Auto-completing search terms

Jessica Gronski

University of California Santa Cruz

June 15, 2007
Goal

Data

Implementation
  Assumptions

Results
  User 9033199
  Real Loss vs. Size
  Ordering

Conclusion and Future work
Problem: Search engines such as google and browsers such as Firefox and Safari all attempt to auto-complete terms as you type into a search bar.

Goal: Given history of words searched for and the first 3 letters of a new search word, try to learn the best mix of the most recently seen (MRU) search terms and the most frequently seen (MFU) search terms.
Problem: Search engines such as google and browsers such as Firefox and Safari all attempt to auto-complete terms as you type into a search bar.

Sub-Goal: I will allow up to five words in any given prediction. Would a three word prediction (or any other size smaller than five) be a subset of the five word prediction?
The data for this project comes from the "anonymized" search data released by AOL in "A Picture of Search".

Each line contains a userid, a search term and a time indicating when the query was received.\(^1\)

Weighted Median Algorithm is implemented to determine the mix of MRU and MFU in the predicted cache. This is an "ARC"ing style two lists so MFU is an approximation. For example for the prefix "www":

<table>
<thead>
<tr>
<th>MRU</th>
<th>MFU</th>
</tr>
</thead>
<tbody>
<tr>
<td>'www.yahoo.com'</td>
<td>'www.carlow.edu'</td>
</tr>
<tr>
<td>'www.treasurehouseinc.co'</td>
<td>'www.weddingclickers.com'</td>
</tr>
<tr>
<td>'www.treasurehouseinc.com'</td>
<td>'www.moveslikethis.com'</td>
</tr>
<tr>
<td>'www.reasurehouseinc.com'</td>
<td>'www.childsupport.state.pa.us'</td>
</tr>
<tr>
<td>'www.behaviormanager.carlow'</td>
<td></td>
</tr>
</tbody>
</table>
Weighted Median Algorithm is implemented to determine the mix of MRU and MFU in the predicted cache. This is an "ARC"ing style two lists so MFU is an approximation. For example for the prefix "www": Expert 0

**MRU:**
- www.yahoo.com
- www.treasurehouseinc.co
- www.treasurehouseinc.com
- www.treasurehouseinc.com
- www.behaviormanager.carlow

**MFU:**
- www.carlow.edu
- www.weddingclickers.com
- www.moveslikethis.com
- www.childsupport.state.pa.us
Weighted Median Algorithm is implemented to determine the mix of MRU and MFU in the predicted cache. This is an "ARC"ing style two lists so MFU is an approximation. For example for the prefix "www":

Expert 1

**MRU:**
- 'www.yahoo.com'
- 'www.treasurehouseinc.co'
- 'www.treasurehouseinc.com'
- 'www.reasurehouseinc.com'
- 'www.behaviormanager.carlow'

**MFU:**
- 'www.carlow.edu'
- 'www.weddingclickers.com'
- 'www.moveslikethis.com'
- 'www.childsupport.state.pa.us'
Weighted Median Algorithm is implemented to determine the mix of MRU and MFU in the predicted cache. This is an "ARC"ing style two lists so MFU is an approximation. For example for the prefix "www":

Expert 2

MRU:
'www.yahoo.com'
'www.treasurehouseinc.co'
'www.treasurehouseinc.com'
'www.behaviormanager.carlow'

MFU:
'www.carlow.edu'
'www.weddingclickers.com'
'www.moveslikethis.com'
'www.childsupport.state.pa.us'
Weighted Median Algorithm is implemented to determine the mix of MRU and MFU in the predicted cache. This is an ”ARC”ing style two lists so MFU is an approximation. For example for the prefix ”www”:

**MRU:**
- 'www.yahoo.com'
- 'www.treasurehouseinc.co'
- 'www.treasurehouseinc.com'
- 'www.treasurehouseinc.com'
- 'www.behaviormanage.carlow'

**MFU:**
- 'www.carlow.edu'
- 'www.weddingclickers.com'
- 'www.moveslikethis.com'
- 'www.childsupport.state.pa.us'
Weighted Median Algorithm is implemented to determine the mix of MRU and MFU in the predicted cache. This is an "ARC"ing style two lists so MFU is an approximation. For example for the prefix "www": Expert 4

MRU:
'www.yahoo.com'
'www.treasurehouseinc.co'
'www.treasurehouseinc.com'
'www.reasurehouseinc.com'
'www.behaviormanager.carlow'

MFU:
'www.carlow.edu'
'www.weddingclickers.com'
'www.moveslikethis.com'
'www.childsupport.state.pa.us'
Weighted Median Algorithm is implemented to determine the mix of MRU and MFU in the predicted cache. This is an "ARC"ing style two lists so MFU is an approximation. For example for the prefix "www": Expert 5

MRU:
'www.yahoo.com'
'www.treasurehouseinc.co'
'www.treasurehouseinc.com'
'www.reasurehouseinc.com'
'www.behaviormanager.carlow'

MFU:
'www.carlow.edu'
'www.weddingclickers.com'
'www.moveslikethis.com'
'www.childsupport.state.pa.us'
The loss is a unit step function returning zero or one depending if the intended search term $x$ is in the prediction set of the expert $i$.

$$L_i(x) = \begin{cases} 
0 & x \in \text{pred}(i) \\
1 & x \notin \text{pred}(i) 
\end{cases}$$

The weighted median algorithm uses exponential gradient algorithm for updating the weights.

$$w_i^t = \frac{w_i^{t-1} e^{-\eta L_i(x^t)}}{\text{normalizing factor}}$$
I auto-complete words not entire search terms and therefore am not a direct comparison with most search engine/ browsers auto-completions programs which complete the entire term.

I assume that the user has already typed three characters.

I run experiments that display up to five guesses completions of a search word.
Losses for Caches of size 1-5

- **1**
- **2**
- **3**
- **4**
- **5**
- **Ideal**

**Axes:**
- Vertical: Misses
- Horizontal: Iterations

**Legend:**
- Red: 1
- Green: 2
- Blue: 3
- Pink: 4
- Cyan: 5
- Orange: Ideal
User 9033199’s weights Cache size 1
User 9033199’s weights Cache size 2
User 9033199’s weights Cache size 3

9033199's weights with Cache size 3

iterations

weight

expert 0
1
2
3
User 9033199’s weights Cache size 4

9033199's weights with Cache size 4

- expert 0
- 1
- 2
- 3
- 4
User 9033199’s weights Cache size 5

9033199's weights with Cache size 5

- expert 0
- 1
- 2
- 3
- 4
- 5

iterations
weight

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
0 500 1000 1500 2000 2500 3000
Low

Percent of avoidable losses

Predicted loss - ideal loss / total search words

Total (non-unique) search words

Cache size: 1 +
2 x
3 *
4 o
5 ▲
High
Question: For $n < m$ is the median size $n$ cache a subset of the median size $m$ cache?
Question: For $n < m$ is the median size $n$ cache a subset of the median size $m$ cache? No

Intuition:

hit in $n \Rightarrow$ hit in $m$
miss in $m \Rightarrow$ miss in $n$

but

hit in $m \not\Rightarrow$ hit in $n$
miss in $n \not\Rightarrow$ miss in $m$

This asymmetry can be exploited to create a counterexample.
Algorithm for counterexample

Goal: $e_{n,n}$ and $e_{0,m}$ are the median experts.

1. Fill lists.
2. Repeat the following step until $w_{n,n} > 0.5$.
   Choose the final item in the list 1.
3. Repeat the following step until $w_{0,m} > 0.5$.
   Choose the final item in list 2.
What if the weights never recover to over 0.5?
Algorithm for counterexample

Goal: $e_{n,n}$ and $e_{0,m}$ are the median experts.

1. Fill lists.
2. Repeat the following step until $w_{n,n} > 0.5$.
   Choose the final item in the list 1.
3. Repeat the following step until $w_{0,m} > 0.5$.
   Choose the final item in list 2.

What if the weights never recover to over 0.5?
Observation

Suppose only expert $i$ wins after timestep $t$.
\[ \forall t \geq T. \exists! i. \text{Loss}(i, t) = 0 \land j \neq i \Rightarrow \text{Loss}(j, t) = 1. \]

\[
    w_{i}^{t+1} = \frac{w_{i}^{t}}{k \ast (\Sigma w_{j}^{t}) + w_{i}^{t}} = \frac{w_{i}^{t}}{k \ast (1 - w_{i}^{t}) + w_{i}^{t}} = \frac{w_{i}^{t}}{k + (1 - k)w_{i}^{t}}
\]

where $k = e^{-\eta}$. What is the value of $p$ s.t. $w_{i}^{t+p} > 0.50$?

Solve the recurrence for

\[ w_{i}^{t+1} = \frac{w_{i}^{t}}{k + (1 - k)w_{i}^{t}} \]
Observation

Suppose only expert $i$ wins after timestep $t$.

$\forall t \geq T. \exists ! i. \text{Loss}(i, t) = 0 \land j \neq i \Rightarrow \text{Loss}(j, t) = 1.$

\[
\begin{align*}
  w_{i}^{t+1} &= \frac{w_{i}^{t}}{k \ast (\Sigma w_{j}^{t}) + w_{i}^{t}} = \frac{w_{i}^{t}}{k \ast (1 - w_{i}^{t}) + w_{i}^{t}} = \frac{w_{i}^{t}}{k + (1 - k)w_{i}^{t}}
\end{align*}
\]

where $k = e^{-\eta}$. What is the value of $p$ s.t. $w_{i}^{t+p} > 0.50$?

Observe that for any threshold $y \neq 0$:

\[
\begin{align*}
  \frac{w_{i}^{t}}{k + (1 - k)w_{i}^{t}} &> y \\
  \frac{1}{y} \ast w_{i}^{t} &> k + (1 - k)w_{i}^{t} \\
  (k - 1 + \frac{1}{y}) \ast w_{i}^{t} &> k \\
  w_{i}^{t} &> \frac{k}{k - 1 + \frac{1}{y}}
\end{align*}
\]
So it takes $e_i$, $p$ steps to surpass 0.5 if $w_i^t > y$: 

<table>
<thead>
<tr>
<th>$p$</th>
<th>$y$</th>
<th>$w_i^t$</th>
<th>$p &gt; \log_k \frac{w_i^t}{1-w_i^t}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>$1 + \frac{1}{k^p}$</td>
<td>1 * $\ln k &lt; \ln \frac{w_i^t}{1-w_i^t}$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>$\frac{1}{k^p}$</td>
<td>$\frac{w_i^t}{1-w_i^t}$</td>
</tr>
<tr>
<td>2</td>
<td>$\frac{1}{1+\frac{1}{k^2}}$</td>
<td>$\frac{w_i^t}{1-w_i^t}$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$\frac{1}{1+\frac{1}{k^3}}$</td>
<td>$\frac{w_i^t}{1-w_i^t}$</td>
<td></td>
</tr>
</tbody>
</table>

Conclusion: No matter the starting weight of expert $i$ in $p$ steps the weight will surpass 0.5 if $\log_k \frac{w_i^t}{1-w_i^t} < p < \log_k \frac{w_i^t}{1-w_i^t} + 1$. 
The weighted median algorithm with underlying MRU and MFU lists seems to have sufficiently small "real" loss (prediction’s loss - ideal loss / search words) but actual loss (prediction’s loss / search words) is still quite large so that the user experience will not necessarily be smooth.

What if the user histories were longer?

There seems to be diminishing returns increasing the Cache size after 4 items.

Predicted caches of increasing sizes are not monotonically increasing sets.

The prefix size need not be of size 3 what if it were less? more?

I found the loss to be sufficiently small using these two underlying lists: MRU and MFU. We could add more experts but why?