DE GRUYTER
OPEN

Int. J. Appl. Math. Comput. Sci., 2014, Vol. 24, No. 1, 199–212
DOI: 10.2478/amcs-2014-0015

amcs

SURVIVAL ANALYSIS ON DATA STREAMS: ANALYZING TEMPORAL EVENTS IN DYNAMICALLY CHANGING ENVIRONMENTS

AMMAR SHAKER, EYKE HÜLLERMEIER

Department of Mathematics and Computer Science
University of Marburg, Hans-Meerwein-Straße, 35032 Marburg, Germany
e-mail: {shaker,eyke}@mathematik.uni-marburg.de

In this paper, we introduce a method for survival analysis on data streams. Survival analysis (also known as event history analysis) is an established statistical method for the study of temporal “events” or, more specifically, questions regarding the temporal distribution of the occurrence of events and their dependence on covariates of the data sources. To make this method applicable in the setting of data streams, we propose an adaptive variant of a model that is closely related to the well-known Cox proportional hazard model. Adopting a sliding window approach, our method continuously updates its parameters based on the event data in the current time window. As a proof of concept, we present two case studies in which our method is used for different types of spatio-temporal data analysis, namely, the analysis of earthquake data and Twitter data. In an attempt to explain the frequency of events by the spatial location of the data source, both studies use the location as covariates of the sources.

Keywords: data streams, survival analysis, event history analysis, earthquake data, Twitter data.

1. Introduction

The so-called *data streams* have recently attracted increasing attention in various fields of theoretical, methodological and applied computer science, such as database systems, data mining, and distributed systems. As the notion suggests, a data stream can roughly be thought of as an ordered sequence of data items that arrive continuously as time progresses (Golab and Tamer, 2003; Garofalakis *et al.*, 2002; Das *et al.*, 2003). Streams of that kind are naturally produced in various applications, for example, network monitoring, telecommunication systems, customer click streams, stock markets, or any type of multi-sensor system.

A data stream system may constantly produce huge amounts of data. As an illustration, imagine a multi-sensor system with 10,000 sensors, each of which sending a measurement every second of time. Regarding data storage, management and processing, the continuous arrival of data items in multiple, rapid, time-varying, and potentially unbounded streams raises new challenges and research problems. Indeed, it is usually not feasible to simply store the arriving data in a traditional database management system in order to perform operations on those data later on. Rather, stream data must generally be processed in an online manner in order to guarantee that results are up-to-date and that queries can be answered with small time delay. The development of corresponding *stream processing systems* is a topic of active research (Cherniack *et al.*, 2003; Krizanovic *et al.*, 2011).

The remarks on data stream processing in general also apply to the analysis of stream data in particular. In fact, mining data streams and learning from data streams have been topics of active research in recent years (Gaber *et al.*, 2005; Gama and Gaber, 2007; Gama, 2012). Roughly speaking, the key motivation behind these and related fields is the idea of a system that learns incrementally, and maybe even in real time, on a continuous stream of data, and which is able to properly adapt itself to changes of environmental conditions or properties of the data-generating process. Systems with these properties have already been developed for different machine learning and data mining problems, such as clustering (Aggarwal *et al.*, 2003; Beringer and Hüllermeier, 2006; Oliveira and Gama, 2012), classification (Hulten *et al.*, 2001; Ikonomovska *et al.*, 2011), and frequent (sequential) pattern mining (Cormode and Muthukrishnan, 2005; Chen *et al.*, 2005).

In this paper, we address another data analysis problem in the context of data streams, namely, the analysis of temporal “events” or, more specifically, questions regarding the temporal distribution of (duration between) the occurrence of events and their dependence on covariates of the data sources. To this end, we develop an incremental, adaptive version of *survival analysis*, which is a standard statistical method for event analysis. The basic mathematical tool in survival analysis is the *hazard function*, which models the “propensity” of the occurrence of an event (marginal probability of an event conditional to no event so far) as a function of time.

Connections between survival analysis on the one hand and machine learning and data mining on the other have already been established by some authors (see, for example, Zupan *et al.*, 2000; Amati *et al.*, 2012). To the best of our knowledge, however, survival analysis has not been considered in the data stream setting so far. This is arguably surprising, for several reasons. Most notably, the temporal nature of event data naturally fits the data stream model, and indeed, “event data” are naturally produced by many data sources. Moreover, survival analysis is widely applicable and routinely used in many application fields. In fact, survival analysis, a term commonly used in medicine, is also known as event history analysis in sociology, reliability analysis in engineering and duration analysis in economics. Although “survival analysis” seems to be most widely used, we shall adopt the term “Event History Analysis” (EHA) in the remainder of this paper, simply because this term is more neutral and less associated with a specific application.

To make event history analysis applicable in the setting of data streams, we develop an adaptive (online) variant of a model that is closely related to the well-known proportional hazard model proposed by Cox and Oakes (1984). In this model, the hazard rate may depend on one or more covariates associated with a statistical entity. More specifically, in the proportional hazard model, the effect of an increase in a covariate by one unit is multiplicative with respect to the hazard rate.

We adopt a sliding window approach, which is a common technique in data stream analysis. In order to estimate the influence of the covariates, we assume the hazard rate to be constant on the current window. The estimate then depends on the frequency and temporal distribution of events falling inside the window, and sliding the window calls for adapting the estimate in an incremental (and as efficient as possible) manner.

The remainder of the paper is organized as follows. As a background, we recall some basic information about data streams in Section 2 and on event history analysis in Section 3. Section 4 is devoted to our extension of EHA and describes the main adaptations that we realized to make this method applicable in a streaming setting. Finally, to evaluate our approach, we present two case studies that are meant as a proof of principle. In both studies, our method is used for a specific type of spatio-temporal data analysis, namely, the analysis of earthquake data (Section 5) and of Twitter data (Section 6).

2. Data stream model

The *data stream model* assumes that input data are not available for random access from disk or memory, such as relations in standard relational databases, but rather arrive in the form of one or more continuous data streams. The stream model differs from the standard relational model in the following ways (Babcock *et al.*, 2002):

- The elements of a stream arrive incrementally in an “online” manner. That is, the stream is “active” in the sense that the incoming items trigger operations on the data rather than being sent on request.

- The order in which elements of a stream arrive are not under the control of the system.

- Data streams are potentially of unbounded size.

- Data stream elements that have been processed are either discarded or archived. They cannot be retrieved easily unless being stored in memory, which is typically small relative to the size of the stream.[1]

- Due to limited resources (memory) and strict time constraints, the computation of exact results will usually not be possible. Therefore, the processing of stream data does commonly produce *approximate* results (Considine *et al.*, 2004).

For data mining and machine learning methods, the above properties have a number of important consequences and come with several challenges. In particular, the standard “batch mode” of learning, in which the entire data as a whole is provided as an input to the learning algorithm, is no longer practicable. Correspondingly, the learner is not allowed to make several passes through the data set, which is commonly done by standard methods in statistics and machine learning. Instead, the data must be processed in a single pass, which implies an incremental mode of learning and model adaptation.

[1] Stored/condensed information about past data is often referred to as a *synopsis*.

Domingos and Hulten (2003) list a number of properties that an ideal stream mining system should exhibit, and suggest corresponding design decisions: the system uses only a limited amount of memory; the time to process a single record is short and ideally constant; the data are volatile and a single data record accessed only once; the model produced in an incremental way is equivalent to the one that would have been obtained through common batch learning (on all data records so far); the learning algorithm should react to concept drift (i.e., any change of the underlying data-generating process) in a proper way and maintain a model that always reflects the current concept.

3. Event history analysis

Event history analysis is a statistical method for modeling and analyzing the temporal distribution of events in the course of time or, more specifically, the duration before the occurrence of an event; the notion of an "event" is completely generic and may indicate, for example, the failure of an electrical device. The method is perhaps even better known as "survival analysis", a term that originates from applications in medicine, in which an event is the death of a patient and the *survival time* the time period $s = t_{\text{event}} - t_{\text{start}}$ between the beginning of the study and the occurrence of this event.

Thus, the basic statistical entities in EHA are subjects, typically described in terms of feature vectors $\boldsymbol{x} \in \mathbb{R}^n$, together with their survival time s. The goal, then, is to model the dependence of s on $\boldsymbol{x}$. In principle, one may thus be tempted to approach this task as a standard regression problem with input (regressor) $\boldsymbol{x}$ and output (response) s.

However, the survival time s is normally not observed for all subjects. Indeed, the problem of *censoring* plays an important role in EHA and occurs in different facets. In particular, it may happen that some of the subjects are still under observation when the study ends at time t_{end}; in other words, these subjects have survived till the end of the study. They are censored or, more specifically, *right censored*, since t_{event} has not been observed for them; instead, it is only known that $t_{\text{event}} > t_{\text{end}}$. Another reason for censoring could be that a subject leaves the study, not since the event of interest occurred, but simply for other reasons (for example, a patient in a breast cancer study may die from a car accident).

3.1. Survival function and hazard rate.

Suppose the time for an event to occur is modeled as a real-valued random variable T with probability density function $f(\cdot)$. Moreover, denote the cumulative distribution function by $F(\cdot)$, i.e.,

$$F(t) \,=\, \mathbf{P}\{T \le t\} = \int_0^t f(x)\,\mathrm{d}x$$

is the probability of an event to occur before the time t. The survival function $S(\cdot)$ is then defined as

$$S(t) \,=\, \mathbf{P}\{T > t\} = 1 - F(t) = \int_t^\infty f(x)\,\mathrm{d}x. \quad (1)$$

Since $S(t)$ is the probability that the event did not occur until time t, it can be used to model the probability of an event that is right censored.

The hazard function or hazard rate $h(\cdot)$ is defined as follows:

$$\begin{aligned} h(t) &= \lim_{\mathrm{d}t \to 0} \frac{\mathbf{P}\{t < T \le t + \mathrm{d}t \,|\, T > t\}}{\mathrm{d}t} \quad (2) \\ &= \frac{f(t)}{S(t)}. \end{aligned}$$

Roughly speaking, $h(t)$ is the conditional probability that the event will occur within a small time interval after t, given that it has not occurred until t. More specifically, $h(t)$ is the limit of this probability when the length of the time interval tends to 0. Mathematically, it is hence a kind of density (and not a probability) function, which means that it may thoroughly assume values larger than 1. Note that the density $f(\cdot)$ can be recovered from the hazard rate and the survival function, since

$$f(t) \,\equiv\, h(t) \cdot S(t).$$

3.2. Modeling the hazard rate.

Since a statistical entity is not always a person, we shall subsequently use the more neutral term "instance" instead of "subject". Suppose such an instance to be described in terms of a feature vector

$$\boldsymbol{x} = (x_1, \ldots, x_n)^\top \in \mathbb{R}^n, \quad (3)$$

where x_i is the i-th property of the instance (for example, the age of a patient in a medical study). Assuming the hazard rate for this instance to depend not only on time but also on the properties (features) x_i, it can be written as $h = h(\boldsymbol{x}, t)$.

Often, the hazard rate is even assumed to be constant over time, in which case it only depends on $\boldsymbol{x}$ but not on t. In this case, we shall also denote it by $\lambda = \lambda(\boldsymbol{x})$. Note that a constant hazard rate gives rise to an exponential survival function:

$$S(t) \,=\, \exp(-\lambda t).$$

In the Cox proportional hazard model (Cox and Oakes, 1984), the hazard rate is modeled as a log-linear function of the features x_i:

$$\begin{aligned}\lambda(\boldsymbol{x}) &= \alpha_0 \cdot \exp\left(\boldsymbol{x}^\top \beta\right) \quad (4)\\ &= \alpha_0 \cdot \exp\left(\sum_{i=1}^{n} \beta_i x_i\right).\end{aligned}$$

In this context, the x_i are also called *covariates*. Extending the covariate vector $\boldsymbol{x}$ in (3) by a constant entry $x_0 \equiv 1$, (4) can be written more compactly as

$$\lambda(\boldsymbol{x}) = \exp\left(\boldsymbol{x}^\top \beta\right), \quad (5)$$

with $\beta_0 = \log(\alpha_0)$. As can be seen, according to the above model, the effect of an increase in a covariate by one unit is multiplicative with respect to the hazard rate; or, stated differently, the hazard rate is proportional to each covariate: increasing x_i by one unit increases $\lambda(\boldsymbol{x})$ by a factor of $\alpha_i = \exp(\beta_i)$.

Statistical methods for event history analysis, such as Cox regression (Cox, 1972), provide estimates of the model parameters β_i and, therefore, of the hazard rate itself. The latter can be used, for example, for prediction purposes: given an estimate of the hazard rate, one can predict the time span till the next event will occur, both in terms of point predictions (e.g., the expected survival time of a patient) and confidence sets (e.g., a confidence interval for the survival time). At least as interesting as the hazard rate itself, however, are the estimates of the parameters β_i, which inform about the influence of different covariates on the hazard rate. For example, if $\beta_i = \log(2)$ is the parameter modeling the influence of the covariate `smoking` (a binary attribute with value 1 if the patient is a smoker and 0 otherwise) in a medical study, this means that—under the model (5) and *ceterus paribus*, i.e., all other covariates being equal—smoking doubles the hazard rate and therefore halves the expected survival time.

Before proceeding, let us note that non-constant hazard functions $h(\boldsymbol{x}, t)$, in which the rate does not only depend on covariates $\boldsymbol{x}$, but also changes with time t, have been studied extensively in the statistical literature, and many parameterized families of functions have been proposed for modeling the influence of time on the rate (Cox and Oakes, 1984). As will become clear later on, however, the constant model $\lambda(\boldsymbol{x})$ is sufficient for our purpose, or at least provides a sufficiently good approximation. This is due to the use of a sliding window approach: roughly speaking, the assumption of a constant rate does not refer to a data stream as a whole but only to the current time window; therefore, by sliding the window, the hazard rate may actually vary in the course of time, too. Overall, our model thus even becomes very flexible, especially since time-dependence is modeled in a non-parametric way.[2]

4. Event history analysis on data streams

Our setting assumes a fixed set of J data streams to be given, each of which corresponds to an instance $\boldsymbol{x}$ characterized in terms of a vector of covariates $(x_1, \ldots, x_n)$. Moreover, each stream produces a sequence of temporal events, i.e., events that are associated with a unique time of occurrence; (see Fig. 1 for an illustration). For simplicity, we assume the underlying time scale to be discrete, i.e., time progresses in discrete steps (such as seconds or minutes).

As an example, imagine that each stream corresponds to a book offered by an online book store, and the covariates are properties of the book (price, genre, etc.). Moreover, an “event” occurs whenever a client is purchasing a book. The hazard rate associated with a book can then be interpreted as a measure of the propensity of people to buy this book. Obviously, this propensity will change in the course of time, and for each book; therefore, it is interesting to monitor the evolution of its hazard rate. Apart from that, it is interesting to figure out the influence of the covariates on the buying behavior of the clients and, perhaps even more importantly, how this influence changes over time. One may expect, for example, that the price of a book will become more important, and will hence have a stronger influence on the hazard rates of all books, in times of an economic crisis.

The previous example has made clear that, when looking at a single data stream, we are interested in events that can occur repeatedly (for the same instance $\boldsymbol{x}$) in the course of time. Events of that kind are called *recurrent events* and need to be distinguished from those that can occur at most once (like the death of a patient in a medical study). More specifically, we are interested in the time duration between the occurrence of two events. For a fixed instance (data stream) $\boldsymbol{x}$, suppose the hazard rate $\lambda = \lambda(\boldsymbol{x})$ to be constant, and let

$$t_1 < t_2 < \cdots < t_k$$

denote the time points at which an event has been observed for this instance; moreover, let $a = t_0 < t_1$ and $b = t_{k+1} > t_k$ denote the start and the end of the observation interval $[a, b]$. The probability of the

[2] To some extent, this is comparable with statistical methods like kernel density estimation or locally weighted linear regression.

observation sequence $\mathcal{T}(\boldsymbol{x}) = \{t_\tau\}_{\tau=1}^{k}$ is then given by

$$\begin{aligned}\mathbf{P}(\mathcal{T}(\boldsymbol{x})) &= \left(\prod_{\tau=1}^{k} f(t_{\tau-1}, t_\tau)\right) \cdot S(t_k, t_{k+1}) \qquad (6)\\ &= \lambda(\boldsymbol{x})^k \cdot \prod_{\tau=1}^{k+1} \exp\big(-\lambda(\boldsymbol{x})(t_\tau - t_{\tau-1})\big),\end{aligned}$$

where

$$\begin{aligned}f(t', t) &= h(t) \cdot S(t', t)\\ &= \lambda(\boldsymbol{x}) \cdot \exp\big(-\lambda(\boldsymbol{x})(t - t')\big)\end{aligned}$$

is the probability that an event occurs at time t if the observation starts at time t'.

4.1. Left censoring.

More generally, suppose that the observation of the first event started at an unobserved time t prior to the start of the observation window at time t_0; this is a situation of *left censoring* that we are facing in our sliding window approach to be detailed below. The probability to observe the duration from t_0 to t_1 is then given by the *conditional* probability of the event at time t_1 given survival until t_0, i.e., by the expression

$$\begin{aligned}f(t_0, t_1) = \frac{f(t, t_1)}{S(t, t_0)} &= \frac{\lambda(\boldsymbol{x}) \cdot \exp\big(-\lambda(\boldsymbol{x})(t_1 - t)\big)}{\exp\big(-\lambda(\boldsymbol{x})(t_0 - t)\big)}\\ &= \lambda(\boldsymbol{x}) \cdot \exp\big(-\lambda(\boldsymbol{x})(t_1 - t_0)\big).\end{aligned}$$

Thus, we eventually obtain the same expression (6). Roughly speaking, this is due to the fact that a process with a constant hazard rate is "memoryless".

4.2. Parallel event sequences.

In our setting, we assume to observe a sequence of recurrent events $\mathcal{T}(\boldsymbol{x}) = \{t_\tau\}_{\tau=1}^{k}$ not only for a single instance $\boldsymbol{x}$, but for a fixed set of J instances $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_J\}$, with $\boldsymbol{x}_j = (x_1^j, \dots, x_n^j)^\top$. Thus, the data relevant to a time window $[a, b]$ are given in the form of J parallel event sequences:

$$\begin{aligned}\mathcal{D} &= \Big(\mathcal{T}(\boldsymbol{x}_1), \dots, \mathcal{T}(\boldsymbol{x}_J)\Big) \qquad (7)\\ &= \Big(\{t_\tau^1\}_{\tau=1}^{k_1}, \dots, \{t_\tau^J\}_{\tau=1}^{k_J}\Big),\end{aligned}$$

where k_j is the number of events for $\boldsymbol{x}_j$ and $\{t_\tau^j\}_{\tau=1}^{k_j}$ the corresponding time points. Assuming independence, the probability of $\mathcal{D}$ is

$$\begin{aligned}&\mathbf{P}(\mathcal{D})\\ &= \prod_{j=1}^{J} \mathbf{P}(\mathcal{T}(\boldsymbol{x}_j))\\ &= \prod_{j=1}^{J} \left[\lambda(\boldsymbol{x}_j)^{k_j} \prod_{\tau=1}^{k_j+1} \exp\Big(-\lambda(\boldsymbol{x}_j)\left(t_\tau^j - t_{\tau-1}^j\right)\Big)\right],\end{aligned}$$

and the logarithm of this probability is

$$\begin{aligned}&\log\left(\prod_{j=1}^{J} \mathbf{P}(\mathcal{T}(\boldsymbol{x}_j))\right)\\ &\quad = \sum_{j=1}^{J} \left[k_j \log\left(\lambda(\boldsymbol{x}_j)\right) - \sum_{\tau=1}^{k_j+1} \lambda(\boldsymbol{x}_j)\left(t_\tau^j - t_{\tau-1}^j\right)\right].\end{aligned}$$

For the model (5), this expression yields the following log-likelihood function for the parameter vector β:

$$\begin{aligned}\ell(\beta) &= \sum_{j=1}^{J} \Bigg[k_j\beta_0 + k_j\left(\sum_{i=1}^{n} \beta_i x_i^j\right)\\ &\qquad - \sum_{\tau=1}^{k_j+1} \beta_0 \exp\left(\sum_{i=1}^{n} \beta_i x_i^j\right)\left(t_\tau^j - t_{\tau-1}^j\right)\Bigg]\\ &= \sum_{j=1}^{J} \Bigg[k_j\beta_0 + k_j\left(\sum_{i=1}^{n} \beta_i x_i^j\right)\\ &\qquad - \beta_0 \exp\left(\sum_{i=1}^{n} \beta_i x_i^j\right)(b - a)\Bigg].\end{aligned}$$

4.3. Adaptive ML estimation.

Parameter estimation on a time window $[a, b]$ can now be done by means of Maximum Likelihood (ML) estimation, i.e., by finding the maximizer of the above likelihood function:

$$\beta^* = (\beta_0^*, \beta_1^*, \dots, \beta_n^*) \;=\; \arg\max \ell(\beta).$$

Unfortunately, there is no analytical expression for β^*, so that the estimator needs to be found by means of numerical optimization procedures. Nevertheless, since the log-likelihood function $\ell(\beta)$ is concave (which can be shown by checking corresponding conditions on the second derivatives), simple gradient-based optimization techniques and online versions of gradient ascent (Bottou, 1998) can be applied and in fact turned out to work rather well.

The use of local optimization techniques is also reasonable as it can be turned quite naturally into an incremental learning algorithm applicable in our streaming setting. Recall that we slide a time window of fixed length w along the time axis. More specifically, the window is repeatedly moved in discrete steps, each time replacing the current window $W_t = [t, t + w]$ by the shifted one $W_{t+\Delta t} = [t + \Delta t, t + w + \Delta t]$. A shift of that kind will of course also change the parallel event sequences (7) associated with the current time window and, therefore, necessitate a re-estimation of the parameter vector β.

Typically, however, the event sequences $\mathcal{T}(\boldsymbol{x}_j)$ will change but slightly, since most of the current events

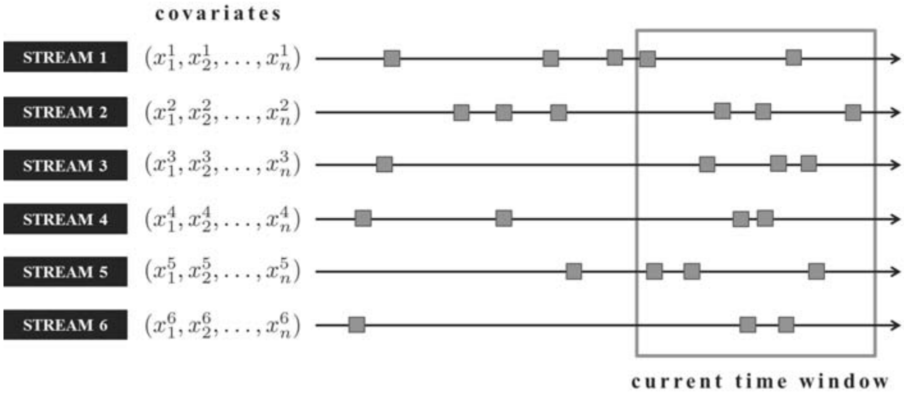

covariates
STREAM 1 $(x_1^1, x_2^1, \dots, x_n^1)$
STREAM 2 $(x_1^2, x_2^2, \dots, x_n^2)$
STREAM 3 $(x_1^3, x_2^3, \dots, x_n^3)$
STREAM 4 $(x_1^4, x_2^4, \dots, x_n^4)$
STREAM 5 $(x_1^5, x_2^5, \dots, x_n^5)$
STREAM 6 $(x_1^6, x_2^6, \dots, x_n^6)$
current time window

Fig. 1. Illustration of our setting consisting of a set of J (here $J = 6$) parallel data streams: every stream corresponds to a statistical entity characterized in terms of a vector of covariates. Moreover, each stream produces a sequence of temporal events (marked by solid squares). A sliding window (indicated by the grey box) is masking outdated events that occurred in the past.

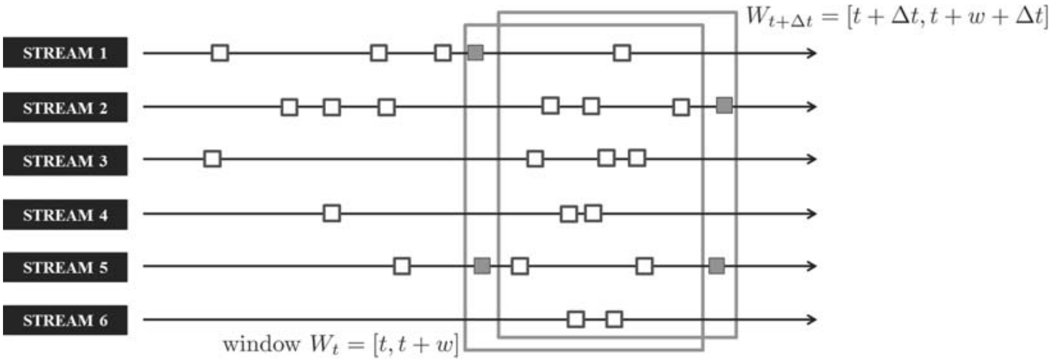

$W_{t+\Delta t} = [t + \Delta t, t + w + \Delta t]$
STREAM 1
STREAM 2
STREAM 3
STREAM 4
STREAM 5
STREAM 6
window $W_t = [t, t + w]$

Fig. 2. Illustration of the shift of the time window: the current window $W_t = [t, t + w]$ is replaced by the new one $W_{t+\Delta t} = [t + \Delta t, t + w + \Delta t]$. While the status of some of the events changes (filled boxes), the status of the others (non-filled boxes) remains the same (either outdated or active).

t_τ^j will remain inside the window—only those close to the lower boundary t will fall out (namely, those with $t \leq t_\tau^j < t + \Delta t$), while new events observed between $t + w$ and $t + w + \Delta t$ will be added (see Fig. 2 for an illustration). In any case, the new ML estimate of β will normally be found in close proximity to the old one. Therefore, the current estimate β_t^*, i.e., the ML estimate for the current time window $W_t = [t, t + w]$, will provide a good initial solution for the re-estimation problem to be solved by our gradient-based optimizer. Indeed, in practical experiments, we found that only a few adaptation steps are generally needed to reach the new ML estimate $\beta_{t+\Delta t}^*$ (with sufficient accuracy).

As mentioned before, what our adaptive estimation procedure eventually produces is a sequence of parameter estimates that (implicitly) represents the evolution of both the parameter β and the hazard rates $\lambda_j = \lambda(\boldsymbol{x}_j)$ over time. More specifically, for a fixed time point τ, let $\mathcal{W}_\tau$ denote the set of all time windows covering this time point:

$$\mathcal{W}_\tau = \big\{ W_t \,|\, \tau \in [t, t + \Delta t] \big\}.$$

Moreover, let β_t^* denote the ML estimation of β on W_t. Then, we define the parameter β at time τ by averaging:

$$\beta(\tau) = \frac{1}{|\mathcal{W}_\tau|} \sum_{W_t \in \mathcal{W}_\tau} \beta_t^* . \tag{8}$$

Correspondingly, the hazard rate λ_j at time τ is given by

$$\lambda(\tau) = \exp\big(\boldsymbol{x}_j^\top \beta(\tau)\big). \tag{9}$$

5. Case study: Earthquake analysis

As a proof of concept, we conducted two case studies in which our streaming version of EHA is used for spatio-temporal data analysis. While the temporal aspect is naturally captured by the hazard rate model, the spatial aspect is incorporated through the use of spatial information as covariates of the data streams. In other words, the vector (3) of covariates describes the spatial location of a data source.

In our first study, we apply our method to the analysis of earthquake data. The data is collected from the USGS[3] (United States Geological Survey), specifically from the catalog of the NEIC[4] (National Earthquake Information Center), whose mission is to quickly discover the most recent destructive earthquakes in terms of location and magnitude and to broadcast this information to international agencies and scientist.

5.1. Data generation. The earthquakes were collected in the time period between January 1, 2000 and the end of March 27, 2012. Since entries in the USGS/NEIC catalog can be added or modified at any time, we stick to the data in the catalog at the time of exportation, namely, April 12, 2012. Table 1 presents an example of earthquake data (a list of 5 earthquakes with their occurrence times and attributes). The online catalog of the USGS/NEIC retains only significant earthquakes with a magnitude bigger than 2.5, though very few micro-earthquakes (with a magnitude less than 1) could be found (and even a few earthquakes with missing magnitude). In total, we collected 319,884 earthquakes around the globe.

Every earthquake is identified by its geographic coordinate, the exact time of occurrence (up to the second), and the magnitude and depth. Figure 3(a) shows a picture of the collected earthquakes, plotted as dots at the place of their rounded geographic location.

Recall that, in our setting introduced in Section 3, we assume to observe event sequences for a fixed set of instances. In order to define these instances, we discretize the globe both in terms of longitude and latitude, and associate one instance with each intersection point. More specifically, with $\phi \in \{-90, -89, \ldots, 90\}$ for longitude and $\eta \in \{-180, -179, \ldots, 180\}$ for latitude, we end up with $181 \times 361 = 65,341$ instances in total. The regions thus produced are obviously not equal in size: since latitudes are not parallel like longitudes, areas near the equator are smaller than those closer to the poles.

Furthermore, recall that each instance is described in terms of features (covariates) x_i, which, according to (5), have a proportional effect on the hazard rate. In order to account for possibly nonlinear dependencies between spatial coordinates and risk of earthquake, we define these features in terms of a fuzzy partition, that is, a partition defined in terms of fuzzy sets (Zadeh, 1965). In contrast to a standard partition defined in terms of intervals, this allows a smooth transition between spatial regions. More specifically, we discretize both longitude and latitude by means of triangular fuzzy sets as shown in Fig. 3(c). The number of fuzzy sets was chosen so as to achieve a reasonable compromise between spatial resolution and computational complexity. A two-dimensional (fuzzy) discretization of the globe is defined in terms of the Cartesian product of these two one-dimensional discretizations, using the minimum operator for fuzzy set intersection. The covariates of an instance $\boldsymbol{x}$ associated with coordinates (ϕ, η) are then simply given by the membership degrees in all these two-dimensional fuzzy sets, i.e., the covariates are of the form

$$x_{i,j} = \min\Big(A_i(\phi), B_j(\eta)\Big),$$

where A_i is one of the 11 fuzzy sets for longitude and B_j one of the 10 fuzzy sets for latitude; thus, each instance is of the form

$$\boldsymbol{x} = \big(x_{1,1}, x_{1,2}, \ldots, x_{1,10}, \ldots, x_{11,10}\big) \in [0,1]^{110}.$$

5.2. Results. Given the data produced in this way and after sorting all earthquakes by their time of occurrence, we are able to apply our method as outlined in Section 3. We set the length of the time window to 3 months and the shift parameter Δ to 1 week. These values appear to be reasonable for this application, although they are not optimized according to any specific criterion.

The results we obtain in terms of time-dependent estimates of the parameters $\beta_{i,j}$, each of which is associated with a covariate $x_{i,j}$ and hence with a spatial (fuzzy) region $A_i \times B_j$, appear to be quite plausible. In fact, several interesting observations could be made for data from the last decade. For example, as can be seen in Fig. 4(a), the occurrence of Sichuan's earthquake in April 2008 comes with a significant increase in the coefficients of the fuzzy sets covering that area: the *FS_83* line increases steeply a few weeks before the shock, which caused about 80,0000 causalities.

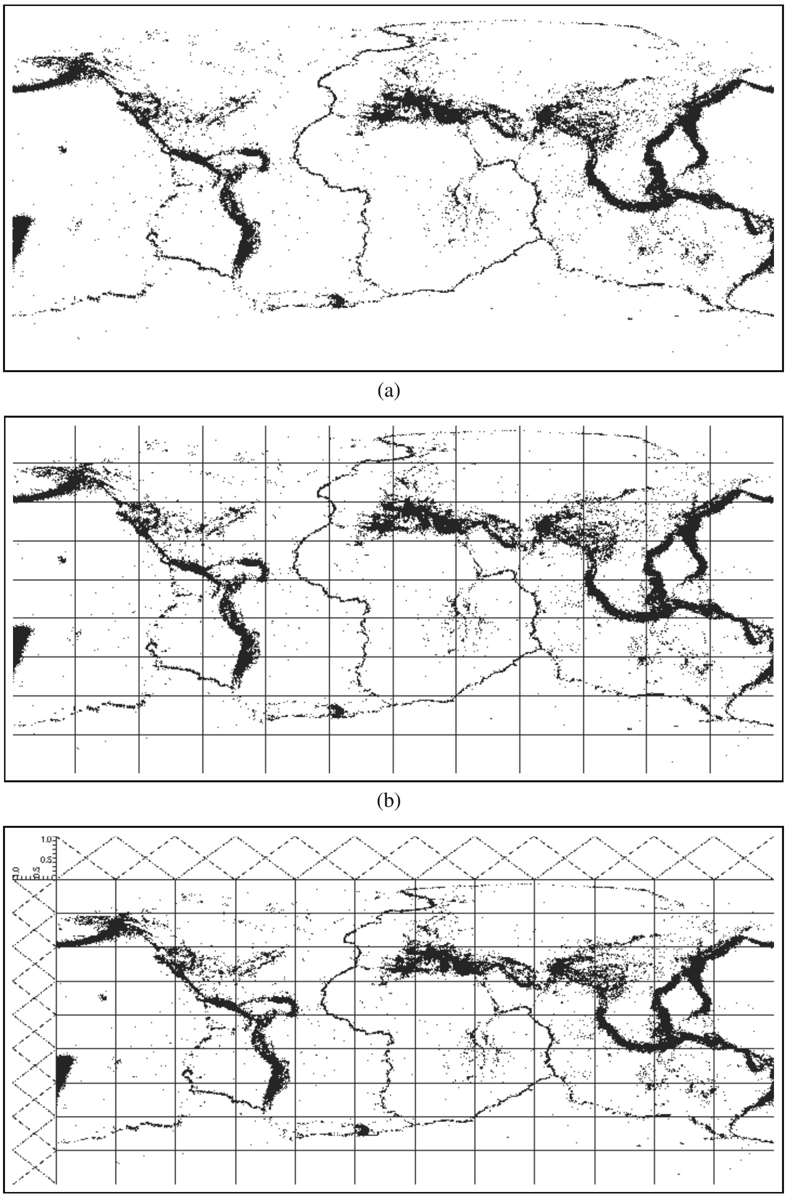
The same can be noticed in Fig.4(b) before Tohoku's earthquake in March 2011, whose location was (38.32°N,142.36°E). The coefficient of the *FS_85* line increases by a factor of 4 till few hours before the earthquake, indicating an increased hazard rate for the (fuzzy) area around $(\phi, \eta) = (120, 120)$. Another interesting observation is the increasing estimated hazard in the epicenter of both earthquakes, as shown in Fig. 5.

[3] http://www.usgs.gov/.

[4] http://earthquake.usgs.gov/regional/neic/.

Table 1. Example of earthquake data: 5 earthquakes that occurred on the first day of 2012.

Year	Month	Day	UTC Time hhmmss.mm	Latitude	Longitude	Mag.	Depth	Catalog
2012	01	01	003008.77	12.008	143.487	5.1	35	PDE-W
2012	01	01	003725.28	63.337	-147.516	3.0	65	PDE-W
2012	01	01	004342.77	12.014	143.536	4.4	35	PDE-W
2012	01	01	005008.04	-11.366	166.218	5.3	67	PDE-W
2012	01	01	012207.66	-6.747	130.007	4.2	145	PDE-W

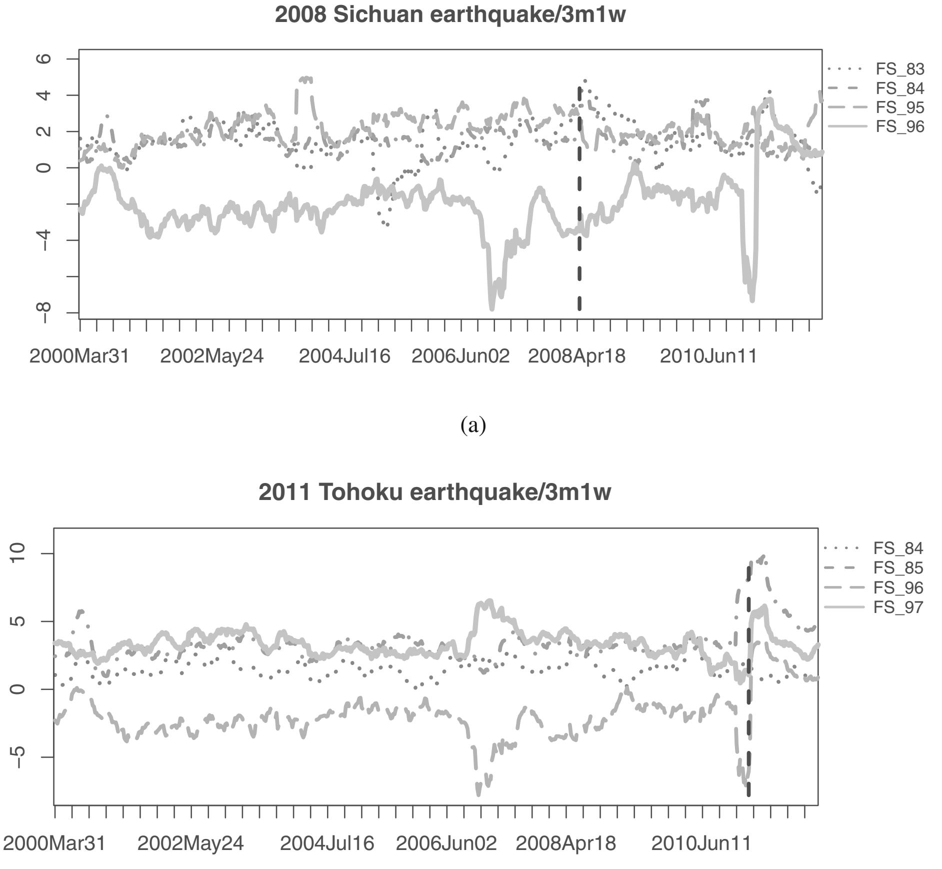


(a)

(b)

(c)

Fig. 3. Collected dataset of earthquakes, plotted by their geographic coordinates. The data contain earthquakes between January 1, 2000 till the end of March 27, 2012: earthquakes only (a), with latitude and longitude lines added (b), fuzzy partitions on the two coordinates (c).



(a)

(b)

Fig. 4. Parameters for those areas with significant earthquakes in 2008 and 2011.

6. Case study: Twitter data

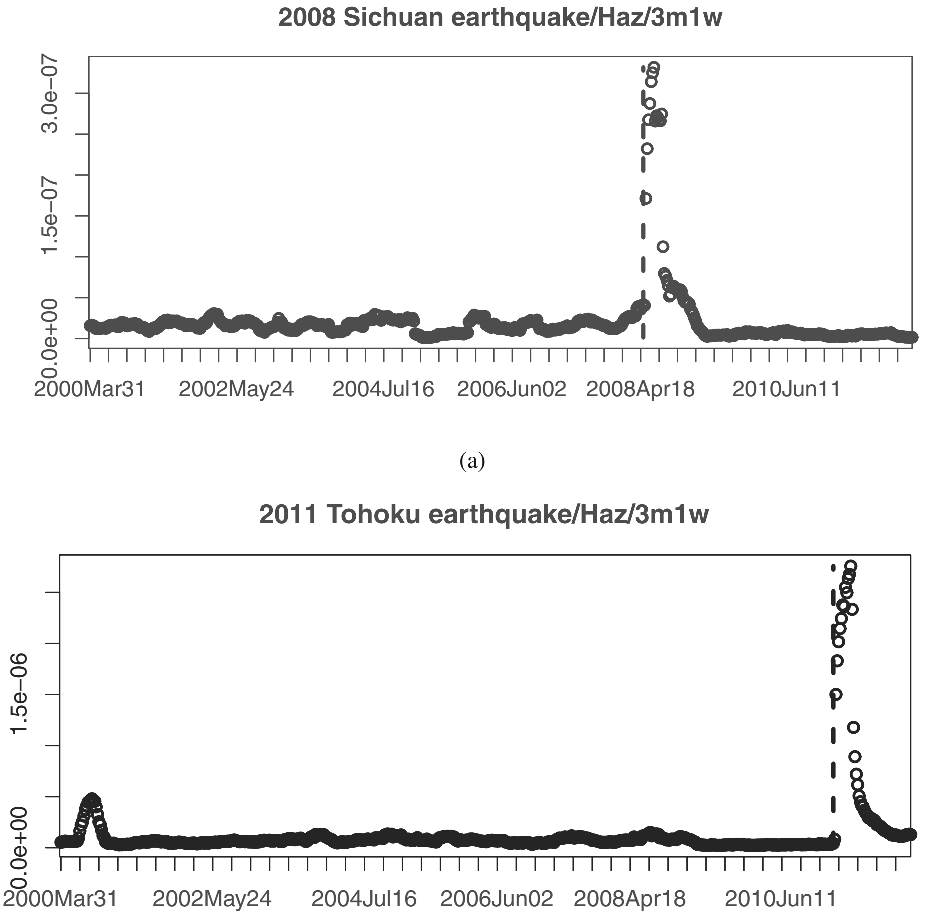
Our second case study is based on data collected from Twitter[5], which is an online microblogging web site. Twitter is a service that allows users to send short messages of up to 140 characters known as “tweets”. Every tweet is attributed by some meta data, including the ID of the user who wrote it and the time the tweet was sent. Further attributes can be extracted from the tweet with the permission of the user. Those attributes indicate the *geolocation* of the user when they posted their tweet; this is supported by a GPS (Global Positioning System) functionality embedded in a mobile device or a tablet PC. The geolocation is represented as a pair $(\texttt{lo}, \texttt{la})$ with an entry for the longitude and for the latitude. Table 2 shows an example of Twitter data written in the Json[6] format.

We collected tweets generated inside the bounding box of Germany, which is determined by the corner points $(\texttt{lo}, \texttt{la}) = (5.53, 47.16)$ and $(15.2, 55.03)$. This data were collected during a timespan of about two months, namely, between March 20, 2012 and May 27, 2012. In total, we collected about 4.9 million tweets coming mainly from Germany and its surrounding countries (Denmark, Poland, Czech Republic, Austria, Switzerland, France, Belgium and the Netherlands). Surprisingly, only 1.8 million of them were from inside Germany.

Like in the previous study on earthquakes, we apply a discretization on the area of Germany, considering every intersection point of the two coordinates with $\phi \in \{5.5, 5.6, \ldots, 15.2\}$ for longitude and $\eta \in \{47.1, 47.2, \ldots, 55.1\}$ for latitude, provided this intersection lies inside the borders of Germany. As a result, we end up with 5013 intersection points, which are considered the instances (data streams) $\boldsymbol{x}_j$ to be kept under observation. The next step is to find a proper representation of the instances in terms of covariates. Here, we decided to describe every instance by the

[5] http://www.twitter.com/.

[6] http://json.org/.



(a)

(b)

Fig. 5. Hazard values for those areas with significant earthquakes in 2008 and 2011.

normalized vector of Mahalanobis distances to the center of each of the 16 German states. Thus, each instance is represented in terms of the vector

$$\boldsymbol{x}_j = (x_1^j, \dots, x_{16}^j) \in \mathbb{R}^{16},$$

where x_i^j is the distance of the location $\boldsymbol{x}_j$ from the (center of the) i-th German state. By sorting the tweets according to their creation times, considering only those coming from Germany and assigning every tweet to the closest instance $\boldsymbol{x}_j$, we obtain a parallel stream of events that can again be processed by our method described in Section 3.

We fix the window size to 3 days and the shift parameter Δt to one day. As a result, we again obtain the time-dependent estimates (8) of the parameters $\beta_1, \dots, \beta_{16}$ associated with the 16 German states. Figure 6 shows how the estimated parameters change over time, compared with the base line hazard α_0 that is also plotted in each subfigure. An increasing parameter β_i can be interpreted as follows: the closer a location $\boldsymbol{x}_j$ to the corresponding state, the higher the hazard or, say, the propensity of users to send a tweet from that location.

As can be seen in Figs. 6(a) and (d), the parameter for the state Berlin is increasing in the time between May 2 and 5 while the parameter for the state Brandenburg is decreasing. Looking for an explanation for this observation, we found that the conference *re:publica*[7] took place during that time. This conference is a big meeting for bloggers from Germany and all around the world. Consequently, one can expect that more bloggers were in Berlin and less in the surrounding area, including the state of Brandenburg.

The opposite can be said about Saxony-Anhalt, which was seen as a gate for travelers, so its parameter was also increasing during that time. The parameters associated with the mentioned states are marked by the '$*$' symbol in Fig. 6. In a similar way, Fig. 6(d) shows how

[7] `http://re-publica.`

Table 2. Two examples of Twitter messages. Unimportant attributes are removed and some others are obfuscated, whereas important attributes are written in bold. Both messages are artificially created.

favorited:false, **text**:'Stau: A8 München Richtung Stuttgart 6 km zur Ausfahrt im Schneckentempo..', truncated:false, **created_at**:Fri Feb 10 10:38:47 +0000 2012, retweeted:false, retweet_count:0, coordinates:type:Point, **coordinates**:[9.55755, 48.6333], ..., entities:user_mentions:[], urls:[], hashtags:[], geo:type:Point, coordinates:[48.6333, 9.55755], ..., place:bounding_box:type:Polygon, coordinates:[[[9.534815, 48.616779], [9.594667, 48.616779], [9.594667, 48.640891], [9.534815, 48.640891]]], place_type:city, ..., country_code:DE, attributes:, full_name:Aichelberg, Göppingen, name:Aichelberg, id:29ef9f01a553e601, country:Germany, ..., id_str:###, user:default_profile:true, notifications:null, ..., time_zone:Berlin, created_at:Fri Sep 03 14:25:38 +0000 2010, verified:false, geo_enabled:true..., favourites_count:0, lang:de, ..., followers_count:335, ..., location:Karlsruhe, ..., name:###, ..., listed_count:21, following:null, screen_name:###, id:###, ..., statuses_count:10935, utc_offset:3600, friends_count:0, ..., id:###, ...
text:'top atmosphere in Weserstadion today, a very good match...', ..., **created_at**:Tue Apr 10 21:37:28 +0000 2012, place:bounding_box:type:Polygon, coordinates:[[[8.481599, 53.011035], [8.990593, 53.011035], [8.990593, 53.228969], [8.481599, 53.228969]]], country:Germany, attributes:, full_name:Bremen, Bremen, .., country_code:DE, name:Bremen, id:9467fbdc3cdbd2ef, place_type:city, coordinates:type:Point, **coordinates**:[8.837596, 53.06693] , retweeted:false, in_reply_to_status_id:null, ..., truncated:false, contributors:null, possibly_sensitive:false, in_reply_to_screen_name:null, favorited:false, user:default_profile:false, follow_request_sent:null, lang:de, friends_count:200, ..., is_translator:false, created_at:Sat May 23 13:01:45 +0000 2009, id_str:###, ..., url:null, following:null, verified:false, ..., location:Germany, ..., statuses_count:4537, ..., time_zone:Berlin, .., utc_offset:3600, followers_count:432, ..., id:###, retweet_count:0

the hazard associated with the state Schleswig-Holstein, marked by the '$+$' symbol, has increased on April 28. This was supposedly a direct effect of hosting a conference for a political German party, namely the *Piratenpartei*.

In a second experiment with the same stream of events, our aim was to observe changes between the city and the countryside. This was done by considering only instances located inside the states of Bremen and Lower Saxony[8]. Instances are now described only by a single binary covariate, indicating whether or not an instance is located in Bremen. Figure 7 shows how the corresponding parameter changes on a weekly basis. Interesting patterns can be observed especially for the weekends. First, there are normal weekends where people move from the condensed area of Bremen to the surrounding state, causing a decrease in the hazard (less Tweets sent from inside Bremen and more from outside); this pattern is marked by the '$+$' symbol. Second, we have weekends on which the local football club (Werder Bremen) hosted a football match in the German soccer league (Bundesliga), causing an increase in the hazard; this group is marked by '$*$' symbol.

To get an idea of the efficiency of the system, Fig. 8 provides a summary of the time required for a single adaptation step in the form of a histogram over runtimes. As can be seen, the adaptation time is in the range of seconds and upper bounded by one minute—compared to the shift length Δt of one day, this time is almost negligible. The peak of the distribution in the vicinity of zero suggests moreover that the adaptation effort is almost zero most of the time.

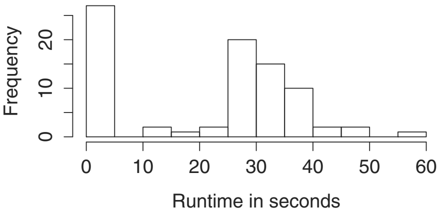


Fig. 8. Time needed for model adaptation.

7. Conclusion

In this paper, we introduced an adaptive approach to survival analysis (event history analysis) on data streams. To this end, we adopted a sliding window approach and proposed an adaptive (online) variant of a model that is closely related to the well-known Cox proportional hazard model. In this approach, maximum likelihood estimation of the model parameters is performed repeatedly, adapting the estimates whenever the time window has been shifted.

As a first proof of concept, we used our method for studying the occurrence of significant earthquakes during the last decade. Here, an event is an earthquake, and a

[8] Bremen is the smallest state in Germany, containing only two cities. It is surrounded by the larger state of Lower Saxony.

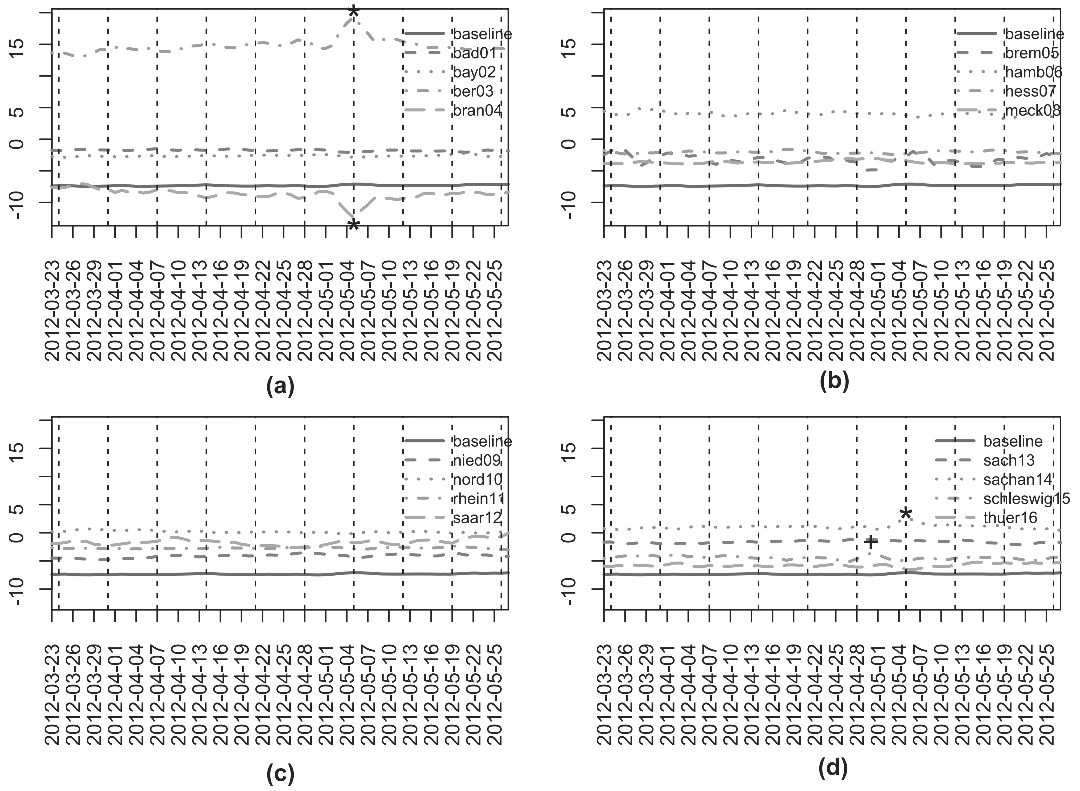


Fig. 6. Hazard rates for the 16 German states together with the base line hazard α_0, on a logarithmic scale.

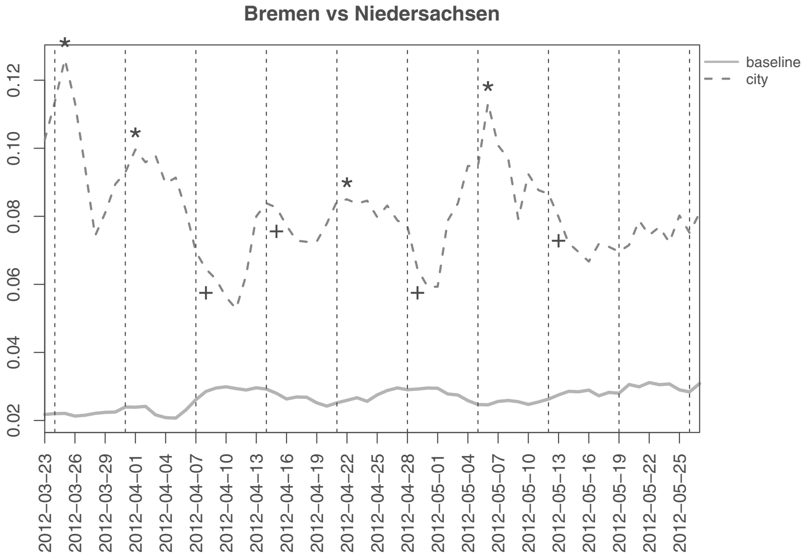


Fig. 7. Base line hazard and hazard rate for the distinction of the city of Bremen from the surrounding state of Lower Saxony.

statistical entity is a two-dimensional region on the globe characterized by its spatial coordinates; more specifically, we make use of fuzzy discretization techniques in order to capture the influence of the spatial location on the hazard rate in a flexible way. The results we obtain are plausible and agree with expectation. For a region such as Tohoku in Japan, one can observe a significant increase in the hazard rate prior to the disastrous earthquake in 2011. Similar observations can be made for other significant earthquakes such as Sichuan's in 2008. Plausible results could also be obtained in a second study using streams of almost 5 million Twitter messages. Interesting patterns or irregularities in the time-dependent parameter estimations of our hazard model could be explained by big events such as conferences or football matches.

Needless to say, our approach can be generalized and refined in various ways. So far, for example, we simply assumed the length of the sliding window to be fixed and predefined. However, noting that the window length should be chosen so as to achieve an optimal compromise between the availability of data (if the window is too short, it may not contain enough events) and the representativeness of the estimated hazard rate for the current time point (if the window is too long, it may cover outdated events that are no longer representative), appropriate means for dynamically adapting the length are clearly desirable.

In the stream-based approach to EHA as presented in this paper, like in event history analysis in general, event data are essentially assumed to be given. Or, stated differently, the production of these data is not considered part of the system; instead, events and the time points of their occurrence are provided by some external source. In many contemporary applications, however, the *detection* of an event can be seen as a non-trivial problem in itself. For example, suppose an event is defined as a Tweet reporting about a disaster or a crime (Li *et al.*, 2012). The production of an event sequence will then call for efficient text mining and analysis techniques. We consider the combination of our approach with methods for *event detection* (Yang *et al.*, 1998; Allan *et al.*, 1998; Weng and Lee, 2011; Sakaki *et al.*, 2013) as an interesting challenge for future work.

Another direction of future work is the use of our approach for *event prediction* on data streams. In fact, although the aspect of prediction has not been in the focus of this paper, we already mentioned that hazard rate models can be used for predicting the time span till the occurrence of the next event, both in terms of point predictions and confidence intervals. Thus, they may provide an interesting alternative to existing approaches to event prediction, which are based on other statistical and machine learning methods (Cheon *et al.*, 2009; Amodeo *et al.*, 2011; Radinsky and Horvitz, 2013).

References

Aggarwal, C.C., Han, J., Wang, J. and Yu, P.S. (2003). A framework for clustering evolving data streams, *Proceedings of the 29th International Conference on Very Large Data Bases, Berlin, Germany,* pp. 81–92.

Allan, J., Papka, R. and Lavrenko, V. (1998). On-line new event detection and tracking, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998), Melbourne, Australia*, pp. 37–45.

Amati, G., Amodeo, G. and Gaibisso, C. (2012). Survival analysis for freshness in microblogging search, *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM-2012), Maui, HI, USA*, pp. 2483–2486.

Amodeo, G., Blanco, R. and Brefeld, U. (2011). Hybrid models for future event prediction, *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM-2011), Glasgow, UK*, pp. 1981–1984.

Babcock, B., Babu, S., Datar, M., Motwani, R. and Widom, J. (2002). Models and issues in data stream systems, *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Madison, WI, USA*, pp. 1–16.

Beringer, J. and Hüllermeier, E. (2006). Online clustering of parallel data streams, *Data and Knowledge Engineering* **58**(2): 180–204.

Bottou, L. (1998). Online algorithms and stochastic approximations, *in* D. Saad (Ed.), *Online Learning and Neural Networks*, Cambridge University Press, Cambridge.

Chen, G., Wu, X. and Zhu, X. (2005). Sequential pattern mining in multiple streams, *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM), Houston, TX, USA*, pp. 585–588.

Cheon, S.-P., Kim, S., Lee, S.-Y. and Lee, C.-B. (2009). Bayesian networks based rare event prediction with sensor data, *Knowledge-Based Systems* **22**(5): 336–343.

Cherniack, M., Balakrishnan, H., Balazinska, M., Carney, D., Cetintemel, U., Xing, Y. and Zdonik, S. (2003). Scalable distributed stream processing, *Proceedings of CIDR-03: 1st Biennial Conference on Innovative Database Systems, Asilomar, CA, USA*.

Considine, J., Li, F., Kollios, G. and Byers, J. (2004). Approximate aggregation techniques for sensor databases, *ICDE-04: 20th IEEE International Conference on Data Engineering, Boston, MA, USA*, pp. 449–460.

Cormode, G. and Muthukrishnan, S. (2005). What's hot and what's not: Tracking most frequent items dynamically, *ACM Transactions on Database Systems* **30**(1): 249–278.

Cox, D. (1972). Regression models and life tables, *Journal of the Royal Statistical Society B* **34**(2): 187–220.

Cox, D. R. and Oakes, D. (1984). *Analysis of Survival Data*, Chapman & Hall, London.

Das, A., Gehrke, J. and Riedewald, M. (2003). Approximate join processing over data streams, *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, CA, USA*, pp. 40–51.

Domingos, P. and Hulten, G. (2003). A general framework for mining massive data streams, *Journal of Computational and Graphical Statistics* **12**(4): 945–949.

Gaber, M.M., Zaslavsky, A. and Krishnaswamy, S. (2005). Mining data streams: A review, *ACM SIGMOD Record* **34**(1): 18–26.

Gama, J. (2012). A survey on learning from data streams: Current and future trends, *Progress in Artificial Intelligence* **1**(1): 45–55.

Gama, J. and Gaber, M.M. (2007). *Learning from Data Streams*, Springer-Verlag, Berlin/New York, NY.

Garofalakis, M., Gehrke, J. and Rastogi, R. (2002). Querying and mining data streams: You only get one look, *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, Madison, WI, USA*, pp. 635–635.

Golab, L. and Tamer, M. (2003). Issues in data stream management, *ACM SIGMOD Record* **32**(2): 5–14.

Hulten, G., Spencer, L. and Domingos, P. (2001). Mining time-changing data streams, *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA*, pp. 97–106.

Ikonomovska, E., Gama, J. and Dzeroski, S. (2011). Learning model trees from evolving data streams, *Data Mining and Knowledge Discovery* **23**(1): 128–168.

Krizanovic, K., Galic, Z. and Baranovic, M. (2011). Data types and operations for spatio-temporal data streams, *IEEE International Conference on Mobile Data Management (MDM), Luleå, Sweden*, pp. 11–14.

Li, R., Lei, K.H., Khadiwala, R. and Chang, K.C.-C. (2012). Tedas: A twitter-based event detection and analysis system, *Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA,* pp. 1273–1276.

Oliveira, M. and Gama, J. (2012). A framework to monitor clusters evolution applied to economy and finance problems, *Intelligent Data Analysis* **16**(1): 93–111.

Radinsky, K. and Horvitz, E. (2013). Mining the web to predict future events, *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM 2013), Rome, Italy,* pp. 255–264.

Sakaki, T., Okazaki, M. and Matsuo, Y. (2013). Tweet analysis for real-time event detection and earthquake reporting system development, *IEEE Transactions on Knowledge and Data Engineering* **25**(4): 919–931.

Weng, J. and Lee, B.-S. (2011). Event detection in twitter, *Proceedings of the 5th International Conference on Weblogs and Social Media (ICWSM 2011), Barcelona, Spain*.

Yang, Y., Pierce, T. and Carbonell, J.G. (1998). A study of retrospective and on-line event detection, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998), Melbourne, Australia*, pp. 28–36.

Zadeh, L. (1965). Fuzzy sets, *Information and Control* **8**(3): 338–353.

Zupan, B., Demšar, J., Kattan, M.W., Beck, J.R. and Bratko, I. (2000). Machine learning for survival analysis: A case study on recurrence of prostate cancer, *Artificial Intelligence in Medicine* **20**(1): 59–75.

Ammar Shaker received his Bachelor's degree in informatics engineering from the University of Damascus (Syria) in 2005 and his Master's degree in data and knowledge engineering from Otto-von-Guericke University Magdeburg (Germany) in 2010. Since June 2010, he has been a research assistant in the Computational Intelligence Lab at Philipps-University Marburg, where he is working on his Ph.D. under the supervision of Prof. Eyke Hüllermeier. His research interests include machine learning and data mining, in particular evolving fuzzy systems and learning from data streams.

Eyke Hüllermeier is with the Department of Mathematics and Computer Science at Marburg University (Germany), where he holds an appointment as a full professor and heads the Computational Intelligence Group. He holds M.Sc. degrees in mathematics and business computing, a Ph.D. in computer science, and a habilitation degree, all from the University of Paderborn (Germany). His research interests are centered around methods and theoretical foundations of intelligent systems, with a specific focus on machine learning and reasoning under uncertainty. He has published more than 200 articles on these topics in top-tier journals and major international conferences, and several of his contributions have been recognized with scientific awards. Professor Hüllermeier is a co-editor-in-chief of *Fuzzy Sets and Systems*, one of the leading journals in the field of computational intelligence, and serves on editorial boards of several other journals, including *Machine Learning* and *IEEE Transactions on Fuzzy Systems*.

Received: 30 January 2013
Revised: 30 August 2013