6.890: Learning-Augmented Algorithms

Piotr Indyk

MIT
Welcome to 6.890!

- Staff:
  - Lecturers:
    - Costis Daskalakis
    - Piotr Indyk
  - TAs: Nikhil Vyas, T.B.D.
  - “Consultant”: Yang Yuan

- Website on stellar

- Scribing doodle form at:
  [https://doodle.com/poll/kigg5ir5cph82rdn](https://doodle.com/poll/kigg5ir5cph82rdn)

- Meeting: 35-225
Learning-Augmented Algorithms

- "Classical" Algorithms (think 6.046)
  + Worst-case guarantees/analysis
  - Limited adaptivity to inputs

- Machine Learning Based Approaches
  + Stronger performance by adapting to inputs
  - No worst-case guarantees and/or analysis

- Learning-Augmented Algorithms
  + Adaptive
  + (Often) worst case guarantees/analysis
6.890: Learning-Augmented Algorithms

- This class is an experiment
  - Goal 1: cover relevant material (mostly from last 2 years)
  - Goal 2: develop new frameworks and applications
- Course style: lectures + project
- Project:
  - Proposal towards the end of March
  - Can be theoretical or practical
  - We will arrange cloud credits for the latter
  - Teams of size up to 2
  - Presentation (towards the end of the course)
- (Optional) scribing*
  - Scribing one lecture
  - Quality notes - they will likely be read
  - Can view this as an extra credit assignment
    (good scribe notes + decent project = A)
  - Plea: please sign up only if reasonably sure you will complete this class

*Preliminary plan
<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-Feb</td>
<td>Intro and overview. Bloom filters via learned oracles</td>
</tr>
<tr>
<td>7-Feb</td>
<td>Intro to ML: generalization etc</td>
</tr>
<tr>
<td>12-Feb</td>
<td>Intro to ML: neural networks</td>
</tr>
<tr>
<td>14-Feb</td>
<td>Learned oracles (Mike Mitzenmacher, Harvard)</td>
</tr>
<tr>
<td>19-Feb</td>
<td>Virtual Monday</td>
</tr>
<tr>
<td>21-Feb</td>
<td>Learned oracles and data structures (Tim Kraska, MIT)</td>
</tr>
<tr>
<td>26-Feb</td>
<td>Learned oracles and streaming algorithms (Piotr)</td>
</tr>
<tr>
<td>28-Feb</td>
<td>Learned oracles and online algorithms (Manish Purohit, Google)</td>
</tr>
<tr>
<td>5-Mar</td>
<td>Learned structures: compressed sensing (Ali Mousavi, Google)</td>
</tr>
<tr>
<td>7-Mar</td>
<td>Learned structures: compressed sensing (Erice Price, UT Austin)</td>
</tr>
<tr>
<td>12-Mar</td>
<td>Learned structures: similarity search (Ilya Razenshteyn, Microsoft)</td>
</tr>
<tr>
<td>14-Mar</td>
<td>Intro to ML - reinforcement learning</td>
</tr>
<tr>
<td>19-Mar</td>
<td>Neural network implementation: scheduling (Mohammad Alizadeh, MIT)</td>
</tr>
<tr>
<td>21-Mar</td>
<td>Neural network implementation</td>
</tr>
<tr>
<td>26-Mar</td>
<td>Vacation</td>
</tr>
<tr>
<td>Date</td>
<td>Event</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>28-Mar</td>
<td>Vacation</td>
</tr>
<tr>
<td>2-Apr</td>
<td>Topic lectures (mostly invited)</td>
</tr>
<tr>
<td>4-Apr</td>
<td>Topic lectures (mostly invited)</td>
</tr>
<tr>
<td>9-Apr</td>
<td>Topic lectures (mostly invited)</td>
</tr>
<tr>
<td>11-Apr</td>
<td>Topic lectures (mostly invited)</td>
</tr>
<tr>
<td>16-Apr</td>
<td>Patriot’s day</td>
</tr>
<tr>
<td>18-Apr</td>
<td>Topic lectures (mostly invited)</td>
</tr>
<tr>
<td>23-Apr</td>
<td>Topic lectures (mostly invited)</td>
</tr>
<tr>
<td>25-Apr</td>
<td>Topic lectures (mostly invited)</td>
</tr>
<tr>
<td>30-Apr</td>
<td>Topic lectures (mostly invited)</td>
</tr>
<tr>
<td>2-May</td>
<td>Topic lectures (mostly invited)</td>
</tr>
<tr>
<td>7-May</td>
<td>Student presentations</td>
</tr>
<tr>
<td>9-May</td>
<td>Student presentations</td>
</tr>
<tr>
<td>14-May</td>
<td>Student presentations</td>
</tr>
<tr>
<td>16-May</td>
<td>Student presentations</td>
</tr>
</tbody>
</table>
FAQ

• What is the difference between this class and an intro to ML class, say 6.036?
  – We already assume basic knowledge of ML. But we have a few lectures on ML essentials, including the next two lectures
  – We will also have a “Cloud Computing 101” session sometime in March

• What is the difference between this class and other classes that use machine learning for vision, natural language processing or other applications?
  – In our case, we learn a well-defined, fully computable function. We just want to evaluate it more efficiently (approximately).

• What is the difference between this class and “theory of deep learning”?

  “Algorithms for machine learning” vs. “Machine learning for algorithms”
Bloom filter

• A bloom filter is an approximate solution to the membership data structure problem
  – Given: a set $S=\{x_1,\ldots,x_n\}$ of elements
  – Goal: given $x$, determine whether $x$ is in $S$
• Bloom filter provides a low-space randomized solution to this problem
• Many applications…
Examples [edit]

- The servers of Akamai Technologies, a content delivery provider, use Bloom filters to prevent "one-hit-wonders" from being stored in its disk caches. One-hit-wonders are web objects requested by users just once, something that Akamai found applied to nearly three-quarters of their caching infrastructure. Using a Bloom filter to detect the second request for a web object and caching that object only on its second request prevents one-hit wonders from entering the disk cache, significantly reducing disk workload and increasing disk cache hit rates.[10]


- The Google Chrome web browser used to use a Bloom filter to identify malicious URLs. Any URL was first checked against a local Bloom filter, and only if the Bloom filter returned a positive result was a full check of the URL performed (and the user warned, if that too returned a positive result).[13][14]

- The Squid Web Proxy Cache uses Bloom filters for cache digests.[15]

- Bitcoin uses Bloom filters to speed up wallet synchronization.[16]

- The Venti archival storage system uses Bloom filters to detect previously stored data.[17]

- The SPIN model checker uses Bloom filters to track the reachable state space for large verification problems.[18]

- The Cascading analytics framework uses Bloom filters to speed up asymmetric joins, where one of the joined data sets is significantly larger than the other (often called Bloom join in the database literature).[19]

- The Exim mail transfer agent (MTA) uses Bloom filters in its rate-limit feature.[20]

- Medium uses Bloom filters to avoid recommending articles a user has previously read.[21]

- Ethereum uses Bloom filters for quickly finding logs on the Ethereum blockchain.
Bloom filter

• Components:
  – An array of $m$ bits, which represents a set $S=\{x_1,\ldots,x_n\}$ of elements
  – $k$ independent random hash functions $h_1\ldots h_k$, each hashing elements to $\{1\ldots m\}$
• Initialization: for each $x$ in $S$, $i$ in $\{1\ldots k\}$, set the bit $h_i(x)$ in the array to 1
• Check: an element $x$ is in $S$ if and only if $h_i(x)=1$ for all $i$ in $\{1\ldots k\}$

\[
\begin{array}{cccccc}
1 & 0 & 0 & 1 & 1 & 0 \\
\end{array}
\quad m=7 \\
\quad k=2
\]
• Can have **false positives**: \( x'' \) not in \( S \) but BF says YES
• But **no false negatives**: if BF says NO, then \( x \) not in \( S \)
• False positive probability for an element \( x'' \) not in \( S \):
  – For each \( i \) \( \Pr[h_i(x'')=1] \leq \frac{kn}{m} \)
  – \( \Pr[ \text{ for each } i', h_i(x'')=1 ] \leq (\frac{kn}{m})^k \)
  – E.g., for \( m=6n, k=3 \), it is less than \( 1/2^3=1/8 \)
How ML can help

• “ML oracle”: a trained function \( f(x) \) that provides some useful information to our algorithm.
  – Often, \( x \) is an element of the input.
• Question: what \( f \) would be useful for the membership data structure problem?
  – It would be nice to have
    \[
    f(x) = 1 \text{ iff } x \text{ in } S
    \]
• Problem:
  – Can’t assume we can learn this exactly
  – There will be* non-zero false positive rate (\( \text{FPR}_f \)) and false negative rate (\( \text{FNR}_f \)).
• Question: how to ensure there are no false negatives, as in the standard Bloom filter?

* FPR and FNR are defined somewhat differently than on the previous slides. More later.
Learned Bloom Filters
[Kraska, Beutel, Chi, Dean, Polyzotis, SIGMOD’18]

Oracle $f$

$\text{Output YES (x in S)}$

$\text{Bloom filter on S^{-}}$

$f(x)=1$\hspace{1cm}$f(x)=0$

$S^{-} = \{x \in S : f(x)=0\}$

No false negatives!
Learned Bloom Filters
[Kraska, Beutel, Chi, Dean, Polyzotis, SIGMOD’18]

- Probability model:
  - Element $x$ is selected according to some distribution $D$ over elements (both in $S$ and outside of it)
  - $\text{FNR}_f = \Pr_x[f(x)=0|x \in S]$; $\text{FPR}_f = \Pr_x[f(x)=1|x \not\in S]$.
  - $\text{FNR}$, $\text{FPR}$ for the whole system defined analogously (for $\Pr_{x,h_1,\ldots,h_k}$)

- We have
  \[ \text{FPR} = \text{FPR}_f + (1-\text{FPR}_f) \text{FPR}_{BF} \]

- Size of $S^-$: $\text{FNR}_f \times n$
Experiments

- Data set:
  - $S$: blacklisted phishing URLs
  - $D$: uniform over $S$ and some non-blacklisted URLs

- ML oracle $f$:
  - Character-level recurrent neural network. Specifically, 16-dimensional GRU with a 32-dimensional embedding for each character
  - Size = 0.0259MB

- FPR/size tradeoff

(from Kraska et al)
Comparison

**Learned BF**
- Better space-accuracy tradeoffs (assuming data has structure that can be discovered using ML)

**Standard BF**
- No training
- Fast evaluation
- "For each $x$" guarantee
Next lectures

• What does it mean to “learn” (generalization)

• What are the recurrent neural networks and how to train them (backpropagation)