* real-world results.
  - **Evaluating** algorithms in emulation: *not predictive* of real-world results.
  - Running in real life: **things change** over time, space, etc.

- **Continual learning** *in situ* mitigates some problems.
  - But hard to arrange...
  - ...only workable for some kinds of algorithms
  - ...and means a substantial shift in how we work.
Networking algorithms have many tunable parameters.

Natural setting for machine learning, but:

Networked systems present unique challenges for machine learning.
Example: Sprout (NSDI 2013) in publication

Sprout, NSDI 2013 (figure 7)
Sprout in real life in America

Stanford Pantheon result (July 31, 2018, T-Mobile in California),
https://pantheon.stanford.edu/result/3455/
Sprout in real life in India

Stanford Pantheon result (August 1, 2018, Airtel in New Delhi),
https://pantheon.stanford.edu/result/3474/
Example: Remy in simulation (SIGCOMM 2013)

Shown here: median results and 1 σ ellipses for eight endpoints contending for a 25 Mbps link, RTT = 150 ms, exponentially-distributed flow lengths and pause times.
Remy in emulation

Stanford Pantheon result (March 8, 2019, “Bottleneck buffer = BDP/3”),
https://pantheon.stanford.edu/result/5809/
Remy in real life

Stanford Pantheon result (March 9, 2019, HostDime India to AWS India),
https://pantheon.stanford.edu/result/5822/
Stanford Pantheon result (March 9, 2019, GCE Sydney to GCE Tokyo),
https://pantheon.stanford.edu/result/5840/
Example: Vivace (NSDI 2018) in publication

Vivace, NSDI 2018 (figure 7)
Vivace in real life

Stanford Pantheon result (March 9, 2019, AWS Brazil-HostDime Colombia),
https://pantheon.stanford.edu/result/5814/ (page 204 of PDF)
Figure 11: Comparing Pensieve with existing ABR algorithms in the wild. Results are for the $QoE_{lin}$ metric and were collected on the Verizon LTE cellular network, a public WiFi network, and the wide area network between Shanghai and Boston. Bars list averages and error bars span ± one standard deviation from the average.
Pensieve in reproduction

Figure 1: Comparison of Pensieve with other ABR algorithms across 10 tests on real world networks

Stanford CS244 student project,
“BBR converges toward a fair share of the bottleneck bandwidth whether competing with other BBR flows or with loss-based congestion control. [...] Unmanaged router buffers exceeding several BDPs, however, cause long-lived loss-based competitors to bloat the queue and grab more than their fair share.”

“Among the evaluated protocols, BBR yields the worst TCP friendliness. ... as we add more CUBIC flows, BBR becomes increasingly aggressive: it effectively treats all competing CUBIC flows as a single “bundle” .... Therefore, in practice, BBR can be very unfriendly when there are multiple competing CUBIC flows.”

Operators pile on BBR

Cubic vs BBR over a 12ms RTT 10G circuit

Coexistence with loss-based congestion control

- Example: 2 CUBIC, 2 BBR, 50M, 40ms, buffer = 1xBDP; start time (0, 2, 4, 6) secs

<table>
<thead>
<tr>
<th></th>
<th>bw</th>
<th>retrans</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUBIC 1</td>
<td>14.8 M</td>
<td>0.40%</td>
</tr>
<tr>
<td>CUBIC 2</td>
<td>12.4 M</td>
<td>0.12%</td>
</tr>
<tr>
<td>BBR 1</td>
<td>10.8 M</td>
<td>0.16%</td>
</tr>
<tr>
<td>BBR 2</td>
<td>10.9 M</td>
<td>0.21%</td>
</tr>
</tbody>
</table>

Improved fairness:
- BBR v1: 94% of bw
- BBR v2: 44% of bw

Proposal (2008): train a model to predict flu incidence from historical search engine queries. Then deploy the model to predict flu in advance of the government.
Detecting influenza epidemics using search engine query data

Jeremy Ginsberg¹, Matthew H. Mohebbi¹, Rajan S. Patel¹, Lynnette Brammer², Mark S. Smolinski¹ & Larry Brilliant¹

Seasonal influenza epidemics are a major public health concern, causing tens of millions of respiratory illnesses and 250,000 to 500,000 deaths worldwide each year¹. In addition to seasonal influenza, a new strain of influenza virus against which no previous immunity exists and that demonstrates human-to-human transmission could result in a pandemic with millions of fatalities². Early detection of disease activity, when followed by a rapid response, can reduce the impact of both seasonal and pandemic influenza³,⁴. One way to improve early detection is to monitor health-seeking behaviour in the form of queries to online search engines, which are submitted by millions of users around the world each day. Here we present a method of analysing large numbers of Google search queries to track influenza-like illness in a population. Because the relative frequency of certain queries is highly correlated with the percentage of physician visits in which a patient presents with influenza-like symptoms, we can accurately estimate the current level of acute influenza activity in real-time.

By aggregating historical logs of online web search queries submitted between 2003 and 2008, we computed a time series of weekly counts for 50 million of the most common search queries in the United States. Separate aggregate weekly counts were kept for every query in each state. No information about the identity of any user was retained. Each time series was normalized by dividing the count for each query in a particular week by the total number of online search queries submitted in that location during the week, resulting in a query fraction (Supplementary Fig. 1).

We sought to develop a simple model that estimates the probability that a random physician visit in a particular region is related to an ILI; this is equivalent to the percentage of ILI-related physician visits. A single explanatory variable was used: the probability that a random search query submitted from the same region is ILI-related, as determined by an automated method described below. We fit a linear model using the log-odds of an ILI physician visit and the log-odds of a flu-related search query as explanatory variables. We examined the ability of the model to predict the actual number of ILI visits in the test data and found that the agreement between predicted and observed values was good.
Accuracy figures

Index based on 45 queries (e.g. "pnumonia").

- Training data (2003–2007): $0.80 \leq r \leq 0.96$ (mean 0.90)
- Verification (2007–2008): $0.92 \leq r \leq 0.99$ (mean 0.97)

“We intend to update our model each year with the latest sentinel provider ILI data, obtaining a better fit and adjusting as online health-seeking behaviour evolves over time.”
Performance in the first year

Outpatient visits for influenza-like illness

Google

CDC

Keith Winstein
keithw@mit.edu
MIT CSAIL

The 2012–2013 Divergence of Google Flu Trends

- Training data (2003–2007): Mean correlation 0.90
- Verification (2007–2008): Mean correlation 0.97
- Actual (March–August 2009): Mean correlation 0.29!

Model retrained in September 2009, now 160 queries. “We will continue to perform annual updates of Flu Trends models to account for additional changes in behavior, should they occur.”
Google Flu Trends plot as of today

(http://www.google.org/flutrends/about/how.html)
Most of plot is training data
Large divergence (3.7×) in New England (HHS region 1)

Keith Winstein keithw@mit.edu MIT CSAIL
The 2012–2013 Divergence of Google Flu Trends
Substantial divergence (+72%) in France

Keith Winstein
keithw@mit.edu
MIT CSAIL

The 2012–2013 Divergence of Google Flu Trends
People turn to Twitter to see what’s trending in the news. But LSA Economics Professor Matthew Shapiro has found a new way to harvest employment information from tweets and hashtags faster and more accurately than the government’s official reports.

LSA Economics Professor Matthew Shapiro is one of the rare people who can claim to be getting work done when looking at social media. He’s not following sports scores or breaking news; he’s tracking the nation’s labor market.

“When we started,” explains Shapiro, “we had no idea if we could track job loss with tweets, but over a two-year period, we’ve seen the social media index perform quite well. In 2011-2012, for example, our index leveled off just like the official numbers. We captured big job-loss fluctuations around Hurricane Sandy, and around the government shutdown in October 2013.”

There were times when Shapiro’s numbers matched the reports, and there were times when they didn’t. When they differed, Shapiro’s numbers were more accurate than the government’s.

When the state of California got new computers, for example, there were delays in processing unemployment claims. Government data reflected the slowdown in processing applications, but social media captured a more accurate picture of what was happening in the labor market.

“Our series was stable,” says Shapiro, the Lawrence R. Klein Collegiate Professor of Economics, “so our numbers were, in some ways, a better indicator.”

By the Numbers

Many important economic indicators, including the Bureau of Labor Statistics unemployment rate and the University of Michigan

by Susan Hutton
Predicting unemployment claims from Twitter (at time of publication)
Predicting unemployment claims from Twitter (post-publication)
Microsoft scientists have demonstrated that by analyzing large samples of search engine queries they may in some cases be able to identify internet users who are suffering from pancreatic cancer, even before they have received a diagnosis of the disease.

The scientists said they hoped their work could lead to early detection of cancer. Their study was published on Tuesday in The Journal of Oncology Practice[...]

The researchers focused on searches conducted on Bing, Microsoft’s search engine, that indicated someone had been diagnosed with pancreatic cancer. From there, they worked backward, looking for earlier queries that could have shown that the Bing user was experiencing symptoms before the diagnosis. Those early searches, they believe, can be warning flags.

While five-year survival rates for pancreatic cancer are extremely low, early detection of the disease can prolong life in a very small percentage of cases. The study suggests that early screening can increase the five-year survival rate of pancreatic patients to 5 to 7 percent, from just 3 percent.
SpamAssassin (spam filtering engine):

- Anybody can propose a spam-filtering algorithm.
- Central party **learns** best weights based on predictive power of each algorithm.
- Weights are **then deployed** in the field.
2007: a rule is added

Rule: “Does the year match 200x?”

Catches a lot of spam.

Zero false positives in training! ... but what happens in 2010?
Yesterday...
Observation

Many apparent ML advances in networking decay with time, or with move from simulator/emulator to deployment.

Is the Internet a uniquely challenging setting for ML?
Unique challenges for ML in networking

- **Model mismatch**: training environment (simulator, emulator, testbed) does not match deployment environment (Internet).

- **Dataset shift**: change in network/userbase/workload over time

- Each participant has only *partial information*

- Failure may not be evident to any individual

- Greedy behavior may be locally rewarded, but globally bad

- *Adversaries, heavy tails, ...*
High level takeaways

▶ Networking presents unique challenges for machine learning.

▶ We don’t know how to emulate the Internet very accurately.
  ▶ Training algorithms in emulation: disappointing real-world results.
  ▶ Evaluating algorithms in emulation: not predictive of real-world results.
  ▶ Running in real life: things change over time, space, etc.

▶ Continual learning in situ mitigates some problems.
  ▶ But hard to arrange...
  ▶ ...only workable for some kinds of algorithms
  ▶ ...and means a substantial shift in how we work.
We don’t know how to emulate the Internet very accurately.

Possible solution: **continual learning** in place

- “Evaluate-first, idea-second” approach: deploy *something* and start collecting real metrics on real traffic
- Retrain frequently
- Compare against held-back “sane baseline”
Puffer: continual learning for Internet video streaming

- Live TV-streaming website (https://puffer.stanford.edu)
- Approved by Stanford lawyers, opened to public December 2018
- Streaming about 1,000 hours of video per day
- Randomizes sessions to different algorithms

**Goal:** realistic testbed and learning environment for research in:
- congestion control
- throughput prediction, and
- adaptive bitrate (ABR)
Algorithms that affect video streaming

- **Congestion control**: when to send each packet

- **Throughput prediction**: how fast can server send in near future?

- **Adaptive bitrate (ABR)**: what version of each upcoming “chunk” to send
Stanford University Launches a Streaming TV Service (for Science)

A research team at Stanford University launched the Puffer streaming service in a bid to improve video streaming algorithms. While live, a computer will be taught how to design new algorithms to reduce glitches and stalls while improving image quality.

By Matthew Humphries  January 18, 2019 10:13AM EST

Usually when a new, free service appears on the internet it is accompanied by some method of generating revenue to keep it running, which usually takes the form of ads. However, Stanford University just launched a free streaming TV service without ads.

As The Streamable reports, the new service is called Puffer and Stanford University
Stream live TV online | No charge to watch

Watch live U.S. TV channels (NBC, CBS, ABC, PBS, FOX, Univision) in your browser.
Stream live TV for free in your browser (NBC, CBS, ABC, PBS, FOX, and Univision). There's no charge, no extra ads, no credit card or personal information required. Not a scam, we promise.

puffer.stanford.edu

redcodefinal 19 points · 20 days ago

I thought this was bulls--- but holy f---, its from stanford.edu.

It works, no email required. First and only time I won't s--- post on a promoted ad!
The channel switching is LIGHTNING fast compared to other IPTV solutions I've seen, very impressive. But I still don't think this is legal, heard of Aereo?

It's part of a study... read the terms of service instead of blindly making comments.

You can't do something illegal just because it's part of a study. Stanford couldn't have a study on how murder affects communities that normally don't have murders and then just send someone in to a small peaceful town to murder at random.

Could make a good movie though.
Puffer algorithm

- Continual learning algorithm for video streaming
- Model-based RL: combines classical control and neural network
- Neural network predicts “how long would each chunk take?”
  - given internal TCP statistics (weakly cross-layer)
  - given size of proposed chunk
  - producing probability distribution (vs. point estimate)

- Classical controller: probabilistic model-predictive control
  - Maximize SSIM
  - Minimize stalls
Data Aggregation
Transmission Time
Predictor
MPC Controller
“OurTube”
Video Server
bitrate
selection
state
update
update
model
daily training
model-based control
Transmission Time Predictor
MPC Controller
Results

- Data from January 19–28, 2019
- 8,131 hours of video streamed
- 3,719 unique users
- 2 congestion-control schemes (BBR and Cubic), 7 ABR schemes
- Daily retraining for Puffer/BBR and Puffer/Cubic
## Results (BBR)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time stalled (lower is better)</th>
<th>Mean SSIM (higher is better)</th>
<th>SSIM variation (mean, lower is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Puffer</td>
<td>0.06%</td>
<td>17.03 dB</td>
<td>0.53 dB</td>
</tr>
<tr>
<td>Buffer-based</td>
<td>0.34%</td>
<td>16.58</td>
<td>0.79</td>
</tr>
<tr>
<td>MPC</td>
<td>0.81%</td>
<td>16.32</td>
<td>0.54</td>
</tr>
<tr>
<td>RobustMPC</td>
<td>0.29%</td>
<td>15.74</td>
<td>0.63</td>
</tr>
<tr>
<td>Pensieve</td>
<td>0.52%</td>
<td>16.19</td>
<td>0.88</td>
</tr>
<tr>
<td>Non-continual Puffer</td>
<td>0.31%</td>
<td>16.49</td>
<td>0.60</td>
</tr>
<tr>
<td>Emulation-trained Puffer</td>
<td>0.76%</td>
<td>15.26</td>
<td>1.03</td>
</tr>
</tbody>
</table>
Stalls vs. SSIM (BBR, < 6 Mbps only)
Results in emulation don’t match real world
People on Puffer watched 2× as long

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Stream duration (mean ± std. err.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(any congestion control)</td>
</tr>
<tr>
<td>Puffer</td>
<td>20.3 ± 1.3 minutes</td>
</tr>
<tr>
<td>Buffer-based</td>
<td>12.9 ± 0.8 minutes</td>
</tr>
<tr>
<td>MPC</td>
<td>9.5 ± 0.5 minutes</td>
</tr>
<tr>
<td>RobustMPC</td>
<td>10.0 ± 0.5 minutes</td>
</tr>
<tr>
<td>Pensieve</td>
<td>10.0 ± 0.5 minutes</td>
</tr>
<tr>
<td>Non-continual Puffer</td>
<td>12.5 ± 0.7 minutes</td>
</tr>
<tr>
<td>Emulation-trained Puffer</td>
<td>7.9 ± 1.3 minutes</td>
</tr>
</tbody>
</table>

Multivariate regression predicts each % of stall is associated with 6 minutes less watch time. Each dB of SSIM predicts 2 minutes more watch time ($R^2 = 0.36$).
## Contributors to transmission-time accuracy

<table>
<thead>
<tr>
<th>TTP ablation</th>
<th>Prediction error rate (lower is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No History + No cwnd inflight</td>
<td>19.8%</td>
</tr>
<tr>
<td>No History + No delivery rate</td>
<td>17.9%</td>
</tr>
<tr>
<td>No History + No RTT</td>
<td>24.0%</td>
</tr>
<tr>
<td>No History</td>
<td>17.1%</td>
</tr>
<tr>
<td><strong>Full TTP</strong></td>
<td><strong>13.7%</strong></td>
</tr>
<tr>
<td>Harmonic Mean</td>
<td>17.9%</td>
</tr>
</tbody>
</table>
Networking research if the Internet is *sui generis*

- Much-debated truism: “Data > algorithms” (even neural networks)

- In networking, “data” not enough. Need responsive *environment*.

- “Training set vs. testing set” is not an acceptable methodology.

- Much research to do in realistic network emulation so we can *train offline*

- Until then, need to *learn in situ* + may need to continually relearn over time

- Not so easy when:
  - Fairness is a concern: greedy behavior may be locally rewarded, but globally bad
  - Hard to attract enough users to get meaningful training and evaluation
  - Hard to compare notes across different deployments
Practical difficulties in deploying learned algorithms

- “We’ll give you a trace to learn or evaluate on.”
  - Hard to predict real performance from an emulator/simulator
  - We need to see more emulator calibration diagrams
- “Give us an algorithm, and we’ll evaluate it on .001% of users.”
  - Any sufficiently complex algorithm needs to be trained/tuned/debugged on real traffic
  - The real learning/developing usually starts after the first flow is served.
- “We invented a new congestion-control algorithm, tuned it for months, deployed it in an A/B test, and it reduced ABR playback stalls and page load times by n%.”
  - Do these gains come at the expense of competing flows?
  - Should we care?
Emulator calibration diagram (example)
High level takeaways

- Networking presents unique challenges for machine learning.
- We don’t know how to emulate the Internet very accurately.
- **Continual learning in situ** may mitigate some difficulties.
  - Win probably not from smarter algorithm, but in continual access to real data.
  - “Evaluate-first, idea-second”

- We are opening Puffer for other researchers to try ideas with daily feedback.
- But, Puffer is one small site with ≈ 1,000 hours/day of usage.
- Algorithm development would probably benefit from tighter collaboration between “people with candidate algorithms” and “people with users.”

Keith Winstein (keithw@cs.stanford.edu) + Francis Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, Philip Levis.

**Funded by:** Platform Lab, VMware, Huawei, Google, Amazon, Dropbox, Facebook, NSF, DARPA.