Probabilistic Counting with Randomized Storage

Benjamin Van Durme and Ashwin Lall
Data Overload

• Lots of text (, images, audio, ...) is good

• But how to process it all?

• Approximate algorithms!

Make the best of what you’ve got
Data Overload

• Lots of text, images, audio, ...) is good

• But how to process it all?

• Approximate algorithms!

More data equals better results

Make the best of what you’ve got
Data Overload

• Lots of text (text, images, audio, ...) is good

• But how to process it all?

• Approximate algorithms!

More data equals better results

Buy/rent a data center?

Make the best of what you’ve got
Bulky Data
Bulky Data in Small Space

1980 ... 1985 ... 1990 ... 1995 ... 2000 ... 2005
Bulky Data in Small Space Online?

1980 ... 2000 ... ...

The New York Times

The New York Times

The New York Times
Outline

• Storing Static Counts
• Counting Online
• Experiments
• Additional Comments
Outline

• Storing Static Counts
• Counting Online
• Experiments
• Additional Comments
Bloom Filters [Bloom ’70]

- Records set membership.
- No false negatives.
- Some false positives.
- Think hashtables, where you throw away the key.
Bloom Filters [Bloom '70]

• Records set membership.
• No false negatives.
• Some false positives.
• Think hashtables, where you throw away the key.

Insert(x)
Bloom Filters [Bloom ’70]

• Records set membership.
• No false negatives.
• Some false positives.
• Think hashtables, where you throw away the key.

Insert(x)
Bloom Filters [Bloom ’70]

• Records set membership.

• No false negatives.

• Some false positives.

• Think hashtables, where you throw away the key.
Bloom Filters [Bloom ’70]

- Records set membership.
- No false negatives.
- Some false positives.
- Think hash tables, where you throw away the key.
Bloom Filters [Bloom ’70]

- Records set membership.
- No false negatives.
- Some false positives.
- Think hashtables, where you throw away the key.
Bloom Filters [Bloom ’70]

- Records set membership.
- No false negatives.
- Some false positives.
- Think hash tables, where you throw away the key.

Lookup(y)
Bloom Filters ...

- Bloom filters are nice when you can tolerate small false positives.
- And your x’s are large.
- For example, Language Modeling.
Motivation: n-grams for MT

... the dog
dog barked
barked at
...
Motivation: n-grams for MT

... the dog 97
dog barked 42
barked at 58
...
Motivation: n-grams for MT

... the dog 97
dog barked 42
barked at 58
...

The cat barked ...

The dog barked ...

Dog barked ...

Thursday, July 16, 2009
Motivation: n-grams for MT

... the dog 97
dog barked 42
barked at 58
...

The cat barked ...

The dog barked ...

Dog barked ...

 VAN DURME & LALL
Motivation: n-grams for MT

the dog 97
dog barked 42
barked at 58

... ?

The cat barked ...

The dog barked ...

Dog barked ...

Thursday, July 16, 2009
We quantise raw frequencies, \( c(x) \), using a logarithmic codebook as follows,

\[
qc(x) = 1 + \lfloor \log_b c(x) \rfloor.
\]
Storing Counts

- Multiple layers of Bloom filters.
- Store exponent, in unary.

\[ c(x) \approx b^{q_c(x)-1} \]
Outline

• Storing Static Counts
• Counting Online
• Experiments
• Additional Comments
The Spectral Bloom Filter (SBF) replaces the bit vector \( V \) with a vector of \( m \) counters, \( C \).
Spectral Bloom Filter [Cohen & Matias ’03]
Spectral Bloom Filter [Cohen & Matias ’03]

Insert(x)
Spectral Bloom Filter [Cohen & Matias ’03]
Spectral Bloom Filter [Cohen & Matias ’03]

Insert(x)

3 3 3
Spectral Bloom Filter [Cohen & Matias ’03]
Spectral Bloom Filter [Cohen & Matias ’03]

Insert(y)

5 5 4 1
Spectral Bloom Filter [Cohen & Matias ’03]

Lookup(x)

5
5
4
1
Spectral Bloom Filter [Cohen & Matias ’03]

\[ \text{Lookup}(x) \]

\[
\begin{array}{ccc}
5 & 5 & 4 & 1
\end{array}
\]
Collect Counts Online

- Count in log-scale, to save space.
Collect Counts Online

- Count in log-scale, to save space.
- Robert Morris (1978) gave us a way to do this.
Morris Bloom Counter

- Spectral Bloom Filter,
- but with Morris style updating.

Lookup(x)
Morris Bloom Counter

- Spectral Bloom Filter,
- but with Morris style updating.

\[
c(x) \approx \frac{b^f - 1}{b - 1}
\]
Morris Bloom Counter

- Same amount of space as Spectral Bloom Filter,
Morris Bloom Counter

- Same amount of space as Spectral Bloom Filter,
- gives exponentially larger max-count,
Morris Bloom Counter

- Same amount of space as Spectral Bloom Filter,
- gives exponentially larger max-count,
- but false positives can therefore have higher relative error.
Reduce False Positive Rate

- Morris Bloom Counter,
Reduce False Positive Rate

- Morris Bloom Counter,
- split into layers,
Reduce False Positive Rate

- Morris Bloom Counter,
- split into layers,
- with different hash functions per layer.

Insert(x)
Reduce False Positive Rate

- Morris Bloom Counter,
- split into layers,
- with different hash functions per layer.
Reduce False Positive Rate

- Morris Bloom Counter,
- split into layers,
- with different hash functions per layer.
Reduce False Positive Rate

- Morris Bloom Counter,
- split into layers,
- with different hash functions per layer.
Talbot Osborne Morris Bloom (TOMB) Counter

- Combination of Morris Bloom Counter with Talbot Osborne count storage.
- Stay tuned for related work by Talbot.
Tradeoff

• Trade number of layers for expressivity.

\[ M = \sum_{i} 2^{h_i} - 1 \]
Tradeoff

- Trade number of layers for expressivity.

\[ M = \sum_{i} 2^{h_i} - 1 \]

\[ h = 4 \]

0, 1, 2, ..., 14, 15
Tradeoff

- Trade number of layers for expressivity.

\[ M = \sum_{i} 2^{h_i} - 1 \]

0, 1, 2, 3, 4, 5, 6
Tradeoff

- Trade number of layers for expressivity.

\[
M = \sum_i 2^{h_i} - 1
\]
Tradeoff

- Trade number of layers for expressivity.

\[ h_1 = 1, \quad h_2 = 3 \]

\[ M = \sum_i 2^{h_i} - 1 \]

\[ 1, 2, 3, \ldots, 7, 8 \]
“Layers”

• Layers are a useful visualization.

• In practice, consecutive layers of equal height are implemented as single vectors with sets of hash functions.
Outline

- Storing Static Counts
- Counting Online
- Experiments
- Additional Comments
Experiment: Count Accuracy

Count all trigrams in Gigaword, randomly query 1,000 values, compare to truth

100 MB

500 MB

IJCAI 2009

Van Durme & Lall
## Experiment: MT

Build counters with varying amounts of memory

<table>
<thead>
<tr>
<th>TRUE</th>
<th>260MB</th>
<th>100MB</th>
<th>50MB</th>
<th>25MB</th>
<th>No Lm</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.75</td>
<td>22.93</td>
<td>22.27</td>
<td>21.59</td>
<td>19.06</td>
<td>17.35</td>
</tr>
<tr>
<td>-</td>
<td>22.88</td>
<td>21.92</td>
<td>20.52</td>
<td>18.91</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>22.34</td>
<td>21.82</td>
<td>20.37</td>
<td>18.69</td>
<td>-</td>
</tr>
</tbody>
</table>

(based on system of Post & Gildea ’08)
Experiment: MT ...

<table>
<thead>
<tr>
<th>True</th>
<th>260MB</th>
<th>100MB</th>
<th>50MB</th>
<th>25MB</th>
<th>No Lm</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.75</td>
<td>22.93</td>
<td>22.27</td>
<td>21.59</td>
<td>19.06</td>
<td>17.35</td>
</tr>
<tr>
<td>-</td>
<td>22.88</td>
<td>21.92</td>
<td>20.52</td>
<td>18.91</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>22.34</td>
<td>21.82</td>
<td>20.37</td>
<td>18.69</td>
<td>-</td>
</tr>
</tbody>
</table>

Three runs per counter size
Experiment: MT ...
Outline

• Storing Static Counts

• Counting Online

• Experiments

• Additional Comments
Related

- Applies method of Manku and Motwani ’02.
- Track most frequent elements in stream.
- Rare elements discarded.
- Strong guarantee on counts for top elements.

Streaming for large scale NLP: Language Modeling

Amit Goyal, Hal Daumé III, and Suresh Venkatasubramanian
University of Utah, School of Computing

NAACL 2009
Data that is not text

- Not just for Comp. Ling.
- E.g., count n-grams over "vocabularies" based on SIFT features.
Humans

- People store large amounts of information in their heads,
- and they do it online.
- Space efficient online counting provides additional area for interfacing with Cog. Sci. community.
Acknowledgements

• Ashwin Lall (co-author)
Acknowledgements

- Ashwin Lall (co-author)
- David Talbot, Miles Osborne
Acknowledgements

• Ashwin Lall (co-author)

• David Talbot, Miles Osborne

• Matt Post, Nick Morsillo, Dan Gildea
Questions?

www.cs.rochester.edu/~vandurme

www.cc.gatech.edu/~alall