An R-package for the surveillance of infectious diseases

Michael Höhle[1]

Department of Statistics, University of Munich, Germany
hoehle@stat.uni-muenchen.de.

Summary. The R-Package 'surveillance' implements algorithms for the detection of aberrations in routinely collected surveillance data. It contains the procedures described by Stroup et al. [10], Farrington et al. [4] and the system used at the Robert Koch Institute, Germany. For evaluation purposes, the package includes example data sets and functionality to generate surveillance data by simulation. To compare the algorithms benchmark numbers like sensitivity, specificity, and detection delay can be computed for a set of time series. Being an open-source package it should be easy to integrate new algorithms. As an example of this process, a simple Bayesian surveillance algorithm is described, implemented and tested.

Key words: epidemiology, monitoring, software, time series of counts

1 Introduction

Public health authorities have, in an attempt to meet the threats of infectious diseases, created comprehensive mechanisms for the collection of disease data. The vast amounts of data resulting from this acquisition demands the development of automated algorithms for the detection of abnormalities. Typically, such an algorithm monitors a univariate time series of counts by a combination of heuristic methods and statistical modelling. Prominent examples of surveillance algorithms are the work by Stroup et al. [10] and Farrington et al. [4]. A comprehensive survey of outbreak detection methods can be found in [3].

The R-package `surveillance` available from CRAN[1] was written with the aim of providing a test-bench for surveillance algorithms. It allows users to test new algorithms and compare their results with those of standard surveillance methods. Real world outbreak datasets are included together with mechanisms for simulating surveillance data. With the package at hand, comparisons like the one described by Hutwagner et al. [6] should be easy to conduct.

[1] http://cran.r-project.org

This paper is organized as follows. Section 2 gives a brief introduction to surveillance data and illustrates how to create new datasets by simulation. Section 3 exemplifies the use of surveillance algorithms by analysing German outbreak data. Finally, Section 4 provides a discussion and indicates directions of future work.

2 Surveillance Data

Denote by $\{y_t\,; t = 1, \ldots, n\}$ the time series of counts. Because such data typically are collected on a weekly basis, the alternative notation $\{y_{i:j}\}$ shall also be used, with $j = \{1, \ldots, 52\}$ being the week number in year $i = \{-b, \ldots, -1, 0\}$. That way the years are indexed such that the most current year has index zero. For evaluation of the outbreak detection algorithms it is also possible for each week to store – if known – whether there was an outbreak that week. The resulting multivariate series $\{(y_t, x_t)\,; t = 1, \ldots, n\}$ is in `surveillance` given by an object of class `disProg` (disease progress), which is basically a `list` containing two vectors: the observed number of counts and a boolean vector `state` indicating whether there was an outbreak that week. A number of time series are contained in the `data` directory, mainly originating from the SurvStat@RKI database at `http://www3.rki.de/SurvStat/` maintained by the Robert Koch Institute, Germany [9]. For example the object `k1` describes cryptosporidiosis surveillance data for the German federal state Baden-Württemberg 2001-2005. The peak in 2001 is due to an outbreak of cryptosporidiosis among a group of army-soldiers in boot-camp. In `surveillance` the `readData` function brings the time series on `disProg` form.

```
> k1 <- readData("k1", week53to52 = TRUE)
> plot(k1, main = "Cryptosporidiosis in BW 2001-2005")
```

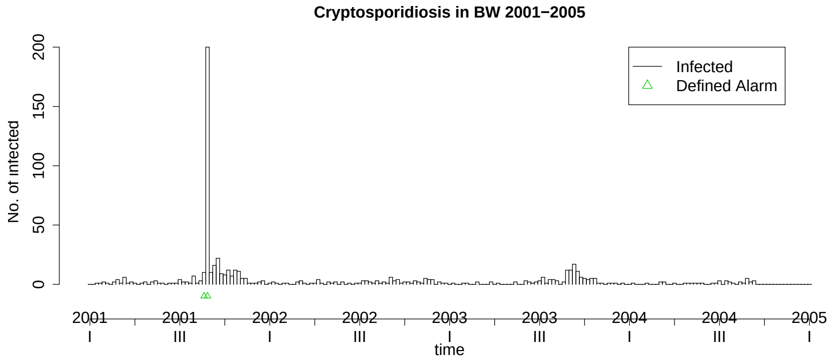

Fig. 1. Weekly cryptosporidiosis counts in Baden-Württemberg 2001-2005.

For test purposes it is also often of interest to generate surveillance data by simulation. A Hidden Markov Model (HMM) is introduced, where a binary state $X_t, t = 1, \ldots, n$, denotes whether there was an outbreak and Y_t is the number of observed counts. The state X_t is given by a homogeneous Markov chain with a 2×2 transition matrix specified by two parameters p and r: $P(X_{t+1} = 0 | X_t = 0) = p$ and $P(X_{t+1} = 1 | X_t = 1) = r$. In addition, the observed Y_t is Poisson-distributed with log-link mean depending on a seasonal effect and time trend, i.e.

$$\log \mu_t = A \cdot \sin\left(\omega \cdot (t + \varphi)\right) + \alpha + \beta t.$$

In case of an outbreak ($X_t = 1$) the mean increases with a value of K, altogether

$$Y_t \sim \text{Po}(\mu_t + K \cdot X_t). \tag{1}$$

The model in (1) corresponds to a single-source, common-vehicle outbreak, where the length of an outbreak is controlled by the transition probability r and the frequencies of outbreaks by p. The advantage of (1) is that it allows for an easy definition of a correctly identified outbreak: each $X_t = 1$ has to be identified. More advanced setups would require different definitions of an outbreak, e.g. as a connected series of time instances, where the number of outbreak cases is greater than zero. Care is then required in defining what a correctly identified outbreak for time-wise overlapping outbreaks means.

In `surveillance` the function `sim.pointSource` is used to simulate such a point-source epidemic; the result is an object of class `disProg`.

```
> sts <- sim.pointSource(p = 0.99, r = 0.5, length = 400,
+     A = 1, alpha = 1, beta = 0, phi = 0, frequency = 1,
+     state = NULL, K = 1.7)
> plot(sts)
```

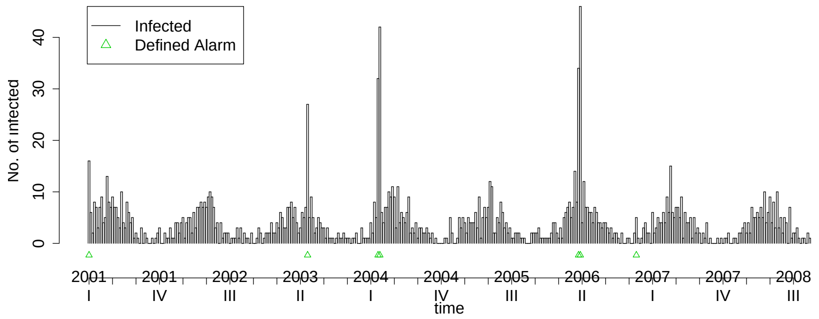

Fig. 2. A simulated time series. The triangles indicate time points, where $X_t = 1$.

3 Surveillance Algorithms

Surveillance data often exhibit strong seasonality, therefore most surveillance algorithms only use a set of so called *reference values*: Let $y_{0:t}$ be the number of cases of the current week (denoted week t in year 0), b the number of years to go back in time and w the number of weeks around t to include from these previous years. For the year zero we use w_0 as the number of previous weeks to include – typically $w_0 = w$. Altogether the set of reference values is:

$$R(w, w_0, b) = \left(\bigcup_{i=1}^{b} \bigcup_{j=-w}^{w} y_{-i:t+j} \right) \cup \left(\bigcup_{k=-w_0}^{-1} y_{0:t+k} \right).$$

This gives the number of cases at time points with similar conditions as at $y_{0:t}$. Note that the number of cases of the current week is not part of $R(w, w_0, b)$.

A surveillance algorithm is a procedure using the reference values to create a prediction $\hat{y}_{0:t}$ for the current week. This prediction is then compared with the observed $y_{0:t}$: if the observed number of cases is much higher than the predicted number, the current week is flagged for further investigations. In order to do surveillance for time $0 : t$ an important concern is the choice of b and w. Values as far back as time $-b : t - w$ contribute to $R(w, w_0, b)$ and thus have to exist in the observed time series.

Four different types of algorithms are implemented in `surveillance`. The Centers for Disease Control and Prevention (CDC) method [10], the Communicable Disease Surveillance Centre (CDSC) method [4], the method used at the Robert Koch Institute (RKI), Germany [1], and a Bayesian approach documented in Riebler [8]. To give an idea of the concepts the Bayesian approach developed in Riebler [8] is presented.

The model assumes that the reference values are identically and independently Poisson distributed with parameter λ and a gamma distribution is used as prior distribution for λ. The reference values are defined to be $R_{\text{Bayes}} = R(w, w_0, b) = \{y_1, \ldots, y_n\}$ and $y_{0:t}$ is the value to predict. Thus, $\lambda \sim \text{Ga}(\alpha, \beta)$ and $y_i | \lambda \sim \text{Po}(\lambda)$, $i = 1, \ldots, n$. Standard derivations show that the posterior distribution is

$$\lambda | y_1, \ldots, y_n \sim \text{Ga}(\alpha + \sum_{i=1}^{n} y_i, \beta + n).$$

Computing the predictive posterior distribution for the next observation

$$f(y_{n+1} | y_1, \ldots, y_n) = \int_0^\infty f(y_{n+1} | \lambda)\, f(\lambda | y_1, \ldots, y_n)\, d\lambda$$

one gets the Poisson-gamma distribution, which is a generalization of the negative binomial distribution. Altogether

$$y_{n+1}|y_1,\ldots,y_n \sim \text{NegBin}(\alpha+\sum_{i=1}^{n} y_i, \tfrac{\beta+n}{\beta+n+1}).$$

Using the Jeffrey's prior $\text{Ga}(\frac{1}{2},0)$ as non-informative prior distribution for λ the parameters of the negative binomial distribution are

$$\alpha+\sum_{i=1}^{n} y_i = \frac{1}{2} + \sum_{y_{i:j}\in R_{\text{Bayes}}} y_{i:j} \qquad \text{and} \qquad \frac{\beta+n}{\beta+n+1} = \frac{|R_{\text{Bayes}}|}{|R_{\text{Bayes}}|+1}.$$

Using a quantile-parameter α, the smallest value y_α is computed, so that $P(y_{n+1} \leq y_\alpha|y_1,\ldots,y_n) \geq 1-\alpha$. Now $A_{0:t} = I(y_{0:t} \geq y_\alpha)$, i.e. if the observed value $y_{0:t}$ is equal or greater than y_α then the current week is flagged as an alarm. For example, the `Bayes1` method uses the last six weeks as reference values, i.e. $R(w,w_0,b) = (6,6,0)$, and is applied to the `k1` dataset with $\alpha = 0.01$ as follows.

```
> k1.b660 <- algo.bayes(k1, control = list(range = 27:192,
+     b = 0, w = 6, alpha = 0.01))
> plot(k1.b660, disease = "k1", firstweek = 1, startyear = 2001)
```

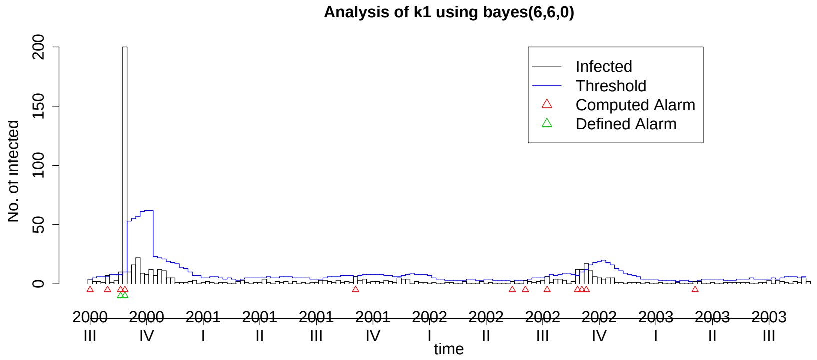

Fig. 3. The `Bayes1` algorithm with $R(w,w_0,b) = (6,6,0)$ applied to the `k1` dataset. Red triangles indicate alarms and should be compared to the known outbreaks (green triangles).

As an example of applying the more traditional algorithms the following call applies the CDC and Farrington procedure to the simulated time series `sts` from Fig. 2. Note that the CDC procedure operates with four-week aggregated data – to better compare the upper bound value, the aggregated number of counts for each week are thus shown as circles in the plot.

```
> par(mfcol = c(1, 2))
> cntrl <- list(range = 300:400, m = 1, w = 3, b = 5, alpha = 0.01)
> sts.cdc <- algo.cdc(sts, control = cntrl)
> sts.farrington <- algo.farrington(sts, control = cntrl)
> plot(sts.cdc, legend = F)
> plot(sts.farrington, legend = F)
```

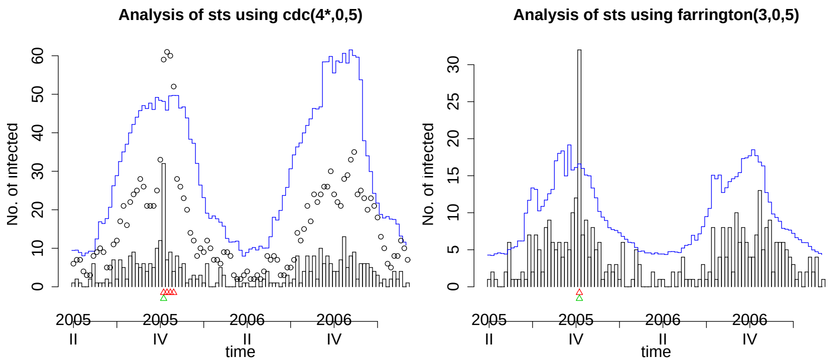

Fig. 4. The Farrington (left) and the CDC (right) algorithm applied to the simulated time series from Fig. 2.

Typically, one is interested in testing and comparing surveillance algorithms. An easy way is to look at the sensitivity and specificity of the procedure. A correct identification of an outbreak is defined as follows: If the algorithm raises an alarm for time t, i.e. $A_t = 1$ and $X_t = 1$, one has a correct classification. If $A_t = 1$ and $X_t = 0$, one has a false-positive. To compute various performance scores the function `algo.quality` can be used on a `SurvRes` object.

```
> print(algo.quality(k1.b660))
```

```
     TP FP TN  FN Sens Spec      dist       mlag
[1,] 2  10 154 0  1    0.9390244 0.06097561 0
```

This computes the number of false positives, true negatives, false negatives, the sensitivity and the specificity. Finally, `lag` is the average number of weeks between the first of a consecutive number of $X_t = 1$'s (i.e. an outbreak) and the first alarm raised by the algorithm.

To compare the results of several algorithms on a single time series, a list of control objects is declared – each containing the name and settings of the algorithm to be applied to the data. A test on a set of time series is then made as follows. Firstly, a list containing all time series is created. Secondly, all the algorithms specified in the afore mentioned control object are applied to each series. Consequently, all predefined algorithms are applied to the 14 surveillance time series from SurvStat@RKI (i.e. `outbrksNames`) as follows:

```
> outbrks <- lapply(outbrksNames, function(name) {
+     enlargeData(readData(name), range = 1:(4 * 52), times = 2)
+ })
> one.survstat.surv <- function(outbrk) {
+     algo.compare(algo.call(outbrk, control = control))
+ }
> algo.summary(lapply(outbrks, one.survstat.surv))
```

```
                    TP  FP   TN  FN sens spec dist mlag
rki(6,6,0)          38  62 2646 180 0.17 0.98 0.83 5.43
rki(6,6,1)          65  83 2625 153 0.30 0.97 0.70 5.57
rki(4,0,2)          80 106 2602 138 0.37 0.96 0.63 5.43
bayes(6,6,0)        61 206 2502 157 0.28 0.92 0.72 1.71
bayes(6,6,1)       123 968 1740  95 0.56 0.64 0.56 1.36
bayes(4,0,2)       162 920 1788  56 0.74 0.66 0.43 1.36
cdc(4*,0,5)         65  94 2614 153 0.30 0.97 0.70 7.14
farrington(3,0,5)  25  26 2682 193 0.11 0.99 0.89 8.21
```

The above results and previous simulation studies show that the Bayesian approach seems to do quite well. However, the extent of the above comparisons do not make allowance for any more supported statements. Consult the work of Riebler [8] for a more thorough comparison using simulation studies.

4 Discussion and Future work

The package provides a framework for the application of surveillance algorithms using the freely available environment for statistical computing "R". Combining the functionality of R with Sweave [7] and LaTeX allows for easy access to SQL databases and automatic generation of reports.

Casting surveillance algorithms into a Bayesian framework and thus interpreting alarm thresholds as quantiles of the posterior predictive distribution gives a new way to see outbreaks compared to the more traditional asymptotic normal based confidence intervals. However, an important issue remains multiple testing and the choice of the correct threshold. Several extensions of the described simple Bayesian approach are imaginable, e.g. the inane over-dispersion of surveillance data could be modeled by using a negative-binomial distribution and mechanisms to correct for past outbreaks could be added. However, in these situations methods like Markov Chain Monte Carlo or heuristic approximations have to be used in order to obtain the required alarm thresholds.

Currently, work is done to implement new algorithms into the package, e.g. the one described in [5]. An important aspect here is the visualisation and handling of multivariate surveillance time series. A further extension is to provide more complex mechanisms for the simulation of epidemics. In particular it would be interesting to include multi-day outbreaks originating from single-source exposure, but with delay due to varying incubation time [6] or SEIR-like epidemics [2].

5 Acknowledgements

My thanks to A. Riebler and C. Lang, University of Munich, for their work on the package implementation. I am grateful to K. Stark and D. Altmann, RKI, Germany, for discussions and information on the surveillance methods used by the RKI. The research was conducted with financial support from the Collaborative Research Centre SFB 386 funded by the German research foundation (DFG).

References

[1] D. Altmann. The Surveillance System of the Robert Koch Institute, Germany. Personal Communication, 2003.

[2] H. Andersson and T. Britton. *Stochastic Epidemic Models and their Statistical Analysis*, volume 151 of *Springer Lectures Notes in Statistics*. Springer-Verlag, 2000.

[3] C.P. Farrington and N. Andrews. Outbreak detection: Application to infectious disease surveillance. In R. Brookmeyer and D.F. Stroup, editors, *Monitoring the Health of Populations*, chapter 8, pages 203–231. Oxford University Press, 2003.

[4] C.P Farrington, N.J. Andrews, A.D. Beale, and M.A. Catchpole. A statistical algorithm for the early detection of outbreaks of infectious disease. *Journal of the Royal Statistical Society, Series A*, 159:547–563, 1996.

[5] L. Held, M. Höhle, and M. Hofmann. A statistical framework for the analysis of multivariate infectious disease surveillance data. *Statistical Modelling*, 5:187–199, 2005.

[6] L. Hutwagner, T. Browne, G.M Seeman, and A.T. Fleischhauer. Comparing abberation detection methods with simulated data. *Emerging Infectious Diseases*, 11:314–316, 2005.

[7] F. Leisch. Sweave, Part I: Mixing R and LaTeX. *R Newsletter*, 2(3): 28–31, 2003.

[8] A. Riebler. Empirischer Vergleich von statistischen Methoden zur Ausbruchserkennung bei Surveillance Daten. Bachelor's thesis, Department of Statistics, University of Munich, 2004.

[9] Robert Koch-Institut. SurvStat@RKI. http://www3.rki.de/SurvStat, 2004. Date of query: September 2004.

[10] D.F. Stroup, G.D. Williamson, J.L Herndon, and J.M Karon. Detection of aberrations in the occurence of notifiable diseases surveillance data. *Statistics in Medicine*, 8:323–329, 1989.