Automated Detection of Epidemics from the Usage Logs of a Physicians' Reference Database

Jaana Heino[1] and Hannu Toivonen[2]

[1] National Public Health Institute, Helsinki, Finland
Department of Computer Science, University of Helsinki
jaana.heino@cs.helsinki.fi
[2] Department of Computer Science, University of Helsinki
hannu.toivonen@cs.helsinki.fi

Abstract. Epidemics of infectious diseases are usually recognized by an observation of an abnormal cluster of cases. Usually, the recognition is not automated, and relies on the alertness of human health care workers. This can lead to significant delays in detection. Since real-time data from the physicians' offices is not available. However, in Finland a Web-based collection of guidelines for primary care exists, and increases in queries concerning certain disease have been shown to correlate to epidemics. We introduce a simple method for automated online mining of probable epidemics from the log of this database. The method is based on deriving a smoothed time series from the data, on using a flexible selection of data for comparison, and on applying randomization statistics to estimate the significance of findings. Experimental results on simulated and real data show that the method can provide accurate and early detection of epidemics.

1 Introduction

The usual way of recognizing an infectious disease epidemic is through an observation of clustering (temporal, geographical, or both) of new cases by someone involved either with the diagnosis of patients or a disease registry. When the epidemic is widely spread – either spatially, temporally, or both – it might be very difficult or outright impossible for a single individual on the field to notice the change, and registries only receive notification after the patient has met a physician, laboratory samples have been analyzed, and reports filed, which causes delay.

In certain diseases, however, early detection would be very desirable. This is the case in, for instance, rare diseases whose etiology is not clear, in order to begin epidemiological studies as early as possible. Another example of the benefits of early detection comes from the detection of food-borne epidemics, where control measures are often much easier to conduct in the early phases of the epidemic.

If the registries could utilize diagnostic hypotheses made by physicians as soon as the patient has met the doctor, the delay would be eliminated. Unfortunately, such data is not available. However, in Finland a database exists that might be a way around this problem. The Physician's Reference Database [1] is a collection of medical guidelines for the primary care, from which physicians often seek information about infectious diseases [2]. According to a preliminary study, an increase in the rate of database searches about a certain disease sometimes correlates to the onset of an epidemic [3].

In this work we develop a simple method for automatic detection and evaluation of such increases.

What Is an Epidemic? *Incidence*, in epidemiology, is defined as the number of new cases per unit of time. *Prevalence* refers to the number of the people with the certain characteristic (e.g. a certain disease) in the population at any given moment; disease prevalence is thus dependent on the incidence of the disease and the duration of the disease.

Incidence and prevalence are well-defined concepts. An *epidemic*, on the other hand, is more difficult to define precisely. In mathematical modeling of infectious diseases (e.g. [4]) an epidemic is said to occur if the introduction of an infectious agent to a population causes people (other than the first introducer) to get sick through people transmitting the disease to each other.

This definition is, however, useful only in theoretical modeling, or when the actual incidence in the absence of an epidemic is zero. Many diseases have a certain *baseline incidence*: when a steady incidence and prevalence are considered the normal situation, even if the transfer happens person-to-person. Often, we also call a cluster of cases an "epidemic" even though no actual person-to-person transmission happens. What "an epidemic" is thus depends on the observer's subjective goals and estimates of local conditions.

Many infectious diseases have a seasonal cycle cycle: the incidence of the disease increases and decreases with a certain, steady interval. Some diseases, like influenza, have yearly peaking epidemics. Some have simply a slightly higher incidence during a certain time of year: for instance, food- and water-borne diseases are more common in warm weather. Some diseases have a longer cycle: for example, a major Pogosta disease epidemic occurs about every seven years in Finland.

A generic goal of automated detection of epidemics is to discover the beginning of any epidemic, whether cyclic or not. For cyclic diseases, an alternative goal is to consider the cyclic variation normal and to detect incidences that are exceptionally high given the normal cycle.

Requirements for Online Surveillance of Epidemics. It is of course important that a detection system achieves high sensitivity (proportion of epidemics detected), so that epidemics are not overlooked. It is also important that epidemics are detected as soon as possible after their onset. However, false alarms severly undermine the credibility of the warning system, which might result in the users no longer taking the warnings seriously. Thus, high specificity (proportion of non-epidemic periods classified as non-epidemic), leading to a higher positive prediction value (probability of epidemic given that the system outputs an alarm), should be a priority even at the cost of some reduction in the detection speed.

With a good system, the user can specify the period to which the present moment is compared: for instance, it should be possible to ask "is this week different from the previous n months", "is this month different from the same month during previous years", and several other questions like that, with the same method.

The method we introduce allows such flexibility and different treatments of cyclic diseases. We will analyze the sensitivity, specificity and positive prediction value of the method on both synthetic and real data.

2 Physician's Reference Database and Data Preprocessing

The Physician's Reference Database is a web-based collection of medical guidelines used by physicians. The database consists of thousands of articles, each described by a number of keywords (typically names of diseases, symptoms and findings). Reading events are recorded in a usage log, allowing one to mine for physicians' active interests in different diseases.

Let A be the set of all articles in the database, and $i = 1, 2, ..., n$ the sequence of days under surveillance. The raw data consists of the number of reading events in a day, $D(a)_i$, for each article $a \in A$. Each of the articles has associated keywords. Let $A(k)$ be the set of articles that contain disease k among its keywords. For each day i and each keyword k, the *daily total count of events of all relevant articles* is

$$D(k)_i = \sum_{a \in A(k)} D(a)_i.$$

Usage data from the reference database is available from October 1st 2000 onward; in the analysis in this work we use data until September 30th 2002. On average, there were 1465 reading events (a user viewing an article) per day, by all users total. The event counts show a notable upward trend: during the first 100 days the average was 633.9 events and during the 100 last days 2696.1.

As trends related to the changing usage of the database are not relevant, it is necessary is to "normalize" daily counts in relation to the overall database usage. We divide the daily event count per keyword by the total event count, giving the basic unit of our data, the *proportional daily event count*:

$$d(k)_i = \frac{D(k)_i}{\sum_{a \in A} D(a)_i}.$$

There are some potential problems with this normalization approach: something that is likely to affect the keyword-specific event counts is also likely to affect the total count. This might cause artefacts, ie. trends or peaks that are not present in the original data, or "dilution" of a smaller epidemic by a bigger one. At the moment we do not try to counteract such possible side effects.

3 The Method

The goal is to detect possible epidemics, observable as exceptionally high proportional daily event counts, and to output an alarm as soon as possible after the beginning of an epidemic. We develop a simple randomization-based framework to recognize significant increases in event counts.

The series is first *smoothed* to remove some of the daily variations while retaining most of the trends. Two smoothing methods, namely sliding average and sliding linear regression, are used (see below).

A *null hypothesis period*, a sequence of days from the past to which the present moment is compared, is chosen. The way the null hypothesis period is designated determines the exact question we are trying to answer. When the present moment is compared to all past non-epidemic times, the question is "is there an epidemic now". Other examples include the last n days ("has the situation changed for the worse recently"), the same months during previous years ("is this June different from what is typical for previous Junes"), and all previous epidemic times ("is there an epidemic now that is worse than typical epidemics of this disease").

We assume that in the absence of an epidemic the proportional daily event counts for a given disease are independent and identically distributed, but note that the independence assumption is most likely not completely true. A person that has read a lot about some disease recently is less likely to review that information than he would be if he knew nothing about the subject. Modeling such dependencies would be tedious, though, and we hope that this dependency averages out among all users.

As we do not know the true distribution behind the data, we cannot obtain a p-value or other comparison based on that. Instead, we use *randomization statistics*. The null hypothesis period is sampled with replacement for samples the size of the smoothing function's window[3], and the smoothed value is calculated for each of these samples. The resulting empirical distribution is used as the distribution of the smoothed values under the null hypothesis of no epidemic, and the p-value of the day in question is taken to be the proportion of the sampled sequences having the same or a higher smoothed value. (For more information on randomization statistics, see for instance [6].)

Testing every day like this could cause a bad positive prediction value for two reasons. First, if the proportion of negatives to positives in the set of objects tested rises, even a good specificity causes bad prediction values eventually. Second, the detection method itself includes randomization, and thus will eventually err if run repeatedly.

To avoid this problem, we require that the smoothed value of a day is both high (when compared with other observed values) and statistically significant (tested with randomization). The first requirement is fulfilled by checking if the smoothed value of that day is higher than a certain percentile of smoothed

[3] Note that here we sample new sequences, that is, individual points until we have w points, not windows of size w from the original series.

values of the original series in the null hypothesis period. Thus, the tested set of days is limited to a subset of all days having a high value of the statistic under surveillance. The use of a cut-off value can also be seen as a crude and quick estimate of our statistical test.

To put this together, given a time series, a window length $w > 1$, a cutoff value $c \in [50, 100[$, a smoothing function f from a window in the series (a run of consecutive points) to a real number, a null hypothesis period, and a p-value, the method works as follows:

(1) for the null hypothesis period, calculate the w-day smoothed values according to f, store these in S_1
(2) check if today's smoothed value exceeds the c'th percentile of all smoothed values in S_1, and if it does:
    (3) resample samples of size w from the null hypothesis period of the original non-smoothed series, and calculate the smoothed value for each of these windows, store these in S_2
    (4) determine the proportion of values in S_2 that are the same or higher than the value for today, and if that proportion is lower than p:
        (5) output an alarm, together with the proportion.

The (non-weighted) w-day *sliding average* is calculated simply by replacing each data point with an average of w days. This can be done either using time points on both sides of the day, or using the previous $w - 1$ days. As in this problem future data is not available, we use the latter method:

$$SA_i = \sum_{j=1}^{w} d_{i-j+1}$$

Sliding linear regression, on the other hand, works by fitting *a line* with the least squares over the w days $[i - w + 1, i]$, and then taking the smoothed value at i to be the value of this linear function at that point. The benefit of this smoothing method is that it reacts faster to abrupt changes in the series; in a way the sliding linear function exaggerates the linear tendencies in the series, while the sliding average tries to smooth them out.

Naturally, the longer the window, the less short-term changes affect the smoothed series. However, as the window stretches only backward in time, as opposed to backward and forward, this causes a lag in the smoothed curves' reaction to changes. See Figure 1 for an example series and smoothings on it.

4 Test Results

4.1 Results on Artificial Test Data

Forty artificial test series were constructed to test the performance of the algorithm with different parameters. Each series is 700 time points long. 20 are constructed from an exponential distribution, another 20 from a normal distribution with values below zero replaced by zero. 10 of each type of series had no epidemics; in the other 20 datasets timepoints $[351, 450]$ were replaced by samples drawn from a similar distribution as the main series, but with a higher mean. Parameters for the distributions were chosen based on the means and variations in the real life test data. See Figure 1 for an example series.

In all the tests on artificial data, the null hypothesis period for timepoint i is $[1, i-w]$ for the non-epidemic series and $[1, i-w] \setminus [351, 450]$ for the epidemic ones. As we know for certain which days are epidemic and which are not, we can calculate sensitivity, specificity and delays exactly. No epidemic was completely missed, giving an *epidemics-wise* sensitivity of exactly 1 for all settings. Below, we have explored the sensitivity and specificity *day-wise*, that is, the algorithm is expected to mark each day either belonging to an epidemic or not. We also examine the detection delay, defined as the number of false negatives from the first day of an epidemic (time point 351) until the first true positive during the epidemic.

In practice, 20,000 samples were enough to produce steady results on the randomization tests; 80,000 were used for the sliding average and 20,000 for the sliding linear, due to the first one's Matlab implementation being so much faster that the extra certainty was worthwhile. Unless mentioned otherwise, the p-value is 0.01. When calculating the performance statistics, the first one hundred days are ignored.

Figure 2.a shows the day-wise specificity of the algorithm for all the test series with different parameter values. Note how specificity drops in the sliding average tests. This is mostly due to the fact that the longer the window, the longer it takes after the epidemic period before the smoothed values return to the baseline. As a faster-reacting function, sliding linear regression does not suffer from similar problems, but the distribution of the specificities widens when the window grows: more series are detected with 100 % specificity, but some epidemic periods are also less well detected than they would have a shorter window.

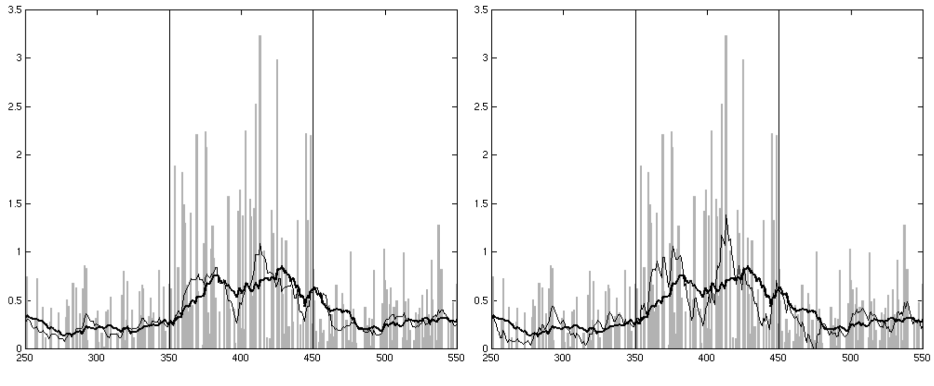
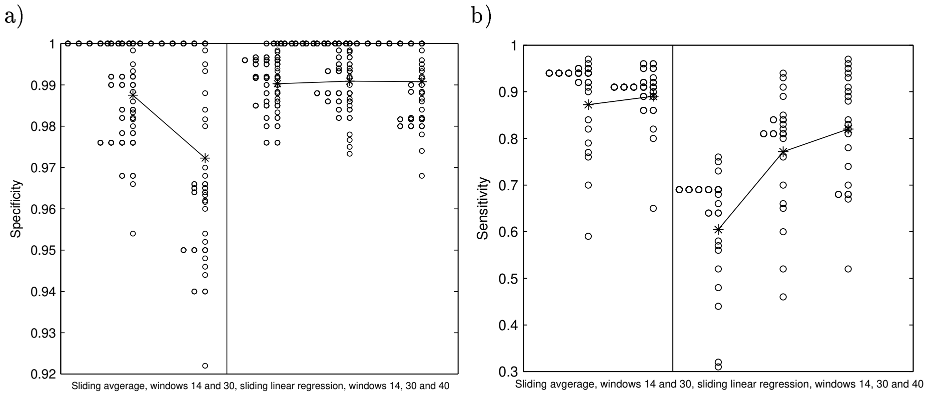


Fig. 1. A portion of an artificial time series (gray bars) with an "epidemic" in the middle of it (bordered by the vertical black lines). On the left; a 14-day sliding average (thin line) and a 30-day sliding average (thick line). On the right, a 30-day sliding average (again the thick line) and a 30-day sliding linear regression (thin line).



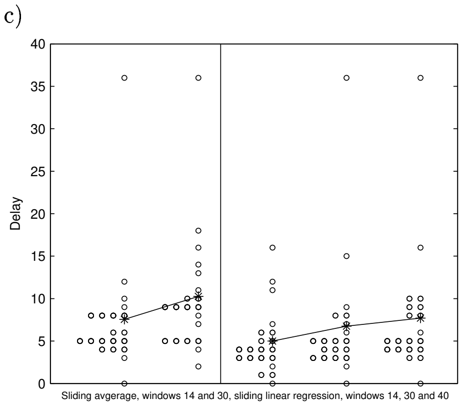


Fig. 2. Performance of the algorithm for cutoff = 99, p = 0.01. In each subfigure, from the left, sliding average with windows 14 and 30, and sliding linear regression with 14, 30 and 40. Circles and squares represent individual test series. Stars connected by a line show the average performance figure. Shown are a) specificity, b) sensitivity, c) delay.

As one would expect, specificity is dependent on the parameters cutoff and p. If window length is kept constant and cutoff and p varied, keeping cutoff $= 100(1-p)$, the specificity is linear on this double parameter (not shown in the figure).

Figure 2.b shows the respective day-wise sensitivity for all the series. Here we see the phenomenon that lowering the window length below a certain limit causes a notable decline in sensitivity. This happens because when the window grows, the distributions of the smoothed values during epidemic and non-epidemic become narrower and overlap less, making them easier to distinguish. Figure 2.c gives the delay in detecting the epidemic, again for the same parameters. As expected, the average delay rises with the window length. Apart from the one outlier, all delays are below 20 days and the vast majority of them below 10 days.

One could argue that shortening the window length to shorten the delay is worthwhile, since as the epidemic still is detected the lower sensitivity does not matter. To some extent this is true. However, one important way to tell a real epidemic from a false positive is whether the situation *persists*. False positives appear singly or in short clusters, real epidemics last for several days or weeks.

While we can afford to lose some sensitivity in order to gain a shorter delay, we cannot let it go altogether.

Experimenting on the effect of window length, keeping other parameters constant, revealed that when the window length shortens, sensitivity stays reasonably good (that is, around 80-90 %) up to a point, and then drops steeply. This drop happens around window length 15 for the sliding average and around window length 40 for the sliding linear regression (at least on the data used in this work). Comparing the specificity and sensitivity of the smoothing methods on these lengths, it can be seen see that while there are some differences on certain series between the methods, their overall performance is close to equal, and that there are no systematic differences depending on the type of the series. (Test results not shown.)

4.2 Results on Real Life Data

Ten diseases were selected as test targets, namely hepatitis A, influenza, diphtheria, legionellosis, Pogosta disease, polio, parotitis, tularemia, varicella and measles. The keyword for each is the (Finnish) name of the disease, and each test series is 729 days long. Parameters used are cutoff = 99,$p = 0.01$, and window lengths 14 (sliding average) and 40 (sliding linear regression).

Unlike with artificial data, we do not now have conclusive knowledge of which days are epidemic and which are not; the only definite example of an epidemic interesting from the public health point of view is the Pogosta epidemic of 2002, which began in August (Figure 3.a). So we count negatives after August 1st 2002 (day 668) in that series as false negatives, and exclude timepoints from day 680 onward from the null hypothesis period. In other series the null hypothesis period is $[1, i - w]$.

Delay of detection. The Pogosta epidemic was detected on day 665 by the sliding average, and on day 666 by the sliding linear regression method (see Figure 3.a). Compared to the shape of the actual epidemic (the thick line at the bottom), we can see that this is remarkably early. The epidemic curve is drawn based on the day the diagnostic blood sample was taken, which is likely to be the same day as the day after the patient's physician first suspected the disease. Even if we had data straight from the physician's office, we could not have seen the epidemic much earlier. (Data for the epidemic curve is based on cases later notified to the Infectious Diseases Registry of the National Public Health Institute.)

On the sliding average there was a short alarm peak also at time-points 636-639. It is unclear whether this is a true first alarm of the epidemic, or a sequence of false positives; in the following it is counted as the latter. No days were falsely classified as negative during the epidemic.

Sensitivity, specificity and positive prediction value. Over all the series, in the sliding average, if we consider only the Pogosta epidemic of 2002 as true positives, specificity was 99.02 %, and the positive prediction value 51.2 %. The sliding linear regression had specificity 99.00 % and positive prediction value of 50.5 %.

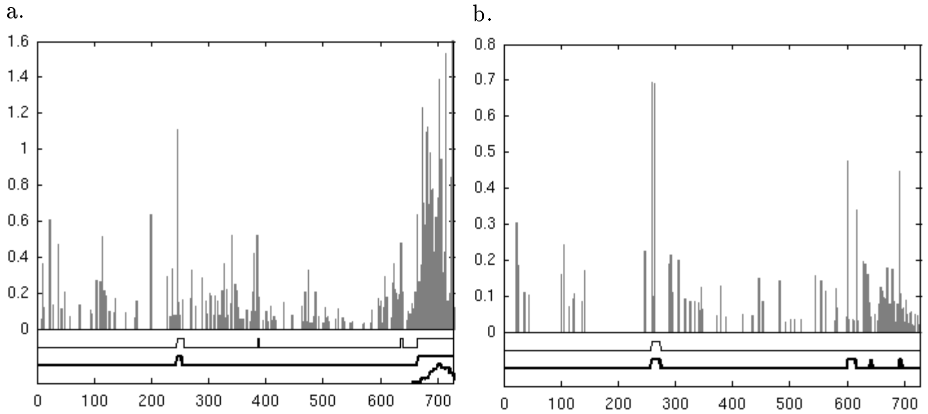


Fig. 3. The output of the algorithm in some situations. a. The Pogosta series with the output of both smoothing methods. Thin line, sliding average; thick line in the middle, sliding linear regression. The null hypothesis period is the whole preceding period minus timepoints 680 and onward. The bottom-most thick line shows the shape of the actual epidemic: weekly incidence according to the day the first diagnostic blood sample was taken, which probably is the same day or close to the day that the patient's physician first suspected the disease. b. The tularemia series with the output of the sliding average smoothing method. Thin line, the null hypothesis period is the whole preceding series; thick line, it is the preceding 180 days.

With the two methods, continuous runs of "false positives" happened in four series (legionella, mumps series, Pogosta, and tularemia) in about the same places. Looking at the series, and bearing in mind that the detection is based only on time previous to those time-points, these four periods of alarms seem reasonable and even desirable. See Figure 3 for two of the cases.

If we count the positives during these periods as true positives, we get a positive predictive value of 85.3 % for the sliding average and 90.1 % for the sliding linear regression. (Stating specificity would require arbitrarily determining which days, if any, around these positives are also positives.) In the real situation, where the series under surveillance and the null hypothesis periods are chosen by an epidemiologist, the positive prediction value (defined through the usefulness of the alarm) will probably be somewhere between these estimates of 50 and 90 %.

4.3 Changing the Null Hypothesis Period

Figure 3.b demonstrates the effect of the null hypothesis period. The thin line shows the output when the null hypothesis period is the whole preceding series; the thick line when it is last 180 days. The yearly tularemia epidemic during the second year does not cause an alarm when the epidemic during the previous year is included in the null hypothesis, and does cause an alarm when it is not.

Another interesting feature is also visible in the lower output. Looking at the second predicted epidemic (from timepoint 600 onward), we can see that there first is an alarm period, but when the alarm has been on for some time, it ceases. When the epidemic rises again, new alarms are put out. This happens because the epidemic time is now not excluded from the null hypothesis. Thus "epidemics inside epidemics" can be detected.

5 Technical Comments

The time requirement of the method for the check on one day is linear to the number of iterations i performed in the randomization and the window length w, as taking a random sample and calculating a mean or fitting a line are all linear. Thus, running the check for m diseases (which is what would normally be done daily) with i iterations is $O(wmi)$, and calculating the results for n days is $O(nwmi)$. Since typically $w << i$, these are close to $O(mi)$ and $O(nmi)$.

Currently, data arrives from the system administrator of the actual database daily, as a text file. The data is read into an (Oracle RDB) database, and the reading events per keyword are calculated and stored. The raw article counts are also stored, to make it simpler to add a new disease keyword to the base (without the need to reread the data files). When the amount of data grows too large, article counts might be preserved for perhaps two years, and only keyword-specific values stored for a longer period, enabling the beginning of a new surveillance with some data, but not requiring too much space.

The test versions of the algorithms as explained in this work were implemented on Matlab. Conversion to a Java program for end users is planned.

6 Related Work

Basically, most methods of online detection of changes in a time series fall into two categories. In the first, we compare each value, at the time of its arrivel, to some baseline value calculated from previous data, and decide if the new value is "different enough" to be considered abnormal [7, 8]. In the second, we fit some model, often a curve – constant [9], linear [10, 11], or more complex [12] – piecewise to the data, searching for the change-points.

Other than these two approaches, data mining of time series data has been studied mainly from the point of view of mining inter-series relations in either patterns (slopes, peaks) or in concurrently occurring different events (see, for example, [13, 14]), which approaches are not directly related to the problem of this study.

Piatetsky-Shapiro and Matheus were among the first data miners to investigate deviations in time series data [7]. Their basic concept is a deviation, defined as a difference between an observed value and a normative value. In addition, they categorize the deviations based on their interestingness, defined as the utility of the finding to a user.

Stern and Lightfoot describe a system for detecting clusters of human infection with enteric pathogens [8]. In it, a smoothing technique is used to determine normal baseline incidences for each pathogen, area and time-of-year, based on data from several previous years. Then weekly counts of cases are compared to a threshold calculated from the base incidence. The system achieves great sensitivity, over 90 %, but the positive predictive value remains at about 60 % or lower - meaning that almost every other alarm is false.

The problem with these approaches is that calculating the normative values requires data from several years, and that they give accurate results only if the difference between normal and abnormal values is rather sharp. Another problem is that the calculation of normative values requires some knowledge or assumption about the shape of the distribution of the values; in our case such knowledge is not readily available.

Ogden and Sugiura [10] describe test statistics for determining whether a time series has undergone a change. The change is defined as a linear change in the parameters of the underlying distribution: the parameter vector is θ from the beginning of the series to some timepoint t_i, then changes in a linear way until it reaches $\theta + \delta$ at $t_j, j > i$. The null hypothesis tested is $\delta = 0$. However, the tests cannot be applied online to decide if there has been a change recently, and require information on the distribution of the data.

Keogh *et al.* [11] explore segmenting of a time series into piecewise linear representation, relative to an error criterion stating whether a line fit is "good enough" (for example, the total error must not exceed a certain value). They describe three basic greedy approaches, only one of them online, and a combination online algorithm that performed well. Guralnik and Srivastava [12] suggest not restricting the function to be fitted to the segments to lines (for instance, one could allow the algorithm to choose the best fit of 0-3 degree polynomials, instead). In his thesis [9] Marko Salmenkivi introduced methods for intensity modeling; that is, assuming a sequence of independent events in time, finding a piece-wise constant function describing the intensity of the Poisson-process producing that sequence.

Most of the above methods is directly suitable for online detection of change points. We experimented with the online algorithm of Keogh *et al.*, trying to use them to detect epidemics. The idea was to segment the series, and then look at the slope of the last segments. Unfortunately, we were unable to calibrate an error criterion that would be both specific enough and produce a sort enough delay, and unable to adapt the method to answer several surveillance questions (for instance, comparing this month's situation to the same months two previous years proved impractical). Similar problems apply to the other change-point detection/segmentation approaches.

7 Conclusions

A method was developed to automatically detect epidemics from an online time-series. Despite its simplicity, the method works reliably. In all the test data, all epidemics were correctly detected. Even when calculated day-wise instead of epidemics-wise, we achieved specificity over 99 % and sensitivity over 80 %. Also the results on real-life data were very encouraging.

A nice feature of the method is the adaptability that is achieved by changing the null hypothesis period. The same method can readily answer several kinds of questions of interest such as detecting acute short-term changes and comparing epidemics.

However, caution must be used before widely applying this – or any – method of online surveillance. It must be kept in mind that all surveillance requires the capacity to deal with both true and false alarms; surveillance is useless unless personnel exist to work on the alarms. A separate prospective study will be necessary to establish the actual benefits of surveillance.

References

1. Physician's Desk Reference and Database (in Finnish). Kustannus Oy Duodecim. Yearly revised edition. URL:http://www.duodecim.fi/kustannus/cdrom/index.html (14.3.2002).
2. Jousimaa. J., Kunnamo. I., Physicians' patterns of using a computerized collection of guidelines of primary health care. *Int J Technol Assess Health Care.* 14:484-93. 1998.
3. Jormanainen. V., Jousimaa. J., Kunnamo. I., Ruutu. P., Physicians' Database Searches as a Tool for Early Detection of Epidemics. *Emerging Infectious Diseases.* 7(3). 2001.
4. Diekman, O., Heesterbek, J. A. P, *Mathematical Epidemiology of Infectious Diseases.* Wiley Series in Mathematical and Computational Biology, John Wiley & Sons Ltd, 2000.
5. Farmer, R. D. T., Miller D. L., Lawrenson R., *Lecture Notes on Epidemiology and Public Health Medicine.* Blackwell Scientific, Oxford, 1996.
6. Manly, B. F. J. *Randomization, Bootstrap and Monte Carlo Methods in Biology*, Second Edition. Volume 41 in the series "Texts in Statistical Science Series". CRC Press, 1997.
7. Piatetsky-Shapiro G., Matheus, C. J., The Interestingness of Deviations. *AAAI-94 Knowledge Discovery in Databases Workshop*, 1994.
8. Stern, L., Lightfoot, D., Automated Outbreak Detection: a quantitative retrospective analysis. *Epidemiol. Infect.*, 122, 103-110, 1999.
9. Salmenkivi. S., Computational Methods for Intensity Models. Department of Computer Science. University of Helsinki. Series of Publications A-2001-2. 2001.
10. Ogden. S., Sugiura. N., Testing Change-points with Linear Trend. *Communications in Statistics B: Simulation and Computation.* 23:287-322. 1994.
11. Keogh, E., Chu, S., Hart, D., Pazzani. M., An Online Algorithm for Segmenting Time Series. *IEEE International Conference on Data Mining. 2001.*
12. Guralnik. V., Srivastava. J., Event Detection from Time Series Data. *Proc. Fifth ACM SIGKDD*, 1999.
13. Last, M., Klein, Y., Kandel, A., Knowledge discovery in Time Series Databases. Correspondence in *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 2001.
14. Das. G., Lin. K., Mannila. H., Renganathan. G., Smyth. P., Rule discovery from time series. *Proc. KDD'98*, 16-22, 1998.