Published in final edited form as:

Pattern Recognit Lett. 2009 February 1; 30(3): 255–266. doi:10.1016/j.patrec.2008.09.008.

Mining aggregates of over-the-counter products for syndromic surveillance

Aurel Cami^{a,*}, Garrick L. Wallstrom^a, Ashley L. Fowlkes^b, Cathy A. Panozzo^b, and William R. Hogan^a

^a Department of Biomedical Informatics, University of Pittsburgh, Pittsburgh, PA 15260

^b Centers for Disease Control and Prevention (CDC), Atlanta, GA 30333

Abstract

There has recently been a surge of research efforts aimed at very early detection of disease outbreaks. An important strategy for improving the timeliness of outbreak detection is to identify signals that occur early in the epidemic process. We have developed a novel algorithm to identify aggregates of “similar” over-the-counter products that have strong association with a given disease. This paper discusses the proposed algorithm and reports the results of an evaluation experiment. The experimental results show that this algorithm holds promise for discovering product aggregates with outbreak detection performance that is superior to that of predefined categories. We also found that the products extracted by the proposed algorithm were more strongly correlated with the disease data than the standard predefined product categories, while also being more strongly correlated with each other than the products in any predefined category.

Keywords

Biosurveillance; Outbreak Detection; Linear Regression; Time Series Aggregation

1. Introduction

In recent years there has been a sharp increase in the research efforts on early detection of disease outbreaks. Disease outbreaks can sicken or kill people very quickly and spread rapidly. The window of opportunity to limit the damage of an outbreak can be as brief as a few days. Hence, the problem of improving the timeliness (earliness) of detection of an outbreak has emerged as an important problem in public health [1].

One of several proposed strategies for improving the timeliness of detection is to identify new signals to monitor [1]. Of particular importance are the signals that occur early in the epidemic process. Sales of over-the-counter (OTC) healthcare products have potential as early indicators of disease outbreaks because symptomatic individuals often treat themselves with OTC products prior to (or even instead of) seeking medical assistance [2]. The development of nationwide systems that collect OTC sales data from retailers and make it

*Corresponding author. Tel.: +1-412-648-6740; fax: +1-412-802-6803; abc25@pitt.edu.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

available to organizations that conduct biosurveillance, such as NRDM [3], has significantly increased the availability of OTC sales data.

A fundamental research problem concerning the use of OTC sales in biosurveillance is to identify groups or categories of products that could be monitored for the detection of outbreaks of various diseases. The typical approach to this problem is to have human experts create aggregates using their domain knowledge. Several studies have looked into the problem of which expert-defined OTC categories can provide early warning of certain disease outbreaks. For example, Welliver et al. [4] found that sales of *cold remedies* (but not antipyretics) can provide early warning of Influenza B outbreaks, while Hogan et al. [5] found that sales of *electrolytes* can provide early warning of respiratory and diarrheal diseases in children.

More recent studies have begun to explore the use of data mining techniques for discovering new product categories that could lead to earlier outbreak detection. The motivation for employing data mining techniques to solve this problem stems from the belief that OTC sales data may contain various patterns—such as those due to marketing influences, product availability, and consumers' perception of which products are useful for certain symptoms and diseases—which may not have been taken into account in the domain knowledge-based construction of product categories. It is hoped that exploitation of such patterns could lead to discovery of new categories with superior outbreak detection performance relative to predefined categories. Ultimately, the product categories discovered through data mining techniques must also be understood from a domain perspective in order to be used effectively by biosurveillance consumers, primarily public health staff. Two previous studies have applied unsupervised clustering techniques to this problem. Magruder et al. [6] first manually grouped OTC products into 61 low-level categories based on age group, dose form and indication. Then, they formed time series for each of the 61 low-level categories and applied hierarchical clustering to group them into 16 categories of similar products. Wallstrom et al. [7] developed a Bayesian model for clustering OTC products and used Markov chain Monte Carlo simulation to learn the parameters of this model. This method was able to uncover a clinically significant distinction between OTC products intended for the treatment of allergy and OTC products intended for the treatment of cough, cold, and influenza symptoms.

Researchers have developed methods for unsupervised clustering of time series for applications in other domains, principally genomic and proteomic data analysis. Many authors, for example Eisen et al. [8] have utilized correlation-based distance functions and hierarchical clustering methods to cluster gene expression profiles. Other authors, for example Tseng [9] have extended classical clustering methods for genomic and proteomic applications. Bayesian approaches include Ramoni et al. [10], which developed a Bayesian method for unsupervised clustering that represents each time series as a Markov chain and clusters similar Markov chains, Ramoni et al. [11], which developed a Bayesian method based on autoregressive models for clustering gene expression profiles, and Medvedovic et al. [12], which developed a Bayesian method based on mixture models.

These existing methods are not compelling for aggregating OTC products in biosurveillance for two reasons. First, all of the above methods are unsupervised whereas disease data are often available for guiding the clustering process. Second, when OTC products are aggregated for biosurveillance, their total (summed) sales are monitored. This unique role of clusters suggests alternative methods. In this paper we propose a new method for grouping OTC products into categories. This method employs data on the activity of a specific disease to find groupings of products that have a strong association with that disease. To emphasize our main target application, i.e., mining of product categories for syndromic surveillance, we

describe the proposed algorithm in terms of OTC products and disease activity data. However, this algorithm could be used in any other domain.

Our algorithm takes two main kinds of inputs: (i) a *disease activity* time series; and (ii) a collection of OTC *product sales* time series (or more generally, any finely granular biosurveillance time series that may be aggregated to look for stronger outbreak signals). We assume that the disease time series contains data about a specific disease (e.g., influenza) and could be of several types, including hospital diagnoses, physician office or Emergency Department (ED) diagnoses, laboratory cultures and other laboratory tests. We also assume that the spatial and temporal resolution of the OTC sales and disease data match. In other words, these two kinds of data correspond to the same geographical area (e.g., a particular market) and have the same temporal granularity (e.g., both contain daily or weekly observations). Given these two inputs, our algorithm employs a novel application of linear regression to identify a group of products having a strong association with the disease time series. Our research hypothesis was that the product groups discovered by our algorithm could lead to earlier detection of disease outbreaks.

The remainder of the paper is structured as follows. Section 2 describes the proposed algorithm. Section 3 lays out the design of an evaluation experiment. Section 4 presents the experimental results. Section 5 discusses the findings of our study and outlines several directions for future work. Appendices A and B provide detailed descriptions of the algorithm.

2. Description of the aggregate-mining algorithm

Our main algorithm is called *Iterative Regression Based Aggregation* (IRBA) and is based on a repeated application of *Regression Based Aggregation* (RBA). Two concepts, namely *time-series aggregation* and *product aggregation*, play a key role in these algorithms. Here, the meaning of the term “aggregate” when applied to time series differs from the meaning of this term when applied to products. An *aggregate* of products is a set of products. *Aggregating* a group of time series means adding them up to form another time series, called the *aggregated time series* of the corresponding group of products. The algorithms RBA and IRBA primarily deal with sets of product aggregates; the member aggregates of any such set will be pair-wise disjoint.

From these remarks, it is clear that the term “aggregate” when applied to products has the same meaning as the widely used term “cluster”. We have opted in favor of using the former term in order to emphasize the fact that in our algorithm the aggregation of products (clustering) always occurs in parallel with the aggregation of their time series (summation).

2.1. Regression based aggregation

The algorithm RBA is based on a regression model specified as follows. Let D denote a disease time series and $\mathcal{P} = \{P_1, \dots, P_n\}$ denote a collection of time series corresponding to OTC product sales. All time series must have the same length k . We denote a set of product aggregates by $\mathcal{A} = \{A_1, A_2, \dots, A_L\}$, where each aggregate A_i contains the set of indices for the products in the aggregate. We denote the associated collection of aggregated time series by $\mathcal{T} = \{T_1, T_2, \dots, T_L\}$. Our model states that D follows a normal linear regression model with independent variables $T_1, T_2, \dots, T_L$. That is, $D_t = \alpha_0 + \alpha_1 T_{1,t} + \dots + \alpha_L T_{L,t} + \varepsilon_t$, for $t = 1, \dots, k$, where

$$T_{i,t} = \sum_{j \in A_i} P_{j,t}$$

and $\varepsilon_1, \dots, \varepsilon_k$ are independent and normally distributed with mean zero and variance σ^2 . The unknown model parameters include the set of aggregates $\mathcal{A}$, the regression coefficients $\alpha_0, \dots, \alpha_L$, and the error variance σ^2 .

The algorithm RBA searches over the space of possible sets $\mathcal{A}$ and for each candidate $\mathcal{A}$ estimates the regression coefficients and the error variance using normal linear regression. Initially, the set of aggregated time series $\mathcal{T}$ is equal to the input set $\mathcal{P}$ and the set of aggregates $\mathcal{A}$ is formed by placing each OTC product in a distinct aggregate. At each step, algorithm RBA attempts to modify the sets $\mathcal{A}$ and $\mathcal{T}$ in two stages. In the first stage, RBA tests the statistical significance of three basic operations: (i) *merge*, which consists of merging two existing aggregates, (ii) *split*, which consists of splitting one product from an existing aggregate, and (iii) *removal*, which consists of removing an existing aggregate. Each basic operation corresponds to a simple change in the underlying regression model. For example, merging aggregates A_1 and A_2 in the above model is equivalent to changing the regression model from $D_t = \alpha_0 + \alpha_1 T_{1,t} + \alpha_2 T_{2,t} + \alpha_3 T_{3,t} + \dots + \alpha_L T_{L,t} + \varepsilon_t$ to $D_t = \alpha_0 + \alpha_1^* (T_{1,t} + T_{2,t}) + \alpha_3 T_{3,t} + \dots + \alpha_L T_{L,t} + \varepsilon_t$. The significance of each basic operation is evaluated by computing the p-value for the associated model change. For the merge example, the p-value is computed for a null hypothesis that $\alpha_1 = \alpha_2$ (the merged model) against an alternative hypothesis that $\alpha_1 \neq \alpha_2$ (the original model). In the second stage, RBA selects one basic operation to carry out according to the following rule: if a split is determined to be significant (in terms of its p-value), then the best possible split is carried out; otherwise, if either a merge or a removal is determined to be significant, then the best one of these two operations is carried out. The algorithm RBA runs until no further significant modification of the set of aggregates $\mathcal{A}$ is found, or until it reaches a predefined maximum number of steps. RBA returns the final version of the set $\mathcal{A}$.

2.2. Iterative regression based aggregation

The use of linear regression as a subroutine in RBA has an important implication: the number of OTC products that can be supplied to RBA is bounded from above by the number of data points in each time series. While the number of available OTC products is more than seven thousands, the number of data points in the disease and OTC sales time series available to us is a small constant times the number of weeks in a year, because the data points correspond to weekly observations over a small number (less than 5) of years.

To scale up, we devised another algorithm, called Iterative Regression Based Aggregation (IRBA), which uses RBA as a subroutine. IRBA takes as its main inputs a disease time series and a (large) set of OTC time series. Then, it repeatedly selects a random “parent” subset of OTC products and applies the algorithm RBA to this subset and to the disease series to create a “child” set of products. This “child” set of products could be formed from the set of aggregates $\mathcal{A}$ in several ways. The most inclusive approach (and the one used in this paper) is to let the “child” set be equal to the union of all aggregates returned by RBA. Other less inclusive approaches, such as letting the “child” set equal the aggregate with the smallest p-value in the final linear regression model, may also be of interest.

Algorithm IRBA uses the parent/child pairs obtained after a suitably large number of repetitions to compute a measure of the strength of the association of each OTC product with the disease data. This measure, called the *retention rate*, is computed as follows. Let

$\mathcal{S}_1 = \{S_1^1, S_2^1, \dots, S_{reps}^1\}$ be the set of “parent” sets corresponding to the runs $1, \dots, reps$ for IRBA. Let $\mathcal{S}_2 = \{S_1^2, S_2^2, \dots, S_{reps}^2\}$ be the set of “child” sets corresponding to the runs $1, \dots, reps$. Then the retention rate $r(i, \mathcal{S}_1, \mathcal{S}_2)$ of a product i with respect to $\mathcal{S}_1$ and $\mathcal{S}_2$ is defined as the ratio between the number of sets S_i^2 containing i and the number of sets S_i^1 containing i . Algorithm IRBA returns the set $Top-\beta$ of products with retention rate greater than a threshold β . For convenience, we will occasionally assume that the output of the algorithm IRBA is the aggregated time series of the products in the set $Top-\beta$.

Pseudo-code descriptions of algorithms RBA, IRBA and their main sub-routines are given in Appendix A, while the techniques employed to set the values of the free parameters of these algorithms are discussed in Appendix B.

3. Design of the evaluation experiment

Our evaluation experiment applied the algorithm IRBA to a collection of time series containing weekly sales counts of more than four thousand OTC products and a time series of weekly laboratory data for influenza. In this section, we describe the datasets used in the evaluation experiment and define the metrics used to evaluate the performance of algorithm IRBA.

3.1. Datasets

The influenza dataset was provided by the Centers for Disease Control and Prevention. Through the National Respiratory and Enteric Virus Surveillance System (NREVSS) [13], laboratories report weekly detections of respiratory and enteric viruses. Laboratories participate voluntarily, and include university and community hospital laboratories, state and county public health laboratories, and commercial laboratories. Data were provided by NREVSS from July 5, 2003 to December 2, 2006. A focused analysis restricted the data to a hospital in the city of Houston, TX. This restriction was needed to match the spatial resolution of the influenza activity and the OTC sales data available to us. The influenza dataset contained weekly test counts for two different test types (antigen detection and virus isolation) as well as the corresponding positive counts for four influenza strains (A-H1N1, A-H3N2, A-Unknown, and B). We aggregated the test and positive counts across both test types and the positive counts across all four influenza strains. After this aggregation, the final laboratory data consisted of the weekly percent positive (computed as the total number of positive tests times 100, divided by the total number of tests). The percent of positive tests is commonly preferred to the absolute number because the total number of tests varies considerably throughout the year. The influenza time series is shown in Fig. 1. The annotations of the time series plotted in Fig. 1 show the dates when the peaks of positive counts were observed in each enclosed flu season.

The dataset of OTC product sales was provided by AC Nielsen Corp. This dataset consists of the weekly sales counts of 4483 distinct healthcare products sold between May 3, 2003 and April 22, 2006 in a region around the city of Houston, TX. AC Nielsen Corp. groups these products into eighteen different categories: Analgesic & Chest Rubs (ACR), Antacids (ANTA), Bronchial Remedies (BRON), Cold Remedies Adult (CRA), Cold Remedies Children (CRC), Cough Drops (CDRO), Cough Syrups & Tablets (CST), Diarrhea Remedies (DIAR), First Aid Hydrocortisones (HYDR), First Aid Thermometers (THM), Nasal Product Internal (NASA), Pain Remedies Alkalizing Effects (PRAL), Pain Remedies Arthritis (PRAR), Pain Remedies Children’s Liquid (PRCL), Pain Remedies Headache (PRHE), Sinus Remedies (SINU), Throat Lozenges (THRO), Vaporizing Products (VAPO).

The number of products in the OTC dataset belonging to each AC Nielsen category is shown in Fig. 2.

The common time period covered by the disease and OTC sales datasets extended from July 5, 2003 to April 22, 2006. After performing the trimming that was needed to align the OTC and disease time series, we obtained the final, cleaned-up version of the input data consisting of 4483 time series of OTC sales counts and one disease time series, each with 147 data points.

3.2. Evaluation metrics

First, we used the *detection algorithm method* [14] to compare the signal generated by IRBA with the time series obtained by aggregating over each AC Nielsen category. The detection algorithm method is used to compare the potential of two different signals for early outbreak detection. It is conducted by running an outbreak-detection algorithm on each signal and by evaluating the performance of the detection algorithm in each case through Activity Monitor Operating Characteristic (AMOC) analysis [15]. In this experiment, we used a CUSUM algorithm [16] with an 8-week moving average as a detection algorithm. The CUSUM algorithm employs as the alarm statistic the cumulative sum of positive deviations from the forecasted values. In our case, the forecasted values are computed with an 8-week moving average. For each desired *false alarm rate* (1/year, 2/year, and so on), a threshold value of the alarm statistic is learned from a non-outbreak portion of influenza data. At each threshold we calculate the *timeliness* of the detection algorithm by computing the difference in weeks between the date when the alarm statistic first exceeds the threshold and the date when the outbreak begins. An AMOC curve graphs the timeliness of the detection algorithm against the false alarm rate. One signal would be considered superior to the other if the timeliness of the detection algorithm with respect to the first signal is smaller than the timeliness with respect to the other signal, at every false alarm rate. If the two AMOC curves intersect at some point, determining the superior algorithm is harder and will depend on other factors, such as the cost of the false alarm and the benefit of the early detection.

In addition to evaluating the outbreak detection performance of the signal produced by our algorithm through AMOC analysis, we also computed several diagnostic metrics. We computed the *mean retention rate* over each AC Nielsen category to measure the extent to which the output of algorithm IRBA agrees with the medical knowledge about influenza. Next, we computed the *correlation* of the signal produced by IRBA and of each signal obtained by aggregating over an AC Nielsen category with the disease time series. Finally, we computed the *mean pair-wise correlation* between the products generated by IRBA and between the products in each AC Nielsen category.

In order to simplify the presentation of our experimental results, we have identified six of the eighteen AC Nielsen categories—ANTA, BRON, DIAR, HYDR, PRAL, PRAR—which we would a priori expect to have little or no association with influenza. We refer to the products belonging to these six categories as “NON-FLU” products. We would expect the products in the remaining twelve categories to be more correlated with influenza than the “NON-FLU” products and we refer to them as “FLU” products. By the design of our algorithm, we would expect the NON-FLU categories to have mean retention rates that are lower than the retention rates of the FLU categories. Likewise, we would expect the aggregated time series of the products with high retention rate in any OTC category to be more strongly correlated with the disease time series than the aggregated time series of the whole OTC category.

We emphasize that this specific division of the predefined categories into two groups is only used to make the presentation clearer and does not affect our conclusions.

3.3. Training and test data for the AMOC analysis

To apply the outbreak detection method to our input data, we first created two training sets: T_{03-04} and T_{04-05} . The first training set was obtained by removing from each input time series the data points corresponding to the 2003–2004 outbreak, whereas the second training set was obtained by removing from each input time series the data points corresponding to the 2004–2005 outbreak. In each of the two cases above, 37 data points were removed, resulting in the removal of approximately 9 weeks before and after each outbreak in addition to the removal of outbreak weeks. By supplying each of these two training sets to the algorithm IRBA, we obtained two different sets of “top” products: $Top-\beta_{0304}$ and $Top-\beta_{0405}$.

Then, we generated one set of AMOC curves by supplying to CUSUM the aggregated time series of the products in the set $Top-\beta_{03-04}$ and by using the 2003–2004 outbreak as the “gold standard” for recognizing false alarms, and a second set of AMOC curves by supplying to CUSUM the aggregated time series of the products in the set $Top-\beta_{04-05}$ and by using the 2004–2005 outbreak as the “gold standard” for recognizing false alarms. The 2005–2006 outbreak was not used analogously to obtain a third set of AMOC curves because some of the data points at the end of this outbreak were trimmed in order to align the clinical and OTC time series (as described in Section 3.1).

4. Results

4.1. AMOC analysis

Figure 3 compares the performance of the signal extracted by our algorithm with the performance of the signals obtained by aggregating over each of the top six FLU categories in terms of retention rate (PRCL, THM, CST, PRHE, CRC, CRA). Figure 3(a) shows the set of AMOC curves obtained by feeding to CUSUM the signal produced by algorithm IRBA taking as input the training set T_{03-04} (see Section 3.3) and by using the flu outbreak of the 2003–2004 season (Fig. 1) as the “gold standard” for recognizing false alarms. Figure 3(b) shows the set of AMOC curves obtained by feeding to CUSUM the aggregated time series produced by algorithm IRBA taking as input the training set T_{04-05} and by using the flu outbreak of the 2004–2005 season as the “gold standard” for recognizing false alarms. All AMOC curves shown in Fig. 3 were obtained with β set at 0.75.

First, it should be noticed that all seven signals perform better in detecting the 2003–2004 outbreak than in detecting the 2004–2005 outbreak. This change in performance is to be expected because the 2003–2004 outbreak was much larger than the 2004–2005 outbreak (see Fig. 1). With respect to the 2003–2004 outbreak, the performance of the signal produced by our algorithm is comparable to the performance of the signal obtained by aggregating over the category PRCL and superior to the performance of the remaining five signals.

With respect to the 2004–2005 outbreak, the performance of the signal produced by our algorithm is comparable to the performance of the signal obtained by aggregating over the category THM and superior to the performance of the remaining five signals. Finally, we note that the performance of the signals obtained by aggregating over each of the twelve categories that are not shown in Fig. 3, was clearly inferior to the performance of the signal extracted by IRBA.

4.2. Mean retention rate

Table 1 gives the mean retention rate and a 95% confidence interval for each AC Nielsen category.

As expected, the FLU categories generally have higher mean retention rate than the NON-FLU categories: among the top seven categories, there are six FLU categories (PRCL, THM, CST, PRHE, CRC, CRA) and only one NON-FLU category (HYDR). Interestingly, the categories of Pain Remedies Children Liquids (PRCL) and Thermometers (THM) have mean retention rates which are much higher than the mean retention rates of the remaining categories (z-scores for the mean retention rates of PRCL and THM categories are 2.7 and 2.1, respectively, while the z-score for the third largest mean retention rate is 0.3). This finding becomes more interesting considering that a similar result was obtained in a previous application of our algorithm to a dataset that belonged to a different region (Southwestern Pennsylvania) and contained a different type of disease time series (emergency department visits for influenza-like illness) [17].

It was unexpected that the category of Hydrocortisones (HYDR)—which a priori would be expected to have no association with influenza—was ranked near the top (it has the fourth highest mean retention rate). By closely examining the input data and the heuristics used in our algorithm, we have derived a plausible explanation for this result (see Section 5).

4.3. Correlation with disease time series and mean pairwise correlation

Table 2 shows the product count, the correlation with the disease time series and the mean pair-wise product correlation for the AC Nielsen categories THM, PRCL and for three sets of products generated with algorithm IRBA: *Top-0.75* (the set of products with retention rate greater than or equal to 0.75), *Top-0.85*, and *Top-0.95*.

All three sets generated with our algorithm have, in aggregate, stronger correlation with the disease time series than the top two predefined categories THM and PRCL. In addition, the sets *Top-0.85* and *Top-0.95* exhibit a higher mean pair-wise correlation than categories THM and PRCL. Table 3 gives a more complete picture of the output of our algorithm.

This table shows, for each AC Nielsen category, the product count and the correlation with the disease time series of the whole category and three sub-categories: the sub-category *Top-0.75* (consisting of the products in that category with retention rate greater than or equal to 0.75), the sub-category *Top-0.85* and the sub-category *Top-0.95*. The results in Table 3 show that for each AC Nielsen category (except the category VAPO, which has the lowest product count) there is a significant increase in the correlation with the disease data going from the whole category to the *Top-0.75* sub-category. In particular, the average increase in correlation over all FLU categories, going from the whole category to the *Top-0.75* subcategory, is nearly 40%. This result agrees well with our expectation that the algorithm IRBA extracts from each category a group of products that are more strongly associated with the disease data than the whole category.

From Table 3 it can be seen that the output of algorithm IRBA contains a certain amount of noise: the sets *Top-0.75*, *Top-0.85* and *Top-0.95* contain about 18%, 16% and 11% NON-FLU products, respectively.

5. Discussion

We have proposed a novel aggregate-mining algorithm and have demonstrated that in terms of the outbreak detection performance, the product category found by this algorithm is comparable to categories PRCL, THM and superior to the remaining predefined categories. Further, the discovered product category compares favorably with the predefined product categories in terms of the correlation with the disease data and the mean pair-wise product correlation. In this section, we discuss the limitations of the proposed method and our

experiment, propose ways of overcoming these limitations and outline several directions for future research.

5.1. Limitations of methods and datasets

The proposed algorithm conducts a greedy search over the huge space of possible product groupings. Hence, the output of this algorithm is not guaranteed to be optimal. Further, as reported in the previous section, the output of our algorithm contains a certain amount of noise (i.e., products that end up in the final set for reasons other than their strong association with the disease time series). Our investigation indicates that this noise arises due to a combination of two main factors: (i) an undesired increase in the correlation with the disease time series resulting from the aggregation of products with time-shifted seasonal patterns; (ii) inflated standard errors of the linear regression coefficients, due to the presence of multicollinearity among the OTC time series. Next, we elaborate on these two factors by revisiting the unexpectedly high retention rate of the category of Hydrocortisones.

It is common medical knowledge that the occurrence of diseases for which HYDR products are used escalates during the summer months, whereas the flu seasons typically occur in winter with some overlap in late fall and early spring. Thus, a time lag between the seasonal periodicity of the HYDR and FLU time series is to be expected. As illustrated in Fig. 4, our input data meets this expectation. Figure 4 plots the (z-normalized) disease time series, the aggregated time series of the category HYDR and the aggregated time series of the category PRCL.

This observation coupled with the fact that time series aggregation in algorithm RBA is performed in parallel with the merging of product aggregates, implies that the insertion of a HYDR product into an aggregate that is mainly composed of FLU products, has the effect of “flattening” the off-outbreak portions of that aggregate’s time series. Since our disease time series is flat between the outbreaks (due to zero positive lab results), this “flattening” should lead to an increased correlation between the time series of the aggregate and the disease time series. To verify this intuition, we selected at random 1000 aggregates, each belonging to the final aggregation of a different run of the algorithm RBA and containing at least one product from the HYDR category and at least one product from either PRCL or THM category. Then, for each selected aggregate, we measured the correlation between the disease time series and (a) the aggregated time series over the products remaining after the removal of HYDR products from the aggregate; (b) the time series of the whole aggregate. We found that the mean correlation in the first case was 0.659 (95% CI: 0.655–0.663), while in the second case it was 0.672 (95% CI: 0.669–0.675), which corroborates our prediction.

In our application, this “flattening” is an undesired effect because the domain knowledge strongly suggests that hydrocortisones would not be taken to treat influenza, and hence public health should not monitor sales of hydrocortisones to detect influenza. A simple method for preventing such undesired effects would be to limit the set of input OTC time series by excluding certain categories known to have little or no association with the disease (such as NON-FLU products for the case of influenza).

The above discussion still does not explain the reason why FLU and NON-FLU products initially get mixed together. We believe that this mixing happens due to the presence of multicollinearity between OTC time series. Multicollinearity between explanatory variables is said to exist when these variables are correlated among themselves. Correlations among OTC time series arise primarily because of the seasonality component they contain. As discussed in depth in Appendix B, multicollinearity is more severe in the initial steps of algorithm RBA (when the number of explanatory variables is larger). It is well known that under severe multicollinearity, the linear regression coefficients are subject to inflated

standard errors. This in turn causes algorithm RBA to perform (especially in its initial steps) some merges between dissimilar products (such as, FLU and NON-FLU ones).

One way of reducing multicollinearity among OTC time series would be to remove the seasonality component from each of them. To remove seasonality, we intend to explore the use of wavelet transforms, as done for example in [18].

Several other characteristics of the data affect the performance of algorithm IRBA. This algorithm would benefit from input data that is more finely grained (e.g., if the data points represented daily rather than weekly observations) and contains a larger number of outbreaks. Improved performance could also result from disease data that is aggregated over several laboratories (and thus is less noisy). Aggregation of influenza data over two test types (antigen detection and virus isolation) in our present experiments may have reduced the strength of the disease signal. Hence, it would also be interesting to repeat the evaluation experiment by limiting the analysis only to the antigen detection test type, which is considered more sensitive than virus isolation and is more widely used.

Finally, we discuss a limitation in our AMOC analysis. As explained in Section 4.1, for each given false alarm rate, a threshold value of the alarm statistic was derived from influenza data. Deriving the thresholds corresponding to different false alarm rates requires the specification of a non-outbreak period. This, in turn, requires a method for determining the outbreak onset. For the case of influenza, determination of outbreak onset is a very difficult problem. All existing methods to solve this problem incorporate some form of subjective determination, typically done by public health staff. For our experiment, we used simple inspection of the influenza series to label each data point (week) as being an outbreak or a non-outbreak one. This method clearly introduces a subjective bias in the computation of the timeliness corresponding to each given false alarm rate (or, threshold). However, since the same labeling of influenza series was used for both signals being compared (e.g., THM vs. Top-0.75), we would expect that the comparison results are not significantly affected by the specific method we employed for determining the outbreak onset. This hypothesis can be rigorously answered by conducting a sensitivity analysis with respect to the method for determining the outbreak onset. We have left this sensitivity analysis as future work.

5.2. Applications of the aggregate-mining algorithm

The main target application of algorithm IRBA is mining of product categories for syndromic surveillance. The AMOC analysis in Section 4.1 indicates that our algorithm holds promise for discovering categories that could lead to early outbreak detection. It is natural to expect that the categories with the greatest potential for early detection would be discovered by taking into account lagged variables. The simplest way of applying IRBA with lagged variables would be to shift the disease time series to the left by a desired time lag before running the algorithm. Doing that for each of several time lags (e.g., 1, 2, ..., 7 days) could result in the discovery of categories that lead the disease signal by that lag. It is likely, however, that different products may have different lags of maximal effect. Hence, this simple proposed lagging scheme may not be sufficient and other more elaborate schemes may be required.

The time resolution of data becomes a critical issue for using our method with lagged variables. With weekly data, such as the one used in the our experiment, it is only possible to search for products groups leading the laboratory data by seven days. Although the existence of such products can not be excluded in advance (see [2]), it is likely that the best “early” product groups would lead laboratory data by fewer days (say, two-three days). Therefore, OTC and disease data with daily resolution become highly desirable for this

application. While daily OTC sales data are available at national level (e.g., through NRDM [3]), acquisition of daily disease data is more challenging.

Our future work will explore the discussed ways of improving the performance of the aggregate-mining algorithm and will further investigate its application in discovering product categories for syndromic surveillance.

Acknowledgments

This research was supported by the National Library of Medicine grant 5R21LM008278. We thank the CDC/NREVSS and the AC Nielsen Corp. for providing the datasets used in this study and the two anonymous reviewers for their particularly insightful suggestions to improve the paper's content.

References

1. Wagner MM, Tsui FC, Espino JU, Dato VM, Sittig DF, Caruana RA, McGinnis LF, Deerfield DW, Druzdzal MJ, Fridsma DB. The emerging science of very early detection of disease outbreaks. *J Public Health Manag Pract* 2001;7:51–59. [PubMed: 11710168]
2. Hogan, WR.; Wagner, MM. Sales of over-the-counter healthcare products. In: Wagner, MM.; Moore, AW.; Aryel, RM., editors. *Handbook of Biosurveillance*. Amsterdam; Boston: Academic Press; 2006.
3. Wagner MM, Robinson JM, Tsui FC, Espino JU, Hogan WR. Design of a national retail data monitor for public health surveillance. *J Am Med Inform Assoc* 2003;10:409–418. [PubMed: 12807802]
4. Welliver RC, Cherry JD, Boyer KM, Deseda-Tous JE, Krause PJ, Dudley JP, Murray RA, Wingert W, Champion JG, Freeman G. Sales of nonprescription cold remedies: a unique method of influenza surveillance. *Pediatr Res* 1979;13:1015–1017. [PubMed: 503653]
5. Hogan W, Tsui FC, Ivanov O, Gesteland PH, Grannis S, Overhage JM, Robinson JM, Wagner MM. Detection of Pediatric Respiratory and Diarrheal Outbreaks from Sales of Over-the-counter Electrolyte Products. *J Am Med Infor Assoc* 2003;10:555–562.
6. Magruder SF, Lewis SH, Najmi A, Florio E. Progress in understanding and using over-the-counter pharmaceuticals for syndromic surveillance. *MMWR Morb Mortal Wkly Rep* 2004;53:117–122. [PubMed: 15714640]
7. Wallstrom G, Hogan W. Unsupervised Clustering of Over-the-counter Healthcare Products into Product Categories. *Journal of Biomedical Informatics* 2007;40:642–648. [PubMed: 17509942]
8. Eisen M, Spellman P, Brown P, Botstein D. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences USA* 1998;95:14863–14868.
9. Tseng GC. Penalized and weighted K-means for clustering with scattered objects and prior information in high-throughput biological data. *Bioinformatics*. 2007 (to appear).
10. Ramoni M, Sebastiani P, Cohen P. Bayesian clustering by dynamics. *Machine Learning* 2002;47:91–121.
11. Ramoni M, Sebastiani P, Kohane IS. Cluster analysis of gene expression dynamics. *Proceedings of the National Academy of Sciences USA* 2002;99:9121–9126.
12. Medvedovic M, Yeung KY, Bumgarner RE. Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics* 2004;20:1222–1232. [PubMed: 14871871]
13. <http://www.cdc.gov/ncidod/dvrd/revb/nrevss/>, web site.
14. Tsui FC, Wagner MM, Dato VM, Chang CC. Value of ICD-9 coded chief complaints for detection of epidemics. *AMIA Ann Symp Proc* 2001:711–715.
15. Fawcett, T.; Provost, F. Activity monitoring: noticing interesting changes in behavior. *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'99)*; San Diego, California, United States: ACM Press; 1999. p. 53–62.
16. Page ES. Continuous Inspection Schemes. *Biometrika* 1954;41:100–115.
17. Hogan WR, Wallstrom GL, Li R. Identifying categories of over-the-counter products with superior outbreak detection performance. *Advances in Disease Surveillance* 2007;2:13.

18. Zhang, J.; Tsui, F-C.; Wagner, MM.; Hogan, WR. Detection of Outbreaks from Time Series Data Using Wavelet Transform. Proceedings of the AMIA 2003 Symposium.; 2003. p. 748-752.
19. Jain AK, Murty MN, Flynn PJ. Data Clustering: A Review. ACM Computing Surveys 1999;31:264–323.
20. Schwarz G. Estimating the dimension of a model. Annals of Statistics 1978;6:461–464.
21. Kutner, MH.; Nachtsheim, CJNJ.; Li, W. Applied Linear Statistical Models. 5. Irwin: McGraw-Hill; 2004.
22. Belsley, DA.; Kuh, E.; Welsch, RE. Regression Diagnostics: Identifying Influential Data and Sources of Collinearity. New York: John Wiley & Sons; 1980.

Appendix A. Detailed description of algorithms RBA and IRBA

1. Algorithm RBA

The pseudo-code description of algorithm RBA and its main subroutines is given in Figs. 5–8.

The two main inputs of RBA (Fig. 5) are the disease time series D and the set $\mathcal{P}$ containing the OTC sales time series. In the pseudo-code in Fig. 5 it is assumed that the aggregation of time series (summation) is performed by calling the procedure AGGREGATE, whose pseudo-code has been omitted due to its simplicity. As described in the main text (Section 2.1), algorithm RBA iterates over a two-stage process, which consists of first selecting a basic operation (merge, split, or removal) and then carrying out this operation. To carry out the operation selected in each step, RBA calls one of the procedures DO-SPLIT, DO-MERGE, and DO-REMOVE. Due to the simplicity of these procedures, we have omitted their pseudo-code. Further, we have assumed the availability of the procedure FIT-LM, which fits a linear regression model and returns a data structure M that encapsulates all the fields provided by the linear regression routines in standard statistical packages (including the estimated regression coefficient vector α , the p-value vector $\mathbf{p}$, the t-value vector $\mathbf{t}$, the covariance matrix $\mathbf{Cov}$, the residual standard error σ , and the residual degrees of freedom df).

The procedure TRY-MERGE, shown in Fig. 6, aims to find a pair of aggregates whose merging would lead to the greatest improvement in the linear model. To do so, TRY-MERGE tests for each pair of aggregates A_i and A_j the hypothesis $H_0: \alpha_i = \alpha_j$, $H_1: \alpha_i \neq \alpha_j$, where α_i and α_j are the coefficients of T_i and T_j , respectively, in the current linear model M . We call the p-value of this test the *merge p-value* associated with the operation of merging aggregates A_i and A_j .

Note that the larger the merge p-value, the weaker the evidence against the null hypothesis. Hence, a large merge p-value is considered by TRY-MERGE as evidence that the coefficients α_i , α_j are very close (although technically a large p-value could also arise due to the low power of the statistical test). TRY-MERGE returns to the calling procedure RBA the *maximum* merge p-value over all pairs of aggregates and a pair of aggregates that achieves this maximum. In turn, RBA considers the merge to be significant if the maximum merge p-value is greater than the threshold *p-cutoff*.

The procedure TRY-REMOVAL, shown in Fig. 7, aims to find an aggregate whose time series has the least association with the disease time series. To do so, for each aggregate A_i , TRY-REMOVAL tests the hypothesis $H_0: \alpha_i = 0$, $H_1: \alpha_i \neq 0$. We call the p-value of this test the *removal p-value* associated with the operation of removing aggregate A_i . Note that the larger the removal p-value, the weaker the evidence against the null hypothesis. Hence, a large p-value is considered by TRY-REMOVAL as evidence that the time series of the aggregate A_i has little association with the disease time series.

TRY-REMOVAL returns to the calling procedure RBA the *maximum* removal p-value over all aggregates and an aggregate that achieves this maximum. In turn, RBA considers the removal operation to be significant if the maximum removal p-value is greater than the threshold *p-cutoff*.

The procedure TRY-SPLIT, shown in pseudo-code in Fig. 8, aims to find in each aggregate a product whose time series is most “dissimilar” to the aggregated time series of the remaining products in that aggregate. To do so for an arbitrary aggregate A_i , TRY-SPLIT temporarily removes from A_i each product j and after each removal it adjusts the current version of the set T in two ways. First, it subtracts the time series of product j from the aggregated time series T_i . Second, it inserts the time series of product j into the set T . After this adjustment (performed by calling the procedure ADJUST whose pseudo-code has been omitted), TRY-SPLIT fits a new linear regression model M' of the time series D versus the adjusted set of time series T' . Next, TRY-SPLIT tests the hypothesis $H_0: \alpha_i = \alpha_{L+1}$, $H_1: \alpha_i \neq \alpha_{L+1}$, where α_i and α_{L+1} denote the coefficients in the model M' of the adjusted time series T'_i and the time series of the product j , respectively. We call the p-value of this test the *split p-value* associated with the operation of splitting product j from the aggregate A_i . Note that the smaller the split p-value, the stronger the evidence against the null hypothesis, i.e., the more dissimilar the product j is to the remaining products in the aggregate A_i .

After computing the split p-value for each aggregate and each product in that aggregate, TRY-SPLIT returns to the calling procedure RBA the *minimum* split p-value over all aggregate/product pairs and an aggregate/product pair that achieves this minimum. In turn, RBA considers the split to be significant if the minimum split p-value is smaller than the threshold *p-cutoff*. Note that an exhaustive search for the minimum p-value over all possible ways of partitioning an aggregate into two subsets is not computationally feasible; this is the reason why TRY-SPLIT only tries splitting a single product from each aggregate.

Next, we take a close look at the relationship between RBA and the well-known method of hierarchical clustering [19]. Hierarchical clustering is an unsupervised clustering technique that can be performed in either *agglomerative* or *divisive* fashion. In the former case, each of the units to be clustered is initially placed in a distinct singleton cluster and each step merges clusters together, until a stopping criterion is satisfied. In the latter case, all units are initially placed together in a single cluster and each step performs cluster splitting until a stopping criterion is satisfied. The decision on how to perform the merging (splitting) is based on various similarity (dissimilarity) measures. The similarity between two units is typically computed as a distance measure (e.g., Euclidean distance) between the vectors representations of these two units. Different techniques can be employed to compute the distance between two clusters; in the *centroid* agglomeration method the distance between two clusters is equal to the distance between their centers.

From these remarks, it is clear that RBA is to some extent analogous to the Hierarchical Agglomerative Clustering (HAC) with centroid agglomeration. Indeed, RBA begins with each product in its own aggregate (or, cluster) and one of the possible operations performed in each step is the merging of “similar” aggregates. Moreover, the time series of an aggregate of products differs from the center of the individual time series of those products only by a constant factor. However, RBA differs from HAC in that it alternates merging of products with two other basic operations (removal and splitting) and in that the similarity measures in RBA are based on the significance levels (p-values) of appropriate statistical tests rather than distances.

The mixing of three basic operations in RBA raises the issue of how to choose one of them to perform in each step. Note that the operations of merge and removal are analogous in that

both are considered statistically significant for large p -values. Hence, when choosing between these two operations, RBA simply picks the one with the largest p -value. The split operation is different from the other two in that it is considered significant for small p -values. Hence, comparing the significance of split with the significance of either merge or removal is not as simple as comparing the significance of merge and removal. However, encountering a step in which a split operation is statistically significant means that the algorithm has found new evidence that calls for the reversal of a previous action. Therefore, RBA considers splitting to have a higher priority than the other two operations and always carries it whenever the split p -value is smaller than the p -cutoff threshold.

2. Algorithm IRBA

The pseudo-code for IRBA is shown in Fig. 9.

As discussed in the main text (Section 2.2), this algorithm was designed to scale the computation up to thousands of OTC time series. It repeatedly calls algorithm RBA with a randomly selected, small “parent” set of products to produce a “child” set of products. The parent/child pairs generated over a sufficiently large number of repetitions are used to compute the retention rate of each product. Algorithm IRBA returns the set of products with retention rate greater than or equal to a threshold β .

Appendix B. Setting the free parameters of the algorithm and implementation

1. Choice of parameter “ p -cutoff” and “max-steps”

The value of the parameter p -cutoff determines whether or not each of the three basic operations tested by the algorithm RBA is to be considered statistically significant. Clearly, this parameter must lie in the interval (0,1). For our experiment we set p -cutoff at 0.1, a threshold value commonly used in hypothesis testing. We leave the full sensitivity analysis for this parameter as future work.

The value of the parameter max -steps is the maximum number of steps that will be tried by the algorithm RBA, unless it encounters a step in which none of the three basic operations is found to be significant. Note that if there were no splitting, it would suffice to set max -steps equal to the number of products supplied to RBA (parameter m), which must be smaller than the length of the time series (which is 147 in our experiment). This is the case because in each step of RBA the number of aggregates would be decremented either due to the merging of two aggregates or due to the removal of an aggregate. With the split operation present, the algorithm RBA may run for a number of steps which is greater than m . Our experiments indicate that with p -cutoff fixed at 0.1, the split operation is chosen over merge/removal less than 10% of the time. Hence, the expected number of iterations until RBA stops should be smaller than $(m + 0.2m)$, i.e., smaller than 200. We set the value of max -steps at 10,000—a value much larger than the expected number of iterations—to make it very unlikely for the algorithm RBA to stop because of reaching max -steps iterations. With max -steps set at 10,000, in our experiment the algorithm RBA always stopped due to encountering a step in which none of the basic operations was found significant.

2. Choice of parameter “ m ”

The value of the parameter m determines the cardinality of the random “parent” set supplied to the algorithm RBA during an iteration of the algorithm IRBA. As we explain next, choosing an optimal value for this parameter strongly depends on the characteristics of the

OTC input data. We have examined several of these characteristics for the particular OTC input data used in our experiment. Below we explain how we arrived at a value m which is a compromise between two undesirable extremes.

First, the value of parameter m must be smaller than the number of data points in the input time series, because RBA uses linear regression as a sub-routine. The length of the time series used in our experiment (obtained after removing the data points of one outbreak) was 110. This value could be seen as an upper bound for m .

Second, we looked at how the choice of m affects the goodness-of-fit of a linear regression model that fits the disease time series against the time series of m OTC products. We used *Bayes Information Criterion* (BIC) [20] as the goodness-of-fit measure. We employed the definition of BIC implemented in the package R, which can be written as:

$$\text{BIC} = -2\log(\mathcal{L}) + m\log(k),$$

where $\mathcal{L}$ denotes the likelihood function of the model evaluated at the maximum likelihood estimators of its parameter vector, m denotes the number of parameters and k denotes the number of observations. According to this definition, a smaller BIC score indicates a better fit of the model. Other slightly different definitions of BIC (e.g., multiplying the right-hand side of the expression above by -1) are sometimes used by other authors.

To estimate the BIC score for different values of m , we varied m from 2 to 110 in increments of 2 and for each fixed value of m we ran an iterative procedure to estimate the BIC score of a linear model with m independent variables. Each repetition of this estimating procedure samples (without replacement) a subset of m OTC products from the given set of 4483 products and computes the value of BIC for the selected sample. For a fixed m , the number of iterations in the BIC-estimation procedure was chosen large enough to ensure that each OTC product occurs in at least 100 samples. Fig. 10 shows the plot of BIC versus m . As highlighted in the inset of Fig. 10, the BIC score is optimized at $m = 10$. This value could be viewed as a lower bound for the value of parameter m to be chosen for our experiment. Indeed, it would be desirable that the sequence of linear models obtained in each run of RBA is increasing with respect to the goodness-of-fit, i.e., the sequence of BIC scores corresponding to successive linear models tends toward the optimal value.

Next, we investigated how the choice of m affects the *multicollinearity* among the explanatory variables in the linear regression model fitted in the algorithm RBA, and the *coefficient of determination* R^2 of this model. Multicollinearity between explanatory variables is said to exist when these variables are correlated among themselves. In the case of OTC product sales, we would expect such correlations to arise due to seasonality effects. It is well-known that under severe multicollinearity the regression coefficients may be subject to large round-off errors as well as large sampling variances [21]. Because the estimated linear regression coefficients play a key role in the algorithm RBA, and because RBA is a step-wise procedure, it is crucial that the value of m is small enough to ensure that the multicollinearity in the initial linear model is reasonably mild.

Several metrics for detecting the presence of multicollinearity, such as the *condition number*, *condition indexes*, *variance decomposition proportions*, and *variance inflation factors*, have been proposed [21,22]. Variance inflation factors (VIF) measure how much the variances of the estimated coefficients are inflated as compared to when the explanatory variables are uncorrelated. A value of 1 means that there is no variance inflation due to multicollinearity; a value of 10 is often taken as evidence that multicollinearity may be unduly influencing the coefficient estimates [21]. We used the mean variance inflation factor

$\overline{VIF}$ (over all linear regression coefficients) as a measure of the multicollinearity among the sales time series of OTC products.

Another factor to be taken into account when choosing the value of parameter m is that the coefficient of determination R^2 (the proportion of the variance in the disease data explained by the m OTC explanatory variables) in the initial linear model should be reasonably large to ensure that the initial fitted model is not of very low quality. Again, this requirement is imposed by the step-wise nature of algorithm RBA.

We used simulation in a manner similar to the case of BIC described earlier, to estimate $\overline{VIF}$ and R^2 for m between 10 and 110 (the lower and upper bounds of m derived earlier). Our simulation showed that for $m = 10$, $R^2 \approx 0.29$ while $\overline{VIF} \approx 1.4$, whereas for $m = 110$, $R^2 \approx 0.9$ while $\overline{VIF} \approx 11.4$. Clearly, neither of these two extremes is a good choice of m for our algorithm. However, as shown in Fig. 11, there is a range of m in which multicollinearity is reasonably mild while the coefficient of determination is reasonably large. Figure 11 plots the *normalized* $\overline{VIF}$ (obtained by dividing each $\overline{VIF}$ value by the maximum value of $\overline{VIF}$) and R^2 , versus m .

At this point, it would seem reasonable to attempt to define a risk function which is a linear combination of $\overline{VIF}(m)$ and $(1-R^2(m))$ and then to set m at the value that minimizes the risk. However, in this approach it is hard to determine what the relative weights of $\overline{VIF}$ and $(1-R^2)$ should be.

For our evaluation experiment we set the parameter m at 50—a “compromise” value which is far from the two undesirable extremes $m = 10$ and $m = 110$. For this choice of m , we have $\overline{VIF} \approx 2.5$ (i.e., on average, in the initial linear model the variance of an estimated regression coefficient is inflated by a factor of 2.5) and $R^2 \approx 0.7$ (i.e., on average, in the initial linear model nearly 70% of the variance in the disease data is explained by the OTC products present in the model).

3. Choice of parameter “reps”

Having fixed the value of parameter m , we determined the value of parameter *reps*—the number of repetitions in the algorithm IRBA—as follows.

Assume that the true retention rate of a product X (computed over all samples of m products containing X) is $r(X) = p$. To estimate the true retention rate of product X , algorithm IRBA draws s samples uniformly at random from the space of all samples that contain product X . For each such sample, algorithm IRBA calls algorithm RBA to determine whether or not X is to be retained. Let the binary variables $Y_1, Y_2, \dots, Y_s$ denote, respectively, the results of running algorithm RBA on the samples $1, \dots, s$. Algorithm IRBA considers $Y_1, Y_2, \dots, Y_s$ to be IID observation from Bernoulli(p) and uses the sample mean

$$\overline{Y}_s = \widehat{p}_s = \frac{1}{s} \sum_{i=1}^s Y_i$$

as the estimator of parameter p .

With the above notation, we used the following procedure to set the value of parameter *reps*:

1. Let ε and α be given real numbers in $(0,1)$;

2. Let s be the smallest integer such that for every value of p ,

$$\mathbb{P}(\overline{Y}_s \in (p - \varepsilon, p + \varepsilon)) \geq 1 - \alpha.$$

3. Choose $reps$ large enough to guarantee that the expected number of “parent” sets containing each product is at least s . It’s easy to show that, setting $reps$ at

$$reps = \frac{n \times s}{m}, \quad (*)$$

suffices for that purpose.

For the experiment described in the main text, we chose $\varepsilon = 0.1$ and $\alpha = 0.05$. Then, employing the usual approximation of the distribution of $\overline{Y}_s$ with the normal distribution $N(p, \widehat{se})$ where

$$\widehat{se} = \sqrt{\widehat{p}_s(1 - \widehat{p}_s)/s},$$

and the fact that the expression $p(1-p)$ is maximized at $p = 0.5$, we got $s \approx 100$. Finally, using the equation (*), we obtained $reps \approx 9,000$.

4. Choice of parameter “ β ”

The value of the parameter β is the threshold above which the retention rate of a product is considered to be “high” by the algorithm IRBA. In our experiments we have found that, qualitatively, the results remain unchanged for values of β greater than 0.75. However, as β increases, some of the conclusions may become stronger. In Section 4 we have reported the results for three values of parameter β (0.75, 0.85, and 0.95) if there was significant change between these three cases, and only for $\beta = 0.75$, otherwise.

5. Implementation and running time

Algorithms RBA and IRBA were implemented in the language R. The experiment was carried out in a server with an Intel Pentium D 3.4 GHz processor running Red Hat Linux. In this machine and with the parameter values specified above, the procedure RBA took approximately 10 seconds per run. Hence, the execution of the procedure IRBA (9000 runs of RBA) was completed in about 25 hours.

Fig. 1.

Plot of the percent influenza positive time series from a single laboratory.

Fig. 2.
The number of OTC products belonging to each of the 18 AC Nielsen-defined categories.

Fig. 3.

AMOC curves obtained by running a CUSUM algorithm on the signal generated by algorithm IRBA with $\beta = 0.75$ and on the signals obtained by aggregating over six FLU categories: PRCL, THM, CST, PRHE, CRC, CRA. For the six plots in (a) the training and detection were done with respect to the 2003–2004 outbreak; for the six plots in (b) the training and detection were done with respect to the 2004–2005 outbreak.

Fig. 4.
Plot of the z-normalized disease, HYDR, and PRCL time series.

PROCEDURE: RBA

INPUT:

k : Time series length
 n : Number of OTC healthcare products
 D : A disease time series
 $\mathcal{P}$: A set of n OTC time series
 $p\text{-cutoff}$: The cut-off value for the statistical significance
 max-steps : The maximum number of steps

OUTPUT:

$\mathcal{A}$: A set of product aggregates

BEGIN

/ Place each product in a distinct aggregate */*

FOR $i = 1$ TO n DO

$A_i = \{i\}$

$\mathcal{A} = \{A_1, \dots, A_n\}$

$\text{step-number} = 0$

$\text{stop} = \text{FALSE}$

WHILE $\text{step-number} \leq \text{max-steps}$ AND $\text{stop} = \text{FALSE}$ DO

$\text{step-number} = \text{step-number} + 1$

$L = \text{LENGTH}(\mathcal{A})$

FOR $i = 1$ TO L DO

$T_i = \text{AGGREGATE}(\mathcal{P}, A_i)$

$\mathcal{T} = \{T_1, \dots, T_L\}$

/ Fit D against the time series in the set $\mathcal{T}$ */*

$M = \text{FIT-LM}(D = \alpha_0 + \alpha_1 T_1 + \dots + \alpha_L T_L + \epsilon)$

$\{\text{split-pval}, \text{split-info}\} = \text{TRY-SPLIT}(D, \mathcal{P}, \mathcal{A}, \mathcal{T})$

IF ($\text{split-pval} < p\text{-cutoff}$)

$\mathcal{A} = \text{DO-SPLIT}(\text{split-info}, \mathcal{A})$

ELSE

$\{\text{merge-pval}, \text{merge-info}\} = \text{TRY-MERGE}(M, \mathcal{A})$

$\{\text{removal-pval}, \text{removal-info}\} = \text{TRY-REMOVE}(M, \mathcal{A})$

IF ($\text{removal-pval} \leq p\text{-cutoff}$ AND $\text{merge-pval} \leq p\text{-cutoff}$)

$\text{stop} = \text{TRUE}$ */* no significant modification found */*

ELSE

IF ($\text{removal-pval} > \text{merge-pval}$)

$\mathcal{A} = \text{DO-REMOVE}(\text{removal-info}, \mathcal{A})$

ELSE

$\mathcal{A} = \text{DO-MERGE}(\text{merge-info}, \mathcal{A})$

RETURN $\mathcal{A}$

END

Fig. 5.
Pseudo-code description of the algorithm RBA.

```

PROCEDURE: TRY-MERGE
INPUT:
   $M$ : The current linear model
   $\mathcal{A}$ : The current list of product aggregates
OUTPUT:
  merge-pval: The largest p-value from a merge
  merge-info: The information about the merge with largest p-value
BEGIN
   $L = \text{LENGTH}(\mathcal{A})$ 
   $\alpha = \text{COEFS}(M)$ 
   $\mathbf{Cov} = \text{COV-MATRIX}(M)$ 
   $\sigma = \text{RESIDUAL-SE}(M)$ 
   $\text{df} = \text{RESIDUAL-DF}(M)$ 
   $\mathbf{se}^2 = \text{DIAGONAL}(\mathbf{Cov})$ 
  merge-pval = 1
  FOR EACH ( $i, j \in \{1, \dots, L\}, i < j$ )
    
$$tval-ij = \frac{\alpha[i] - \alpha[j]}{\sigma \sqrt{\mathbf{se}^2[i] + \mathbf{se}^2[j] - 2\mathbf{Cov}[i, j]}}$$

    
$$pval-ij = 2(1 - F_t(|tval-ij|, \text{df}))$$

    IF ( $pval-ij > \text{merge-pval}$ )
      merge-pval = pval-ij
      merge-agg1 =  $i$ 
      merge-agg2 =  $j$ 
      merge-info = {merge-agg1, merge-agg2}
  RETURN {merge-pval, merge-info}
END

```

Fig. 6.
Pseudo-code description of the procedure TRY-MERGE.

PROCEDURE: TRY-REMOVAL**INPUT:**

M : The current linear model

$\mathcal{A}$: The current set of aggregates

OUTPUT:

removal-pval: The largest p-value

removal-info: The index of the aggregate with the largest p-value

BEGIN

$\mathbf{p} = \text{P-VALUES}(M)$

removal-info = INDEX(MAX($\mathbf{p}$))

removal-pval = $\mathbf{p}[\text{removal-info}]$

RETURN {*removal-pval*, *removal-info*}

END

Fig. 7.

Pseudo-code description of the procedure TRY-REMOVAL.

```

PROCEDURE: TRY-SPLIT
INPUT:
   $D$  : The disease time series
   $\mathcal{P}$  : The original set of product time series
   $\mathcal{A}$  : The current set of product aggregates
   $\mathcal{T}$  : The current set of aggregated time series
OUTPUT:
  split-pval: The p-value of the best split found
  split-info: The information about the best split found
BEGIN
   $L = \text{LENGTH}(\mathcal{A})$ 
  aggregate-min-pv: array (1: $L$ )
  aggregate-min-pv-product: array (1: $L$ )
  FOR  $i=1$  TO  $L$  DO
    aggregate-min-pv[  $i$  ] = 1
    FOR  $i=1$  TO  $L$  DO
      IF (aggregate  $A_i$  has more than one products)
        FOR EACH product  $j$  in  $A_i$  DO
          /* Adjust the set of time series */
           $\mathcal{T}' = \text{ADJUST}(\mathcal{P}, \mathcal{T}, i, j)$ 
           $M' = \text{FIT-LM}(D = \alpha_0 + \alpha_1 T'_1 + \dots + \alpha_{L+1} T'_{L+1} + \epsilon)$ 
           $\alpha = \text{COEFS}(M')$ 
           $\text{Cov} = \text{COV-MATRIX}(M')$ 
           $\sigma = \text{RESIDUAL-SE}(M')$ 
           $\text{df} = \text{RESIDUAL-DF}(M')$ 
           $\text{se}^2 = \text{DIAGONAL}(\text{Cov})$ 
           $tval-ij = \frac{\alpha[i] - \alpha[L+1]}{\sigma \sqrt{\text{se}^2[i] + \text{se}^2[L+1] - 2\text{Cov}[i, L+1]}}$ 
           $pval-ij = 2(1 - F_t(|tval-ij|, \text{df}))$ 
          IF ( $pval-ij < \text{aggregate-min-pv}[i]$ )
            aggregate-min-pv[  $i$  ] =  $pval-ij$ 
            aggregate-min-pv-product[  $i$  ] =  $j$ 
        /* best aggregate to split */
      split-aggregate = INDEX(MIN(aggregate-min-pv))
      /* p-value associated with the splitting of the best aggregate */
      split-pval = aggregate-min-pv[split-aggregate]
      /* the product in the best aggregate that leads to smallest p-value */
      split-product = aggregate-min-pv-product[split-aggregate]
      split-info = {split-aggregate, split-product}
    RETURN {split-pval, split-info}
  END

```

Fig. 8.
Pseudo-code description of the procedure TRY-SPLIT.

PROCEDURE: IRBA

INPUT

 k : Time series length n : Number of OTC healthcare products D : A disease time series $\mathcal{P}$: A set of n OTC time series p -cutoff: The cut-off value for statistical significance max -steps: The maximum number of steps m : The cardinality of a “parent” set $reps$: The number of iterations β : The threshold value for the retention rate

OUTPUT

 Top - β : The set of products with retention rate greater than β

BEGIN

FOR $i = 1$ TO $reps$ DO $S_i^1 = \text{SAMPLE}(\mathcal{P}, m)$ /* i -th “parent” set */ $\mathcal{A} = \text{RBA}(k, n, D, S_i^1, p\text{-cutoff}, max\text{-steps})$ $S_i^2 = \text{GENERATE-CHILD-SET}(\mathcal{A})$ /* i -th child set */ Top - $\beta = \{\}$ $\mathcal{S}_1 = \{S_1^1, S_2^1, \dots, S_{reps}^1\}$ $\mathcal{S}_2 = \{S_1^2, S_2^2, \dots, S_{reps}^2\}$ FOR EACH product i DO $R = r(i, \mathcal{S}_1, \mathcal{S}_2)$ /* retention rate of product P_i */IF ($R \geq \beta$)Add product i to Top - β RETURN Top - β

END

Fig. 9.

Pseudo-code description of the procedure IRBA.

Fig. 10. Plot of BIC versus m . The inset shows the portion of the BIC plot for $m \leq 22$ and highlights the value of m where BIC reaches its minimum.

Fig. 11. Plot of R^2 and normalized $\overline{VIF}$ versus m . The vertical dotted line marks our choice of parameter m .

Table 1

Mean retention rates and 95% confidence intervals for AC Nielsen categories

Category	Mean Retention Rate	Lower 95%	Upper 95%
PRCL	0.714	0.705	0.722
THM	0.654	0.646	0.661
CST	0.468	0.461	0.474
HYDR	0.460	0.450	0.470
PRHE	0.452	0.449	0.455
CRC	0.451	0.444	0.458
CRA	0.446	0.443	0.448
PRAL	0.433	0.418	0.448
ANTA	0.426	0.423	0.430
ACR	0.426	0.420	0.431
PRAR	0.409	0.395	0.422
SINU	0.395	0.387	0.403
CDRO	0.390	0.383	0.397
THRO	0.369	0.362	0.376
DIAR	0.356	0.345	0.366
NASA	0.344	0.337	0.350
BRON	0.341	0.318	0.363
VAPO	0.285	0.266	0.304

Table 2

Product count, correlation with disease time series and mean pair-wise correlation of categories THM, PRCL and sets *Top-0.75*, *Top-0.85*, *Top-0.95*

Category	Product Count	Correlation with Disease Time Series	Mean Pair-wise Correlation
THM	130	0.686	0.042
PRCL	95	0.812	0.074
Top-0.75	950	0.873	0.066
Top-0.85	623	0.902	0.092
Top-0.95	300	0.906	0.148

Table 3

Product count and mean correlation with the disease time series for each AC Nielsen category and three sub-categories of each category: *Top-0.75*, *Top-0.85*, and *Top-0.95*

Category	All products			<i>Top-0.75</i>			<i>Top-0.85</i>			<i>Top-0.95</i>		
	Count	COR		Count	COR		Count	COR		Count	COR	
ACR	287	0.413		49	0.585		26	0.587		7	0.595	
ANTA	607	0.2		129	0.418		76	0.424		28	0.55	
BRON	15	0.229		2	0.312		1	0.262		0	—	
CRA	1104	0.496		234	0.689		160	0.69		74	0.712	
CRC	176	0.503		44	0.74		32	0.726		18	0.761	
CDRO	180	0.457		30	0.711		19	0.587		10	0.564	
CST	230	0.484		39	0.65		24	0.706		10	0.749	
DIAR	66	0.067		10	0.452		4	0.384		1	0.262	
HYDR	81	-0.33		17	0.005		8	-0.058		1	0.306	
THM	130	0.686		62	0.793		45	0.76		26	0.765	
NASA	176	0.342		15	0.535		10	0.477		6	0.482	
PRAL	36	0.164		7	0.448		6	0.417		1	0.251	
PRAR	46	-0.111		5	0.445		2	0.34		2	0.34	
PRCL	95	0.812		55	0.847		47	0.846		37	0.849	
PRHE	937	0.47		215	0.774		143	0.755		73	0.883	
SINU	128	0.249		18	0.412		7	0.33		2	0.372	
THRO	169	0.432		18	0.545		13	0.696		4	0.54	
VAPO	20	0.338		1	0.334		0	—		0	—	

NOTE: A — sign means that the correlation could not be computed due to missing data