…there seems something else in life besides time, something which may conveniently called “value”, something which is measured not by minutes or hours but by intensity, so that when we look at our past it does not stretch back evenly but piles up into a few notable pinnacles, and when we look at the future it seems sometimes a wall, sometimes a cloud, sometimes a sun, but never a chronological chart

- E. M. Foster
Bursty and Hierarchical Structure in Streams

John Kleinberg

Presented by:
Dave Kauchak
Department of Computer Science
University of California, San Diego
dkauchak@cs.ucsd.edu
Document organization

- **Topic**
  - Traditional clustering methods

- **Time**
  - Document streams
    - E-mail
    - News
    - Published literature
Document stream example
…there seems something else in life besides time, something which may conveniently called “value”, something which is measured not by minutes or hours but by intensity, so that when we look at our past it does not stretch back evenly but piles up into a few notable pinnacles, and when we look at the future it seems sometimes a wall, sometimes a cloud, sometimes a sun, but never a chronological chart

- E. M. Foster
Modeling Bursty Streams

- \( n+1 \) documents
  - For example, \( n+1 \) e-mails

- \( x_1, x_2, \ldots, x_n \) time gaps between messages
  - For example, differences in receive times for e-mails

- Generative model of gaps using:
  - Exponential density function
  - Two state automaton
  - Infinite state automaton
Exponential Density Function

- Model gaps using
  \[ f(x) = \alpha \cdot e^{-\alpha x} \]
  with parameter, \( \alpha \)
- Probability that the gap exceeds \( x \) is \( e^{-\alpha x} \)
- Expected value of a gap is \( 1/\alpha \)
Two State Model

- Want to distinguish between “low” and “high” densities

\[ f_0(x) = \alpha_0 \cdot e^{-\alpha_0 x} \]
\[ f_1(x) = \alpha_1 \cdot e^{-\alpha_1 x} \]

\[ \alpha_0 < \alpha_1 \]

with probability

\( p \) change states
Finding best state sequence

- Given: $n+1$ points generate $x_1, x_2, \ldots, x_n$ inter-arrival gaps
- Calculate most likely state sequence $q=q_0, q_1, q_2, \ldots q_n$
- Use Bayes rule

$$\arg \max_q p(q \mid x) = \frac{p(x \mid q)p(q)}{p(x)}$$
\[ p(q \mid x) = \frac{p(x \mid q) p(q)}{p(x)} \]

Prior:
\[
p(q) = \left( \prod_{t \neq t+1} p \right) \left( \prod_{t = t+1} 1-p \right) = p^b (1-p)^{n-b} = \left( \frac{p}{1-p} \right)^b (1-p)^n
\]

Conditional:
\[
p(x \mid q) = \prod_{t=1}^n f_t(x_t)
\]

\[
\arg \min_q \log p(q \mid x) = b \log \left( \frac{1-p}{p} \right) + \left( \sum_{t=1}^n - \log f_t(x_t) \right) - n \log(1-p) + \log(p(x))
\]
Minimize

\[
\arg \min_{q} \log p(q \mid x) = b \log \left( \frac{1 - p}{p} \right) + \left( \sum_{t=1}^{n} -\log f_t(x_t) \right) - n \log (1 - p) + \log p(x)
\]

favors small number of state transitions

favors sequences that conform to \( x \) values
Infinite State Model

- $n+1$ messages over time length $T$
- Even spacing of $\hat{g} = \frac{T}{n}$
- Consider more and more “bursty” or dense messages
- For each $q_i$, for $i = 0, 1, \ldots$
  
  $$f_i(x) = \alpha_i \cdot e^{-\alpha_i x} \quad \alpha_i = \frac{s^i}{\hat{g}}$$

  where $s > 1$ is a scaling parameter
- Models inter-arrival gaps that decrease monotonically
Graphical representation

\[ \tau(i, j) = \begin{cases} 
  i < j & (j - i) \lambda \ln n \\
  i \geq j & 0 
\end{cases} \]
Computing min-cost state sequence

- Minimize cost function similar to two state model
  \[ c(q \mid x) = \left( \sum_{t=0}^{n-1} \tau(t, t+1) \right) + \left( \sum_{t=1}^{n} -\ln f_t(x_t) \right) \]

- Difficult to do for an infinite state model

- Instead, show if \( q^* \) is optimal for a \( k \) state model then it is also optimal for an infinite state model where
  \[
  k = 1 + \log_s T + \log_s \delta(x)^{-1} \quad \delta(x) = \min_{i=1}^{n} x_i
  \]

- We can then use standard methods from graphical model theory to learn optimal state sequence
Proof

- Let $q^* = q^*_1, q^*_2, \ldots, q^*_n$ be an optimal state sequence of a $k$ state finite model
- Let $q = q_1, q_2, \ldots, q_n$ be an arbitrary sequence of an infinite state model
- Show $c(q^*|x) \leq c(q|x)$
- We’ll do this by generating a $q'$ for a $k$ state model and show $c(q'|x) \leq c(q|x)$
- Since $q^*$ is optimal then, $c(q^*|x) \leq c(q'|x) \leq c(q|x)$
**Proof continued** \[ c(q'|x) \leq c(q|x) \]

- If \( q \) does not contain an index greater than \( k-1 \) then this equality is true since \( q^* \) is optimal.
- Otherwise, consider the state sequence derived from \( q \), \( q' = q'_1, q'_2, \ldots, q'_n \) where \( t' = \min(t, k-1) \).
- The cost function contains two parts:
  \[
  c(q|x) = \left( \sum_{t=0}^{n-1} \tau(t, t+1) \right) + \left( \sum_{t=1}^{n} -\ln f_t(x_t) \right)
  \]
- By definition of \( \tau \):
  \[
  \sum_{t'=0}^{n-1} \tau(t', t'+1) \leq \sum_{t=0}^{n-1} \tau(t, t+1)
  \]
Proof continued

- For a given $x_t$, what state $(j)$ minimizes 
  \[ -\ln f_j(x_t) = \alpha_j x_t - \ln \alpha_j \]

- Concave up, with min at $\alpha = 1/x_t$

- So $\geq$ the min is achieved at one of 
  $\alpha_{j^*} \leq x_t^{-1} \leq \alpha_{j^*+1}$

- Also, for $j'' \geq j^* \geq j^*+1$, then 
  $-\ln f_{j''}(x_t) \geq -\ln f_j(x_t)$

- Since 
  \[ k = 1 + \log_s T + \log_s \delta(x)^{-1} \]
  \[ \alpha_{k-1} = \hat{g}s^{k-1} = \frac{n}{T} \cdot s^{k-1} \geq \frac{1}{T} \cdot s^{\log_s T + \log_s \delta(x)^{-1}} \]
  \[ = \frac{1}{T} \cdot \frac{T}{\delta(x)} = \frac{1}{\delta(x)} \]

- Finally, since $\delta(x) \leq x_t$ the index $k-1$ is at least as large as 
  the $j$ for which $-\ln f_j(x_t)$ is minimized our proof is complete
Result of proof

- Given this proof, calculating an optimal solution in the infinite model is the same as calculating the optimal solution for the $k$ state model.
- We can do this using the forward dynamic programming algorithm, or the Viterbi algorithm to calculate the optimal sequence.
Forward Dynamic Programming

Starting with $i = 1$ for each $x_i$, greedily pick best state based on values for all states for $x_{i-1}$
Extracting hierarchical structure

- Given and optimal state sequence:
  \[ q = q_1, q_2, \ldots, q_n \]
- Identify bursts of intensity \( j \), which are maximal intervals where \( q \) is in a state of \( j \) or higher
- This results in nested bursts of intensity, creating a hierarchy
Hierarchical example

optimal state sequence
0 1 2 3

bursts
0 1 2 3

time

tree representation
0 1 2 3
E-mail streams

- Collection from June 9, 1997 to August 23, 2001
- 34,344 e-mail messages
- Subsets collected by searching for messages with a particular string
  - “ITR”: related to NSF program
  - “prelim”: term for non-final tests
“ITR”
"ITR" - Hierarchy

- 10/28: letter of intent deadline (large proposals)
- 11/15: letter of intent deadline (large proposals)
- 1/2: pre-proposal deadline (large proposals)
- 2/14: full proposal deadline (small proposals)
- 4/17: full proposal deadline (large proposals)
- 7/11: unofficial notification (small proposal)
- 9/13: official announcement of awards

Minutes since 1/1/97

Message #
“ITR” - Tree

- Unofficial notification (7/11)
- Pre-proposal deadline (1/5)
- Letter of intent deadline (11/15)
- Unofficial notification (7/11)
3 courses

2 prelims per course
Interesting but…

- How is this useful?
  - Can identify bursty time periods
  - Hierarchical representation

- Can we identify most interesting words based on burstiness?
  - Enumerate bursts associated with each word
  - Compute a weight associated with the bursts
  - Rank words according to weight
Modeling Conference Papers

- Different than e-mail modeling: papers come in batches at consistent intervals
- Instead of inter-arrival time, interested in relevant and irrelevant documents
- Document is considered relevant if it contains a particular word $w$ that we are interested in
- Given $n$ batches of documents, model how many relevant documents are in a batch
Modeling paper: Binomial distribution

- Each batch contains $r_t$ relevant documents out of $n_t$
- We can model this using a binomial distribution defined by parameter $0 \leq p \leq 1$

$$p(i, n_t, r_t) = \binom{n_t}{r_t} p_i^{r_t} (1 - p_i)^{n_t - r_t}$$

- As before, we would like to model multiple states of “burstiness”

$$p_0 = \frac{\sum_{t=1}^{n} r_t}{\sum_{t=1}^{n} d_t} \quad \quad P_i = P_0 s^i$$
Paper modeling: A finite state model

- Since $0 \leq p \leq 1$ (and $s > 1$) then there will only be a finite number of states.
- We define the transition functions $\tau$ as before:

$$\tau(i, j) = \begin{cases} 
    i < j & (j - i)\lambda \ln n \\
    i \geq j & 0
\end{cases}$$

- As before, the minimum cost sequence can be calculated using Bayes rule and forward dynamic programming.
Most bursty topics

- Only interested in enumerating bursts of positive intensity so use a two state model
- Bursts of positive intensity are those intervals in which the state is $q_1$ and not $q_0$.
- For a positive intensity burst $[t_1, t_2]$ the weight is assigned as

$$ weight([t_1, t_2]) = \sum_{t=t_1}^{t_2} \ln p(1, r_t, d_t) - \ln p(0, r_t, d_t) $$

improvement in cost for using $q_1$ instead of $q_0$
3 data sets

- For all words $w$ in the data sets, relevant documents are those that contain the word $w$
- The weights are calculated for each $w$
- 3 data sets examined
  - Database conferences: SIGMOD, VLDB 1975-2001
  - State of the Union Speeches: 1790-2001
<table>
<thead>
<tr>
<th>Word</th>
<th>Interval of burst</th>
<th>Word</th>
<th>Interval of burst</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>1975 SIGMOD — 1979 SIGMOD</td>
<td>deductive</td>
<td>1985 VLDB — 1994 VLDB</td>
</tr>
<tr>
<td>application</td>
<td>1975 SIGMOD — 1982 SIGMOD</td>
<td>objects</td>
<td>1987 VLDB — 1992 SIGMOD</td>
</tr>
<tr>
<td>theory</td>
<td>1977 VLDB — 1984 SIGMOD</td>
<td>warehouse</td>
<td>1996 VLDB —</td>
</tr>
<tr>
<td>distributed</td>
<td>1977 VLDB — 1985 SIGMOD</td>
<td>similarity</td>
<td>1997 SIGMOD —</td>
</tr>
<tr>
<td>database</td>
<td>1982 SIGMOD — 1987 VLDB</td>
<td>indexing</td>
<td>1999 SIGMOD —</td>
</tr>
<tr>
<td>nested</td>
<td>1984 VLDB — 1991 VLDB</td>
<td>xml</td>
<td>1999 VLDB —</td>
</tr>
</tbody>
</table>

- Technical words even though all words examined
- Trend of “data base” vs. “database”
<table>
<thead>
<tr>
<th>Word</th>
<th>Interval of burst</th>
</tr>
</thead>
<tbody>
<tr>
<td>grammars</td>
<td>1969 STOC — 1973 FOCS</td>
</tr>
<tr>
<td>automata</td>
<td>1969 STOC — 1974 STOC</td>
</tr>
<tr>
<td>languages</td>
<td>1969 STOC — 1977 STOC</td>
</tr>
<tr>
<td>machines</td>
<td>1969 STOC — 1978 STOC</td>
</tr>
<tr>
<td>recursive</td>
<td>1969 STOC — 1979 FOCS</td>
</tr>
<tr>
<td>classes</td>
<td>1969 STOC — 1981 FOCS</td>
</tr>
<tr>
<td>some</td>
<td>1969 STOC — 1980 FOCS</td>
</tr>
<tr>
<td>sequential</td>
<td>1969 FOCS — 1972 FOCS</td>
</tr>
<tr>
<td>equivalence</td>
<td>1969 FOCS — 1981 FOCS</td>
</tr>
<tr>
<td>programs</td>
<td>1969 FOCS — 1986 FOCS</td>
</tr>
<tr>
<td>program</td>
<td>1970 FOCS — 1978 STOC</td>
</tr>
<tr>
<td>on</td>
<td>1973 FOCS — 1976 STOC</td>
</tr>
<tr>
<td>complexity</td>
<td>1974 STOC — 1975 FOCS</td>
</tr>
<tr>
<td>problems</td>
<td>1975 FOCS — 1976 FOCS</td>
</tr>
<tr>
<td>relational</td>
<td>1975 FOCS — 1982 FOCS</td>
</tr>
<tr>
<td>logic</td>
<td>1976 FOCS — 1984 STOC</td>
</tr>
<tr>
<td>vlsi</td>
<td>1980 FOCS — 1986 STOC</td>
</tr>
<tr>
<td>probabilistic</td>
<td>1981 FOCS — 1986 FOCS</td>
</tr>
<tr>
<td>how</td>
<td>1982 STOC — 1988 STOC</td>
</tr>
<tr>
<td>parallel</td>
<td>1984 STOC — 1987 FOCS</td>
</tr>
<tr>
<td>algorithm</td>
<td>1984 FOCS — 1987 FOCS</td>
</tr>
<tr>
<td>graphs</td>
<td>1987 STOC — 1989 STOC</td>
</tr>
<tr>
<td>learning</td>
<td>1987 FOCS — 1997 FOCS</td>
</tr>
<tr>
<td>competitive</td>
<td>1990 FOCS — 1994 FOCS</td>
</tr>
<tr>
<td>randomized</td>
<td>1992 STOC — 1995 STOC</td>
</tr>
<tr>
<td>approximation</td>
<td>1993 STOC —</td>
</tr>
<tr>
<td>improved</td>
<td>1994 STOC — 2000 STOC</td>
</tr>
<tr>
<td>codes</td>
<td>1994 FOCS —</td>
</tr>
<tr>
<td>approximating</td>
<td>1995 FOCS —</td>
</tr>
<tr>
<td>quantum</td>
<td>1996 FOCS —</td>
</tr>
</tbody>
</table>

“how”, “on”, “some” represent titling conventions:
“How to construct random fields”
“How to generate and exchange secrets”
<table>
<thead>
<tr>
<th>Word</th>
<th>Interval of burst</th>
</tr>
</thead>
<tbody>
<tr>
<td>gentlemen</td>
<td>1790 - 1800</td>
</tr>
<tr>
<td>militia</td>
<td>1801 - 1816</td>
</tr>
<tr>
<td>british</td>
<td>1809 - 1814</td>
</tr>
<tr>
<td>enemy</td>
<td>1812 - 1814</td>
</tr>
<tr>
<td>savages</td>
<td>1812 - 1819</td>
</tr>
<tr>
<td>spain</td>
<td>1818 - 1821</td>
</tr>
<tr>
<td>likewise</td>
<td>1818 - 1824</td>
</tr>
<tr>
<td>chambers</td>
<td>1833 - 1835</td>
</tr>
<tr>
<td>french</td>
<td>1833 - 1835</td>
</tr>
<tr>
<td>bank</td>
<td>1833 - 1836</td>
</tr>
<tr>
<td>france</td>
<td>1834 - 1835</td>
</tr>
<tr>
<td>texas</td>
<td>1843 - 1846</td>
</tr>
<tr>
<td>annexation</td>
<td>1844 - 1846</td>
</tr>
<tr>
<td>mexican</td>
<td>1845 - 1847</td>
</tr>
<tr>
<td>her</td>
<td>1846 - 1847</td>
</tr>
<tr>
<td>mexico</td>
<td>1846 - 1847</td>
</tr>
<tr>
<td>steamers</td>
<td>1847 - 1849</td>
</tr>
<tr>
<td>oregon</td>
<td>1847 - 1852</td>
</tr>
<tr>
<td>california</td>
<td>1848 - 1852</td>
</tr>
<tr>
<td>kansas</td>
<td>1856 - 1858</td>
</tr>
<tr>
<td>slavery</td>
<td>1857 - 1860</td>
</tr>
<tr>
<td>whilst</td>
<td>1857 - 1860</td>
</tr>
<tr>
<td>slaves</td>
<td>1859 - 1863</td>
</tr>
<tr>
<td>rebellion</td>
<td>1861 - 1871</td>
</tr>
<tr>
<td>emancipation</td>
<td>1862 - 1864</td>
</tr>
</tbody>
</table>

**State of the Union: 1790-2001**

- **war of 1812**
- **Treaty with Spain**
- **Texas annexation**
- **Mexican-American war**
- **Westward expansion, Oregon trail**
- **Gold rush**
- **Kansas Massacre/Slavery issues**
- **Civil war**
- **Emancipation Proclamation**
Content and time interleaved

- Fix e-mail arrival times, but randomly permute content of the messages
- Use two state model to measure total weight for all words
- Original order is an order of magnitude larger than random (369,980 vs. 25,141)
- Random hierarchies aren’t as deep (3865 vs. 16.7 over depth 2)
Other uses

- Web access logs
- Identifying interesting messages by retrieving those messages at the beginning of a burst