Temporal Dynamics of On-Line Information Streams

Jon Kleinberg

Department of Computer Science, Cornell University. kleinber@cs.cornell.edu

1 Introduction

A number of recent computing applications involve information arriving continuously over time in the form of a *data stream*, and this has led to new ways of thinking about traditional problems in a variety of areas. In some cases, the rate and overall volume of data in the stream may be so great that it cannot all be stored for processing, and this leads to new requirements for efficiency and scalability. In other cases, the quantities of information may still be manageable, but the data stream perspective takes what has generally been a static view of a problem and adds a strong temporal dimension to it.

Our focus here is on some of the challenges that this latter issue raises in the settings of text mining, on-line information, and information retrieval. Many information sources have a stream-like structure, in which the way content arrives over time carries an essential part of its meaning. News coverage is a basic example; understanding the pattern of a developing news story requires considering not just the content of the relevant articles but also how they evolve over time. Some of the other basic corpora of interest in information retrieval — for example, scientific papers and patents — show similar temporal evolution over time-scales that can last years and decades. And the proliferation of on-line information sources and on-line forms of communication has led to numerous other examples: e-mail, chat, discussion boards, and weblogs (or "blogs") all represent personal information streams with intricate topic modulations over time.

Indeed, all these information sources co-exist on-line — news, e-mail, discussion, commentary, and the collective output of professional and research communities; they form much of the raw material through which Internet users navigate and search. They have also served to make the "time axis" of information increasingly visible. One could argue that these developments have led to a shift in our working metaphor for Internet and Web information, from a relatively static one, a "universal encyclopedia," to a much more dynamic one, a current awareness medium characterized by the complex development of topics over time-scales ranging from minutes to years.

What are the right techniques for dealing with the temporal dynamics of information streams? Some of the complexities inherent in such streams can be seen in the paradigmatic examples of news and e-mail. Each provides a reader with a sequence of documents exhibiting a *braided* and *episodic* character: *braided*, in the sense that many parallel, topically coherent streams are merged and interwoven into the single stream that the reader sees; and *episodic*, in the sense that topics generally grow in intensity over a temporally coherent period, and then fade again. Despite these twin sources of complexity, however, these information streams in their raw forms lack any explicit organizing structure beyond the granularity of individual articles or messages. Thus, a first step in working with such information streams is to define appropriate structures that abstract their intertwined topics, their multiple bursty episodes, and their long-term trends.

In this survey we discuss a number of approaches that have been proposed in recent years for working with the temporal properties of information streams. In Section 2 we give an overview of some of the basic techniques; we begin with one of the earliest systematic efforts on these problems, the Topic Detection and Tracking initiative, and then go on to discuss three subsequent approaches — threshold-based methods, state-based methods, and trend-based methods — that search for different types of patterns in information streams. In Section 3 we discuss some recent applications of these techniques to the analysis of weblogs, queries to Web search engines, and usage data at high-traffic Web sites. Finally, we conclude in Section 4 with some thoughts on directions for further research.

Analyzing the temporal properties of these types of information streams is part of the broader area of *sequential pattern mining* within the field of data mining, and can be viewed as an application of *time-series analysis*, a fundamental area in statistics. In order to keep the scope of this survey manageable, we have not attempted to cover these more general topics; we refer the reader to the papers of Agrawal and Srikant and of Mannila et al. [2, 28] for some of the foundational work on sequential pattern mining, and to the text by Hand et al. [18] for further results in this area.

2 Techniques: Thresholds, State Transitions, and Trends

Topic Detection and Tracking.

The Topic Detection and Tracking (TDT) research initiative represented perhaps the first systematic effort to deal with the issues raised by text information streams. Observing that it is tricky to pin down the underlying set of problems that need to be solved, it sought to define a concrete set of tasks based on the general problem of identifying coherent topics in a stream of news stories.

Allan et al. [5] and Papka [30] describe some of the initial considerations that went into the formulation of these tasks. To begin with, the TDT researchers distinguished between the notion of a *topic* — a staple of information retrieval research — and an *event*, the latter referring to a unique occurrence that happened at a specific time. For example, “baseball” would be considered a topic, whereas “Game 6 of the 1975 World Series” would be considered an event. This kind of distinction is natural in the context of news coverage, although of course the conceptual boundary between topics and events is quite flexible. Further refinements of these definitions added other notions, such as *activities.*

Within this framework, one imagines a TDT system operating roughly as follows. Given a stream of news articles, the system should automatically recognize those stories that discuss an event it has not seen before, and should begin tracking each of these events so as to identify the sub-stream of further stories that discuss it. Implicit in this description is a pair of basic tasks that can be evaluated separately: *new event detection*, in which the goal is to recognize the first story to discuss an event, and *event tracking*, in which the goal is to group stories that discuss the same event. Other TDT tasks have been defined as well, including the *segmentation* of a continuous stream of news text into distinct stories; this is a necessary prerequisite for many of the other tasks, and crucial for transcriptions of audio broadcasts, where the boundaries between stories are not generally made explicit in a uniform way. Finally, a distinction is made between the *retrospective* versions of these tasks, in which the full corpus of news stories is available for analysis, and the *on-line* versions, in which decisions must be made in real-time as news stories appear. One challenge inherent in the detection and tracking tasks is that events can have very different temporal “shapes”: an unexpected event like a natural disaster comes on with a sharp spike, while an expected event like an election has a slow build-up and potentially a quicker decay after the event is over.

A range of different techniques from information retrieval have been shown to be effective on the TDT tasks, including clustering methods for event tracking that trade off between content similarity and temporal locality. The volume edited by Allan [4] covers much of the state of the art in this area. The general framework developed for the TDT project has proved useful in a number of subsequent efforts that have not directly used the TDT task descriptions. For example, recent work on the *Newsjunkie* system [15] sought techniques for determining the *novelty* in individual news stories, relative to previously seen stories concerned with the same general topic or event; quantifying novelty in this sense can be viewed as a relaxation of the task of new event detection.

Information Visualization.

At roughly the same time as the TDT project, the information visualization community also began investigating some of the issues inherent in text information streams [19, 29, 37]. The goal was to find visual metaphors by which users could navigate large collections of documents with an explicit temporal dimension.

The ThemeRiver system [19] depicts a text collection that evolves over time, such as a corpus of news stories, using a "river" metaphor: differently colored currents in the river indicate different topics in the collection, and the currents vary in width to indicate news stories that have a large volume of relevant articles at a particular point in time. Note that this visualization approach nicely captures the view, discussed above, of text information streams as a braided, episodic medium: the individual "braids" appear in ThemeRiver as the colored currents, while the episodes stand out as gradual or sudden widenings of a particular current.

Timelines and Threshold-Based Methods.

Timelines are a common means of representing temporal data, appearing in computational applications (see e.g. [6, 31]) and familiar from more traditional print media as well. Swan, Allan, and Jensen [33, 34, 35] considered the problem of creating timelines for document streams, again focusing on collections of news articles. They framed the problem as a selection task: the construction of a timeline should involve retrospectively identifying the small set of most significant episodes from a large stream of documents, assocating a descriptive tag with each one, and displaying these tagged episodes in their natural temporal order. Formalizing this notion requires an implementable definition of *episodes* in a text stream, together with a way of ranking episodes by significance.

Swan et al. base their definition of episodes on time-varying *features* in the text — words, named entities, and noun features are all examples of possible features for this purpose. Now, each feature has an average rate at which it appears in the corpus; for a news stream, this would be the total number of occurrences of the feature divided by the number of days represented in the corpus. An episode is then associated with a contiguous interval of time during which one of these features exceeds its average rate by a specific threshold; Swan et al. determine this threshold on a per-feature basis using a χ^2 test, and group together consecutive days over which a given feature exceeds its threshold. Thus, for example, if we were following news articles early in a U.S. Presidential election year, we might find that for a number of days in a row, the word "Iowa" appears a significant factor more frequently than it standardly does in the news, reflecting coverage of the Iowa caucuses. The most significant episodes computed this way can then be included in the timeline, each with a start and end time, and each associated with a specific word or phrase.

State-Based Methods.

The number of occurrences of a given feature can be quite noisy, varying widely from one day to the next even in the middle of an event in which the feature figures prominently. Swan et al. observe that this poses difficulties in the construction of intervals to represent episodes on a timeline; a feature being tracked may oscillate above and below the threshold, turning something that intuitively seems like a single long interval into a sequence of shorter ones, interrupted by the below-threshold days. Thus, to continue our example from above, there may not be a single interval associated with the word "Iowa" early in election coverage, but rather a set of short ones separated by small gaps. This was also an observation in the development of ThemeRiver, that Sundays and other days with lighter news coverage show up as recurring "pinch points" in the representation of the stream. Swan and Allan proposed heuristics to deal with this problem [34]; for example, one can merge two intervals corresponding to the same feature if they are separated by only a single day on which the feature was below threshold.

In [23], the author proposed a method for defining episodes in a stream of documents that deals with the underlying noise by explicitly positing a source model for the words in the documents. The motivation comes from queueing theory, where a bursty source of network traffic is often modeled as a probabilistic automaton: at any given point in time, the automaton can be in one of several different states, the rate at which traffic is generated is determined by this state, and transitions between states are determined probabilistically [7, 13, 21].

In the case of documents, one can imagine each word to be a type of "traffic" generated by its own source, modeled as an automaton. Some words are highly "bursty" in the sense that their frequency spikes when a particular event is in the news; in the probabilistic model, this corresponds to a contiguous interval of time during which the automaton is in a state corresponding to a high rate of traffic. Such *word bursts* are the intervals that correspond to discrete episodes in this model; one can identify them using a Viterbi-style algorithm [32], taking the times at which a given word occurs as input and computing the most likely sequence of states of the automaton that is assumed to be generating it. Note the distinction with the threshold-based methods described earlier: rather than associating episodes with intervals when the *observed rate* of the word is consistently high, it associates them with intervals in which the *state* of the underlying automaton has a high rate. Thus, by modeling state transitions as relatively low-probability events, the bursts that are computed by the algorithm tend to persist through periods of noise: remaining in a single state through such a noisy period is viewed as more probable than performing many transitions.

Moreover, an automaton with a large set of states corresponding to increasingly rapid rates can expose the natural nested structure of word bursts: in the midst of an elevated use of a particular term, the rate of use may increase further, producing a "burst within a burst." This nesting can in principle be iterated arbitrarily, yielding a natural hierarchy. To continue our example of U.S. presidential election coverage in the news, one could imagine over a multi-year period seeing bursts for the word "convention" in the summers of presidential election years, with two inner, stronger bursts corresponding to the Democratic and Republican conventions.

A timeline for a document stream can be naturally constructed using a two-state automaton associated with each word — a slower *base state* corresponds to the average rate of appearance of the word, while a *burst state* corresponds to a faster "burst rate." Using just two states does not expose any nested or hierarchical structure, but it makes it very simple to interpret the burst intervals. For each word, one computes the intervals during which the automaton is in the burst state. Now, by definition, in each such interval the automaton is more likely to be in the burst state than the base state; we define the *weight* of the interval to be the factor by which the probability of the burst state exceeds the probability of the base state over the course of the interval. Essentially, the weight thus represents our "confidence" that the automaton is indeed in the burst state. A timeline with a given number of items can simply be defined to consist of the intervals with the highest weight.

This approach is considered in the context of several different kinds of document streams in [23], including e-mail and titles of scientific papers. To give a sense for the kind of timelines that can be constructed this way, Figure 1 shows an example from [23], the results of this method applied to the document stream consisting of titles of all papers from the database conferences SIGMOD and VLDB, 1975-2001. A two-state automaton is associated with each word, in which the burst state has twice the rate of the base state, and the 30 burst intervals of highest weight are depicted. Note that all words are included in the analysis, but, matching our intuition, we see that stop-words do not occur on the list because they do not tend to be very bursty. On the other hand, certain words appearing on the list do reflect trends in language use rather than in technical content; for example, the bursts for 'data,' 'base,' and 'bases' in the years 1975-1981 in Figure 1 arise in large part from the fact that the term 'database' was written as two words in a significant number of the paper titles during this period.

Figure 1 is primarily just an illustrative example; Mane and Börner have used this burst detection approach in a similar but much more extended way as part of a large-scale scientometric analysis of topics in the *Proceedings of the National Academy of Sciences* over the years 1982-2001 [27]. The bursty topics can be viewed as providing a small set of natural "entry points" into a much larger collection of documents; for example, they provide natural starting points for issuing queries to the collection, as well as the raw material for a clustering analysis of the content. As an instance of the latter application, Mane and Börner computed a two-dimensional representation of term co-occurrences for a collection of words and phrases selected to maximize a combination of burstiness and frequency, and this representation was then evaluated by domain experts.

Another observation based on the example in Figure 1 is that the state-based model has the effect of producing longer bursts than a corresponding use of thresholds; although the burst state had twice the rate of the base

Word	Interval of burst
data	1975 SIGMOD — 1979 SIGMOD
base	1975 SIGMOD — 1981 VLDB
application	1975 SIGMOD — 1982 SIGMOD
bases	1975 SIGMOD — 1982 VLDB
design	1975 SIGMOD — 1985 VLDB
relational	1975 SIGMOD — 1989 VLDB
model	1975 SIGMOD — 1992 VLDB
large	1975 VLDB — 1977 VLDB
schema	1975 VLDB — 1980 VLDB
theory	1977 VLDB — 1984 SIGMOD
distributed	1977 VLDB — 1985 SIGMOD
data	1980 VLDB — 1981 VLDB
statistical	1981 VLDB — 1984 VLDB
database	1982 SIGMOD — 1987 VLDB
nested	1984 VLDB — 1991 VLDB
deductive	1985 VLDB — 1994 VLDB
transaction	1987 SIGMOD — 1992 SIGMOD
objects	1987 VLDB — 1992 SIGMOD
object-oriented	1987 SIGMOD — 1994 VLDB
parallel	1989 VLDB — 1996 VLDB
object	1990 SIGMOD — 1996 VLDB
mining	1995 VLDB —
server	1996 SIGMOD — 2000 VLDB
sql	1996 VLDB — 2000 VLDB
warehouse	1996 VLDB —
similarity	1997 SIGMOD —
approximate	1997 VLDB —
web	1998 SIGMOD —
indexing	1999 SIGMOD —
xml	1999 VLDB —

Fig. 1. The 30 bursts of highest weight using titles of all papers from the database conferences SIGMOD and VLDB, 1975-2001.

state, most of the terms did not maintain a rate of twice their overall average throughout the entire interval. Indeed, only five terms in these paper titles appeared continuously at twice their average rate for a period of more than three years. David Jensen makes the observation that this kind of state-based smoothing effect may be more crucial for some kinds of text streams than for others: using just the titles of papers introduces a lot of noise that needs to be dealt with, while news articles tend to be written so that even readers joining the coverage many days into the event will still be able to follow the context [20].

In a sense, one can view a state-based approach as defining a more general, "relaxed" notion of a threshold; the optimization criterion inherent in determining a maximum-likelihood state sequence is implicitly determining how far above the base rate, and for how long, the rate must be (in an amortized sense) in order for the automaton to enter the burst state. Zhu and Shasha considered a more direct way of generalizing the threshold approach [38]. For each possible length k, they allow a user-defined threshold $t(k)$; any interval of length k containing at least $t(k)$ occurrences of the feature is declared to be a burst. Note that this means that a single feature can have overlapping burst intervals of different lengths. They then go on to develop fast algorithms for enumerating all the bursts associated with a particular feature.

Trend-Based Methods.

Thus far we have been discussing models for episodes as regions of increased density — words fluctuate in their patterns of occurrence, and we have been seeking short intervals in which a word occurs an unusual number of times. Liebscher and Belew propose a different structure of interest in a stream of documents: the sets of words that exhibit the most pronounced rising and falling trends over the entire length of the corpus [26]. For streams that are grouped over time into relatively few bins — such as yearly tabulations of research papers like we considered in Figure 1 — they apply linear regression to the set of frequencies of each term (first averaging each value with its two neighbors). They also argue that less sparse data would be amenable to more complex time-series analysis. Features with the most positive slopes represent the strongest rising terms, while those with the most negative slopes represent the strongest falling terms.

In addition to summarizing long-range trends in the stream, Liebscher and Belew also propose an application to *temporal term weighting*: if a user issues a query for a particular term that rises significantly over time, early documents containing the term can be weighted more heavily. The argument is that such early documents may capture more fundamental aspects of a topic that later grew substantially in popularity; without temporal weighting, it would be hard to favor such documents in the results of a search. A similar approach can be applied to terms that decrease significantly over time.

Two-Point Trends.

A number of search engines have recently begun using the notion of rising and falling terms to present snapshots of trends in search behavior; see for example Google's *Zeitgeist* and Ask Jeeves's *Top Searches* [8, 16]. The canonical approach for doing this is to compare the frequency of each search term in a given week to its frequencies in the previous week, and to find those for which the change has been largest.

Of course, this definition depends critically on how we choose to measure change. The tricky point here is that there are many natural ways to try

Normalized absolute change:

Falling Words	Rising Words
to	iraq
i	are
president	and
security	iraqi
it	of

Relative change:

Falling Words	Rising Words
welfare	aids
she	rising
mexico	instead
created	showing
love	government

Probabilistic generative model:

Falling Words	Rising Words
homeland	iraq
trade	iraqi
security	aids
senate	seniors
president	coalition

Fig. 2. Most prominent falling and rising words in text of weekly U.S. Presidential radio addresses, comparing 2002 to 2003.

quantifying a trend involving just two data points — i.e., in the case of the search engine application, the two data points are the frequencies of occurrence in the previous week and the current week. To make this concrete, suppose we have text from two different time periods, and we want to define the amount of *change* experienced by a given word w. We define each occurrence of each word to be a *token*, and let n_0 and n_1 denote the total number of token in the text from the first and second periods respectively. Let $f_0(w)$ and $f_1(w)$ denote the total number of occurrences of the word w in the first and second periods respectively, and define $p_0(w) = f_0(w)/n_0$ and $p_1(w) = f_1(w)/n_1$.

Now, two basic ways to rank words by the significance of their "falling" and "rising" patterns would be to measure absolute change or relative change. We define the *absolute change* of w to be $f_1(w) - f_0(w)$; we can also define the *normalized absolute change* as $f_1(w)(n_0/n_1) - f_0(w)$ to correct for the fact that the amount of text from the first and second periods may differ significantly. Analogously, we define the *relative change* of w to be $f_1(w)/f_0(w)$ and the *normalized relative change* to be $(n_0 f_1(w))/(n_1 f_0(w))$. Charikar et al. [10], motivated by the application to trend detection at Google, discuss streaming algorithms for computing these types of change measures on large datasets. While the exact methods for measuring change in search frequency have not been made public, Ask Jeeves indicates that it ranks terms by a variant of relative change, while Charikar et al. suggest that Google ranks terms by a variant of absolute change.

If one thinks about it intuitively, absolute and relative change are each quite extreme measures for this task, though in opposite directions. In order to be among the most highly ranked for absolute change, a word must have a very high rate of occurrence; this means that the lists of most significant risers and fallers will be dominated by very common words, and even fluctuations in stop-word use can potentially swamp more interesting trends. On the other hand, in order to be among the most highly ranked for relative change, a word generally must occur extremely rarely in one of the two periods and have a massive spike in the other; such trends can often be artifacts of a particular transient pattern of usage. (As a degenerate case of this latter issue, it is not clear how to deal with words that failed to occur in one of the two periods.)

As an illustration of these issues, Figure 2 shows the five most significant falling and rising words from the text of weekly U.S. Presidential radio addresses, with the two periods consisting of the years 2002 and 2003. A ranking by normalized absolute change mainly puts stop-words in the top few positions, for the reason suggested above; mixed in with this are more topic-specific words that experienced huge fluctuations, like “Iraq.” The ranking by relative change is harder to interpret, but consists primarily of words that occurred very rarely in one year or the other.

Is there a principled way to interpolate between these two extremes, favoring topic-specific words that experienced large changes and that, while relatively frequent, were still not among the most overwhelmingly common words in the corpus? Our proposal here is that the state-based models considered earlier provide a very simple measure of change that performs such an interpolation. Recall that, in ranking bursts produced by a two-state automaton, we used a *weight* function that measured the probability of the burst relative to the probability of the automaton having remained in its base state. For the present application, we can imagine performing the following analogous calculation. Suppose that, after observing the first period, we posit the following model for generating the words in the second period: n_1 tokens will be generated independently at random, with the j^{th} token taking the value w with probability $p_0(w)$. In other words, the observed word frequencies in the first period are assumed to generate the tokens in the second period via a simple probabilistic generative model.

Now, the significance of an increase in word w in the second period is based simply on the probability that $f_1(w)$ tokens out of n_1 would take the value w; given the model just defined, this is equal to

$$\binom{n_1}{f_1(w)} p_0(w)^{f_1(w)} (1 - p_0(w))^{n_1 - f_1(w)}.$$

(Since this will typically be a very small quantity, we work with logarithms to perform the actual calculation.) By choosing the words with $p_1(w) > p_0(w)$ for which this probability is lowest, we obtain a list of the top rising terms; the analogous computation produces a list of the top falling terms. The third part of Figure 2 shows the results of this computation on the same set of U.S. Presidential radio addresses; while we see some overlap with the previous two lists, the probabilistic generative model arguably identifies words in a way that is largely free from the artifacts introduced by the two simpler measures.

Of course, there are many other ways to interpolate between absolute and relative change. The point is simply that a natural generative model can produce results that are much cleaner than these cruder measures, and hence, without significantly increasing the difficulty of the ranking computation, yield trend summaries of this type that are potentially more refined.

3 Applications: Weblogs, Queries, and Usage Data

Weblogs.

Personal home pages have been ubiquitous on the Web since its initial appearance; they are the original example of spontaneous Web content creation on a large scale by users who in many cases are not technically sophisticated. *Weblogs* (also referred to as *blogs*) can be viewed as a more recent step in the evolution of this style of personal publishing, consisting generally of dated, journal-style entries with links and commentary. Whereas a typical home page may be relatively static, a defining feature of weblogs is this explicit temporal annotation and regular updating. A significant sub-population of webloggers focus on news events, commenting on items of interest that they feel are being ignored or misportrayed by traditional news organizations; in some cases, these weblogs have readerships of substantial size. As a result, it is also natural to consider the space of weblogs as containing a stream of news that parallels the mainstream news media, much more heterogeneous both in focus and in authoring style.

All this makes it natural to apply the type of temporal analysis we have been discussing to the information stream consisting of indexable weblog content. In one of the first such applications to this domain, Dan Chan, motivated in part by the word burst model of [23], implemented a word-burst tracker on his widely-used weblog search site Daypop [11]. As we saw earlier with bursty topics from collections of research papers [27], the Daypop application provides another basic example of the way in which a list of bursty words can provide natural entry points into a corpus — in this case to search for items of interest in current news and commentary. By including word bursts computed both from weblog text and from headlines in the mainstream news, the site allows one to identify both overlaps and differences in the emphases of these two parallel media.

In a roughly concurrent study, Kumar et al. focused on a combined analysis of weblog content and link structure [24]. Because so much of weblog discourse takes place through reference to what other webloggers are reading and writing, the temporal dynamics of the link structure conveys important information about emerging topics and discussion. Kumar et al. work on the problem of identifying *subgraph bursts* in this structure, consisting of a significant number of link appearances among a small set of sites in a short period of time. They observe that this poses greater computational challenges than the analogous search for bursty words: whereas word bursts can be identified one word at a time, subgraph bursts need a more global analysis, since they are not the result of activity at any one site in isolation.

From a methodological point of view, one appealing feature of weblogs as a domain is the richness of the data; in principle, one can track the growth of a topic at an extremely fine-grained level, using both the detailed temporal annotations and the links that webloggers use to indicate where they learned a piece of information. Adar et al. [1] and Gruhl et al. [17] exploit this to determine not just *whether* a given topic burst has occurred, but to some extent *how* and *why* it occurred. This involves modeling the spread of the topic as a kind of epidemic on the link structure — identifying the “seeds” of the topic, where it appeared earliest in time, and then tracing out the process of *contagion* by which the topic spread from one weblog to another. Adar et al. use this to define a ranking measure for weblogs, based on an estimate of their ability to start such an epidemic process. Gruhl et al. attempt to learn parameters of these epidemic processes, using an instance of the *General Cascade Model* for the diffusion of information in social networks, proposed by Kempe et al. [22].

Search Engine Queries.

The logs of queries made to a large search engine provide a rich domain for temporal analysis; here the text stream consists not of documents but of millions of very short query strings issued by search engine users. Earlier, we discussed a simple use of query logs to identify the most prominently rising and falling query terms from one week to the next. We now discuss two recent pieces of work that perform temporal analysis of such logs for the purpose of enhancing and improving search applications.

Vlachos et al. [36] study query logs from the MSN search engine, providing techniques to identify different types of temporal patterns in them. For example, the frequency of the query “cinema” has a peak every weekend, while the frequency of the query “Easter” build to a single peak each spring and then drops abruptly. Vlachos et al. apply a threshold-based technique to a moving average of the daily frequencies for a given query in order to find the burst periods for the query, and they propose a “query-by-burst” technique that can identify queries with burst periods that closely overlap in time. Using Fourier analysis, they also build a representation of the temporal periodicities in a query’s rate over time, and then apply time-series matching techniques to identify other queries with very similar temporal patterns — such similarity can be taken as a form of evidence in the identification of related queries.

Diaz and Jones [12] also build temporal profiles, but they use the timestamps of the documents *returned* in response to a query, rather than the timestamps of the invocations of the query by users. Thus, in a corpus of news articles, a query about a specific natural disaster will tend to produce articles that are tightly grouped around a single date. Such temporal profiles become features of a query that can be integrated into a probabilistic model of document relevance; experiments in [12] show that the use of such features can lead to performance improvements.

Usage Data.

On-line activity involves user behavior over short time-scales — browsing, searching, and communicating — as well as the creation of less ephemeral written content, including the text of pages and files on the Web. On-line information streams encode information about both these kinds of data, and the division between the two is far from clear. While research paper archives and news streams clearly represent written content, the written material in weblogs is quite closely tied to the usage patterns of its authors — their recent history of reading and communication. The text that is posted on forums and discussion boards also reflects the dynamics of visitors to sites on a rapid time-scale; and search engine query logs also occupy a kind of middle ground, consisting of text encoding the behavior of searchers at the site.

To conclude our survey of different application areas, we consider an on-line information stream that encodes a basic form of usage data — the sequence of downloads performed by users at a high-traffic Web site. Many active Web sites are organized around a set of items that are available for purchase or download, using a fairly consistent high-level metaphor: at an e-commerce site like amazon.com, or an archive of research papers like arxiv.org, there is navigational structure at the front, followed by a large set of “description pages, one associated with each item that is available. Each description page provides the option to acquire the corresponding item.

In joint work with Jon Aizen, Dan Huttenlocher, and Tony Novak [3], the author studied the dynamics of download behavior at one such site, the Internet Archive, which maintains a publically accessible media collection consisting of old films, live concerts, free on-line books, and other items available for download. The basic definition in [3] is the “batting average” (henceforth, the BA) of an on-line item: the number of acquisitions divided by the number of visits to the description page. Thus a high BA is a reflection of item quality, indicating that a large fraction of users who viewed the item’s description

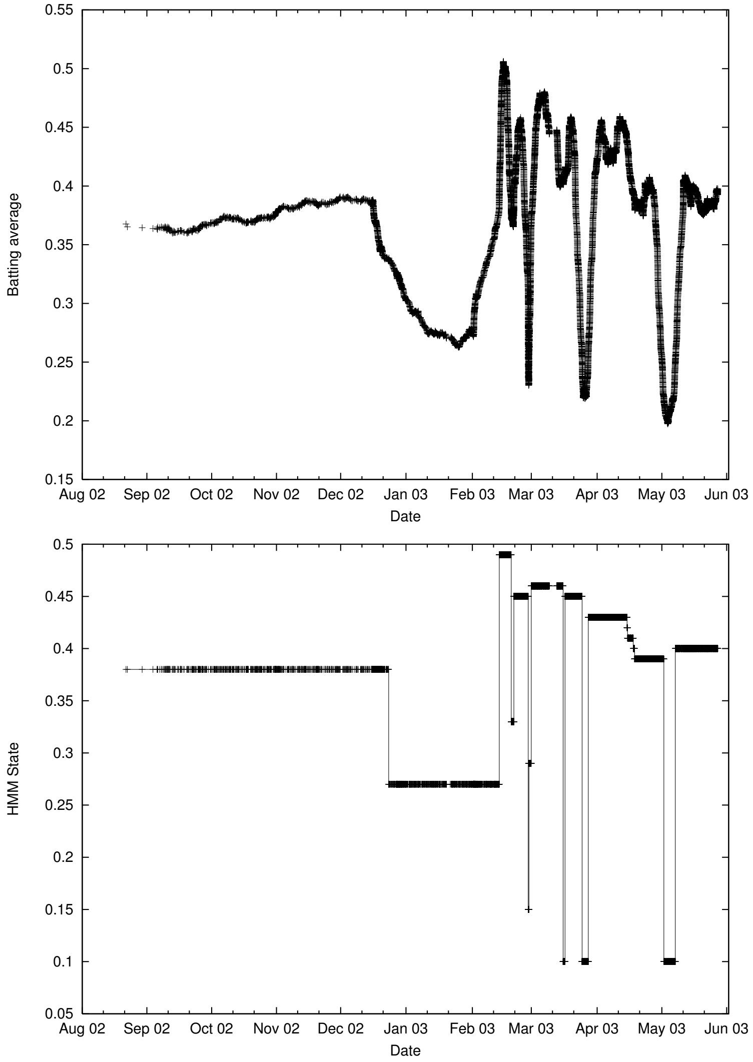

Fig. 3. Tracking the batting average of a downloadable movie at the Internet Archive, as a function of time. The upper plot shows smoothing by averaging with a Gaussian sliding window; the lower plot shows the maximum-likelihood state sequence in a Hidden Markov model.

chose to download it. Note that this is different from a simple "hit counter" which measures the raw number of item visits or downloads; the BA addresses the more subtle notion of users' reactions to the item description.

Suppose we want to measure fluctuations in the BA over time, looking for evidence of bursty behavior in which the BA of an item changes suddenly; in order to do this it is necessary to define a meaningful notion of the “instantaneous BA” of an item as a function of time. Such an instantaneous BA must be synthesized from the 0-1-valued sequence of user download decisions: each entry in the usage log simply reflects whether a particular chose to download the item or not. In [3], two different methods are considered for determining an instantaneous BA from this sequence. First, one can apply Gaussian smoothing, computing an average of each point in the download sequence with its neighbors, weighted by coefficients that decay according to a Gaussian function. Alternately, one can treat the download decisions as the 0-1-valued outputs of a Hidden Markov model (HMM), with hidden states representing underlying download probabilities; the maximum-likelihood state sequence can then be taken to represent the item’s BA at each point in time. A large number of states was used in the HMM computation, so as to densely sample the set of possible probabilities; in order to be able to handle long download sequences efficiently, this required the use of an improved maximum-likelihood algorithm due to Felzenszwalb et al. [14] that took advantage of the structure of the state transitions so as to run in time linear in the number of states, rather than quadratic. The HMM approach produces sharp, discrete changes in the BA while the Gaussian smoothing approach yields more gradual changes; Figure 3 illustrates this, showing the BA as a function of time for each of these approaches applied to the same downloadable movie at the Internet Archive. One can see how the sharp steps in the HMM plot approximately line up with the more gradual curves in the Gaussian-smoothed plot.

Are these sharp steps useful in the context of the underlying application? In [3], it is argued that the discrete breakpoints in the BA produced by the HMM in fact capture a crucial feature of popularity dynamics at sites like the Internet Archive. Specifically, the download properties of an item, as reflected in measures like the BA, often change abruptly rather than gradually, due to the appearance of an external link from some other high-traffic site — which suddenly drives in a new mix of users with new interests — or due to on-site highlighting — for example, featuring the item on a top-level page at the Internet Archive — which also raises the item’s visibility. The addition of links or promotional text happens at a discrete point in time, and experiments in [3] show that state transitions in the HMM align closely with the times when these events happen. To continue with the plot in Figure 3, for example, the first two transitions in the HMM plot align with on-site highlighting performed by the Internet Archive, the next two sharp drops correspond to the appearance of links to the item’s description from high-traffic weblogs, and the final three drops corresponds to technical problems on the Internet Archive site. Examples like this, as well as the more systematic evaluation performed in [3], suggest how accurate tracking of changes to an item’s BA can help in understanding the set of events both on and off the site that affect the item’s popularity.

4 Conclusions

In studying the temporal dynamics of on-line information streams, an issue that arises in many contexts is the problem of *alignment*: we want to align the "virtual" events that we find in the stream with a parallel set of events taking place outside the stream, in the "real world." One sees this, for example, in aligning bursts in a news stream or weblog index with current events; or in aligning interesting temporal dynamics in search engine query terms with seasonal patterns, upcoming holidays, names in the news, or any of a range of other factors that drive search behavior. It would be interesting to formalize this alignment problem more fully, so that it can be used both to refine the analysis of these streams and to better evaluate the analyses that are performed. We have discussed some concrete examples of steps in this direction above; these include the use of time-series matching techniques to compare temporal profiles of different search engine queries [36], and the alignment of item popularity with on-site highlighting and off-site referrers at the Internet Archive. Another interesting piece of work in this direction is that of Lavrenko et al. [25], which aligns financial news stories with the behavior of financial markets. They and others make the point that the "virtual events" in the stream may not only follow as consequences of real-world events, but may also influence them — as when an unexpected news story about a company leads to a measurable effect on the company's stock price.

Relatively little work has been done on designing algorithms for the problems considered here in a "pure" streaming model of computation, where the data must be produced in one or a small number of passes with limited storage. For some of the applications, it is not clear that the data requirements are substantial enough that the use of such a streaming model will be necessary, but for other applications, including the analysis of query logs and clickstreams, there is a clear opportunity for algorithms developed in this style. The work of Ben-David et al. [9] and Charikar et al. [10] take steps in this direction, considering different ways of measuring change for streaming data.

Another direction that would be interesting to consider further is the problem of predicting bursts and other temporal patterns, when the data must be processed in real-time rather than through retrospective analysis. How early into the emergence of a bursty news topic, for example, can one detect it and begin forming estimates of its basic shape? This is clearly related to the problem of general *time-series prediction*, though there is the opportunity here to use a significant amount of domain knowledge based on the content of the text information streams being studied. Gathering domain knowledge involves developing a more refined understanding of the types of temporal patterns that regularly occur in these kinds of information streams. We know, for example, that the temporal profiles of some search engine queries are periodic (corresponding to weekends or annual holidays, for example) while others look like isolated spikes; we know that some news stories build up to an expected event while others appear suddenly in response to an unexpected one; but we do not have a careful classification or taxonomy of the full range of such patterns. Clearly this would be very useful in helping to to recognize temporal patterns as they arise in on-line content.

Finally, it is worth noting that as the applications of these techniques focus not just on news stories and professionally published documents but on weblogs, e-mail, search engine queries, and browsing histories, they move into domains that are increasingly about individual behavior. Many of these large data streams are personal; their subject is *us.* And as we continue to develop applications that extract detailed information from these streams — media players that have a more accurate picture of your changing tastes in music than you do, or e-mail managers that encode detailed awareness of the set of people you've fallen out of touch with — we will ultimately have to deal not just with technical questions but with yet another dimension of the way in which information-aware devices interact with our daily lives.

References

1. E. Adar, L. Zhang, L. A. Adamic, R. M. Lukose. "Implicit Structure and the Dynamics of Blogspace." Workshop on the Weblogging Ecosystem, at the International WWW Conference, 2004.
2. R. Agrawal, R. Srikant, "Mining sequential patterns," *Proc. Intl. Conf. on Data Engineering*, 1995.
3. J. Aizen, D. Huttenlocher, J. Kleinberg, A. Novak. "Traffic-Based Feedback on the Web." *Proc. Natl. Acad. Sci.* 101(Suppl.1):5254-5260, 2004.
4. J. Allan, ed., *Topic Detection and Tracking: Event Based Information Retrieval*, Kluwer, 2002.
5. J. Allan, J.G. Carbonell, G. Doddington, J. Yamron, Y. Yang, "Topic Detection and Tracking Pilot Study: Final Report," *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, Feb. 1998.
6. R. Allen, "Timelines as information system interfaces," *Proc. International Symposium on Digital Libraries*, 1995.
7. D. Anick, D. Mitra, M. Sondhi, "Stochastic theory of a data handling system with multiple sources," *Bell Syst. Tech. Journal* 61(1982).
8. Ask Jeeves, Top Searches at http://static.wc.ask.com/docs/about/jeevesiq.html?o=0
9. S. Ben-David, J. Gehrke D. Kifer, "Detecting Change in Data Streams," *Proc. 30th Intl. Conference on Very Large Databases (VLDB)*, 2004.
10. M. Charikar, K. Chen, M. Farach-Colton, "Finding Frequent Items in Data Streams," *Proc. Intl. Colloq. on Automata Languages and Programming*, 2002.
11. Daypop. http://www.daypop.com.
12. F. Diaz, R. Jones. "Using Temporal Profiles of Queries for Precision Prediction," *Proc. SIGIR Intl. Conf. on Information Retrieval*, 2004.
13. A. Elwalid, D. Mitra, "Effective bandwidth of general Markovian traffic sources and admission control of high speed networks," *IEEE Trans. Networking* 1(1993).
14. P. Felzenszwalb, D. Huttenlocher, J. Kleinberg, "Fast Algorithms for Large-State-Space HMMs with Applications to Web Usage Analysis," *Advances in Neural Information Processing Systems (NIPS)* 16, 2003.

15. E. Gabrilovich, S. Dumais, E. Horvitz. "NewsJunkie: Providing Personalized Newsfeeds via Analysis of Information Novelty," *Proceedings of the Thirteenth International World Wide Web Conference*, May 2004.
16. Google, Zeitgeist at http://www.google.com/press/zeitgeist.html.
17. D. Gruhl, R. Guha, D. Liben-Nowell, A. Tomkins. "Information Diffusion through Blogspace." *Proc. International WWW Conference*, 2004.
18. D. Hand, H. Mannila, P. Smyth, *Principles of Data Mining*, MIT Press, 2001.
19. S. Havre, B. Hetzler, L. Nowell, "ThemeRiver: Visualizing Theme Changes over Time," *Proc. IEEE Symposium on Information Visualization*, 2000.
20. D. Jensen, personal communication, July 2002.
21. F.P. Kelly, "Notes on effective bandwidths," in *Stochastic Networks: Theory and Applications*, (F.P. Kelly, S. Zachary, I. Ziedins, eds.) Oxford Univ. Press, 1996.
22. D. Kempe, J. Kleinberg, E. Tardos. "Maximizing the Spread of Influence through a Social Network." *Proc. 9th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2003.
23. J. Kleinberg. "Bursty and Hierarchical Structure in Streams." Proc. 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2002.
24. R. Kumar, J. Novak, P. Raghavan, A. Tomkins. "On the bursty evolution of Blogspace." *Proc. International WWW Conference*, 2003.
25. V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, J. Allan, "Mining of Concurrent Text and Time Series," *KDD-2000 Workshop on Text Mining*, 2000.
26. R. Liebscher, R. Belew. "Lexical dynamics and conceptual change: Analyses and implications for information retrieval." *Cognitive Science Online* 1(2003).
27. K. Mane, K. Börner. "Mapping topics and topic bursts in PNAS." Proceedings of the National Academy of Sciences 101(Suppl.1):5287-90, 2004.
28. H. Mannila, H. Toivonen, A.I. Verkamo, "Discovering frequent episodes in sequences," *Proc. Intl. Conf. on Knowledge Discovery and Data Mining*, 1995.
29. N. Miller, P. Wong, M. Brewster, H. Foote, "Topic Islands: A Wavelet-Based Text Visualization System," *Proc. IEEE Visualization*, 1998.
30. R. Papka, *On-line New Event Detection, Clustering, and Tracking*, Ph.D. thesis, Univ. Mass. Amherst, September 1999.
31. C. Plaisant, B. Milash, A. Rose, S. Widoff, B. Shneiderman. "LifeLines: visualizing personal histories," *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1996.
32. L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE* 77(1989).
33. R. Swan, J. Allan, "Extracting significant time-varying features from text," *Proc. 8th Intl. Conf. on Information Knowledge Management*, 1999.
34. R. Swan, J. Allan, "Automatic generation of overview timelines," *Proc. SIGIR Intl. Conf. on Information Retrieval*, 2000.
35. R. Swan, D. Jensen, "TimeMines: Constructing Timelines with Statistical Models of Word Usage," *KDD-2000 Workshop on Text Mining*, 2000.
36. M. Vlachos, C. Meek, Z. Vagena, D. Gunopulos. "Identifying Similarities, Periodicities and Bursts for Online Search Queries." *Proc. ACM SIGMOD International Conference on Management of Data*, 2004.
37. P. Wong, W. Cowley, H. Foote, E. Jurrus, J. Thomas, "Visualizing sequential patterns for text mining," *Proc. IEEE Information Visualization*, 2000
38. Y. Zhu and D. Shasha. "Efficient Elastic Burst Detection in Data Streams," Proc. ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2003.