Using a Distributed Search Engine to Identify Optimal Product Sets for Use in an Outbreak Detection System

Ruhsary Rexit*†, Fuchiang (Rich) Tsui*, Jeremy Espino*, Sahawut Wesaratchakit*, Ye Ye*, Panos K. Chrysanthis†
*The RODS Laboratory, Department of Biomedical Informatics
†The ADMT Laboratory, Department of Computer Science
University of Pittsburgh, Pittsburgh, Pennsylvania 15260 USA

***Abstract*—This study tests an approach for identifying sets of over-the-counter (OTC) thermometer products whose aggregate sales correlate optimally with aggregate counts of emergency department (ED) visits where patients have symptoms consistent with Constitutional syndrome such as fever and chills. We show that by using a distributed search engine alongside search algorithms (Brute-force), we can quickly identify a minimum set of OTC thermometer products whose sales are optimally correlated to the ED data. We used the Pearson correlation coefficient function to measure the degree of correlation between OTC and ED time series. The optimal OTC product set—comprising 9 thermometer products found by the Brute-force algorithm—has a correlation coefficient value of 0.96. We believe the approach used in this study can be used to efficiently identify different optimal OTC sets for detection of different types of disease outbreaks.**

Index Terms*—*Distributed search, syndromic surveillance, outbreak detection, time series analysis.

I. INTRODUCTION

Syndromic Surveillance is a public health surveillance methodology using individual and population health indicators that are available before a confirmed diagnoses or laboratory confirmation to identify outbreaks or health events and to monitor the health status of a community [1]. Public health surveillance systems have used various data sources as health indicators, including over-the-counter (OTC) medication sales, emergency department (ED) chief complaints, school absenteeism data, and web search queries [2], [3], [4]. Among these, ED data generally serves as the core data source of many Syndromic Surveillance systems such as BioSense [5] and RODS [2]. Researchers have shown that common outbreaks can be detected 1 to 2 weeks earlier with ED data than through conventional disease reporting methods [6].

A common methodology employed in Syndromic Surveillance systems is aggregating health-related temporal events into time series that are analyzed algorithmically for detection of outliers. For example, in an Syndromic Surveillance system using OTC medication sales as an indicator to detect influenza outbreaks, the epidemiologist would analyze a time series of the daily sales of all cough syrup, thermometers, and fever reducers in a specific geographic region. If daily sales of these products exceed some threshold (e.g., 3 times the standard deviation from a baseline value), that could indicate a disease outbreak.

To provide epidemiologists ad-hoc data queries and analyses *on the fly*, a Syndromic Surveillance system (a type of outbreak detection systems) is in need of distributed computing to meet the challenge of timely detection of disease outbreaks. Given the increasing volume of monitored OTC products sales in United States, it becomes a challenge for Syndromic Surveillance systems to meet the near-real time requirements. We previously employed data warehouse approach that allows OLAP queries but the performance is not close to real-time and it requires much larger data storage for storing the large fact table and pre-computed statistics results [7]. One plausible solution is to employ distributed computing, which means that parts of query processing, e.g., filtering and aggregation need to take place over collaborating computers, possibly in the cloud, and as close to the generating sites as possible.

The selection of health indicators (such as specific medications sold) to be used for the detection in Syndromic Surveillance systems also require filtering and data aggregation. Traditionally, this selection process has been performed manually by public health experts but if we want to accelerate this process and make it more accurate, then we need an efficient distributed solution for processing large volume of aggregated data and time series, similar to the support required by a Syndromic Surveillance system for the detection.

Contributions: In this paper, we propose a framework that can work with a minimum set of OTC products whose sales optimally correlate to ED visits and produce heuristic methods that yield close to optimal results in much less time than Brute-force method. The framework utilizes a distributed search engine to efficiently generate time series of time-stamped records (such as unit sales of certain OTC products). We demonstrate this framework using OTC thermometer sales and emergency department visit data.

Specifically, using our framework, we evaluate three different search algorithm to identify a set of thermometer products whose sales over time optimally or close to optimally correlates with ED visits for symptoms (such

COLLABORATECOM 2012, October 14-17, Pittsburgh, United States

DOI 10.4108/icst.collaboratecom.2012.250728

as fever) consistent with Constitutional syndrome. The three search algorithms are *brute-force*, *greedy*, and a *dynamic programming* (Knapsack solution). To measure the degree of correlation, we first obtained time series for the unit sales of a set of thermometer products and ED visit data. Then, we measured the correlation coefficient value between these two time series using the Pearson correlation coefficient function [8], [9]. Using a limited data set, our results show that the Knapsack search exhibits the worst performance whereas the greedy search is competitive to the brute-force search, which produces the optimal OTC products.

Roadmap: Section II introduces our system environment, including experimental datasets, and search/query techniques we applied in the frame work, data filtering and our evaluation process. Section III presents the experimental results for two different data filtering methods and a comparison results of all three search algorithms. Section IV includes our analysis of the experimental results and Section V concludes with future work.

II. METHODS

In this section, we present our experimental dataset, system setup, search algorithms, optimization function and evaluation, as well as the filtering processes to reduce the impact of noisy data.

A. Experimental Datasets

1) OTC Medication Sales: We obtained thermometer sales data from the National Retail Data Monitor (NRDM). NRDM is a public health surveillance tool that collects and analyzes daily over-the-counter point of sale data to rapidly identify disease outbreaks. NRDM was built by the RODS Laboratory at the University of Pittsburgh in collaboration with the food and drug retail industry, as well as state and local health departments [10].

NRDM collects daily sales data from over 33,304 (30,820 active) stores from 15 (12 active) different retailers across the United States and has been operational since 2003. NRDM has a transactional database of 1.23 billion records for over 9,000 medications over a period of 9+ years. Each record contains the product ID, Universal Product Code (UPC), date of sale, total unit sales, promoted unit sales and store zipcode. We obtained transactional sales data for Pennsylvania from 2009 to 2011 to load into our search engine.

2) ED Visit Data: We retrieved time series of daily ED visits for Constitutional chief complaints for Allegheny County for the year 2009 from the Pennsylvania RODS system. The PA RODS System is a public health surveillance system for the state of Pennsylvania that collects de-identified ED visit data from 166 (111 active) hospitals since 1999. Emergency department visits and daily aggregated number of different syndrome categories were obtained from emergency departments. Hospitals send patient visit data including registered chief complaint to RODS Laboratory from clinical encounters over virtual private networks and leased lines using the Health Level 7 (HL7) message protocol. The data are sent in real time. CoCo (Complaint Coder) automatically classifies the registration chief complaint from the visit into one of seven syndrome categories (Respiratory, Botulinic, Gastrointestinal, Neurologic, Rash, Constitutional, Hemorrhagic) using Bayesian classifiers [2], [11].

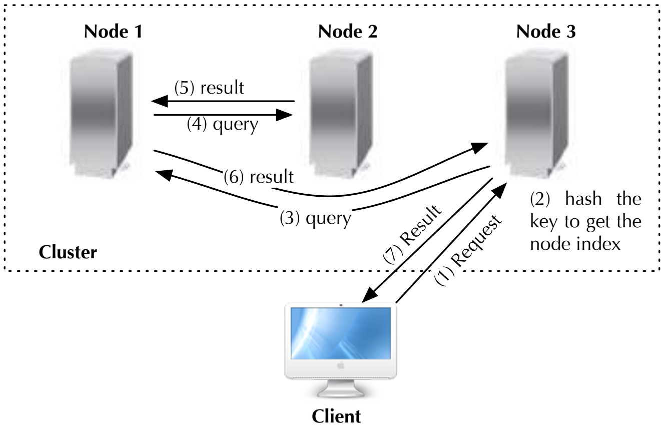


Fig. 1. Distributed computing Elastic Search scheme

The NRDM offers 23 OTC categories. We chose the thermometer sales category as our indicator of the influenza outbreak mainly because researchers found a strong correlation (correlation coefficient is 0.91) between patients with Constitutional syndrome visiting emergency departments (EDs) and OTC thermometer sales in Pennsylvania in past influenza seasons [12]. Villamarin et al. also demonstrated high correlation (0.89) between actual and predicted ED visits using thermometer sales data [13]. For our preliminary experiment, we selected daily aggregated ED visits for the Constitutional category because it generalizes complaints such as fever, chills, or malaise.

B. System Setup

We constructed a distributed search engine for the OTC data using Elastic Search. Elastic Search is a distributed search engine built on top of a text search engine called Lucene developed by Shay Banon [14], [15].

A distributed search engine is a system wherein data records are stored over a network of computers (or nodes) which act collaboratively to answer queries as well as to balance the workload among them automatically and transparently. These data records are indexed locally within each node, which means there is no global catalog (hash table) of data distribution but each node has partial catalog. Thus, data retrieval topologically is not a star but rather a star-chain on shown in Figure 1.

The Figure 1 is a simple scheme of distributed search engine with three nodes. When a query is issued to the network (distributed system), the query is directed to the most lightly loaded node, Node 3 (Step 1 in Figure 1). Based on its local catalog, Node 3 identifies which records stored locally meet the query parameters and which nodes store records that might meet the query parameters (Step 2). Then, the query is forwarded to all the identified nodes with records which might be part of the query result, Node 1 in our example (Step 3). When Node 1 receives the request from Node 3, it carries out the same steps as Node 3, identifying and forwarding the query to Node 2 (Step 4). When a node, such as Node 2, receives a query and is not aware of any other node in the chain to further forward the query, it returns the records part of the query result which are stored locally to Node 1 (Step 5). In turn, Node 1 appends its records which are part of the query result to the ones received from Node 2 and sends them to Node 3 (Step 6), which in turn, sends them to the Client (Step 7).

We loaded three years of transactional OTC data for Pennsylvania (from January 1, 2009 to December 31, 2011) on a three node search cluster. Each node was allocated 20 GB of RAM and two CPUs. The data was distributed over 5 shards with replicas (10 total shards) and comprised 18.5 million records. The data occupied 6.2 GB.

We constructed an API using Java to facilitate the creation of NoSQL queries for the search engine. This API returned time series for a set of product IDs aggregated over days, weeks, months, years or any time period.

C. Algorithms

We implemented the three search algorithms—Brute-force, greedy, and dynamic programming (Knapsack algorithm)—using the Java programming language.

1) Brute-force Search Algorithm: Using Brute-force search or exhaustive search, we generated every possible combination of thermometer products that was queried from our (distributed) search engine and computed the correlation coefficient value of each set. We retained the set with the highest correlation value. The advantage of this approach is that it is optimal, i.e., it searches the entire space of available thermometer product sets. The disadvantage is that its time complexity is proportional to the number of candidate solutions. Specifically, the time complexity of brute-force search is $O(2^N)$ [16] where N is the number of thermometer products.

2) Greedy Search Algorithm: We designed a search algorithm by using a greedy strategy that started the computation from an initial set of all thermometer products, and iteratively the product from the set that, when removed, would improve the correlation value the most. This algorithm is not guaranteed to find an optimal subset of products, but the output result in this case was close to the optimal result. Also, its time complexity was less than that of the brute-force search. The time complexity of this type of greedy search is $O(N^2)$.

3) Knapsack Algorithm: The Knapsack algorithm is a dynamic programming method for solving optimization problems. For this approach, we computed the solutions to the subproblems once and stored the solutions in a table so that they could be reused later. This algorithm, like the greedy search, shows a reduced time complexity of $O(N^2)$ [16]. It reduced time complexity at the expense of memory. The Knapsack solution selects one OTC product at each step, and adds it to the knapsack. If adding this OTC product to the subset of OTCs in the knapsack increases the correlation value of the subset of OTC products with ED visits, it remains in the Knapsack. Otherwise, the OTC product is discarded (is not kept in the knapsack) and move to the next OTC product in the list. Once an OTC product is eliminated from the knapsack, it does not have another opportunity to cluster with the subset in the Knapsack. By performing in this way, the knapsack solution eliminates some of the possible combinations that may actually include the optimal subset. Thus, the order of the input set determines the output set. Although it is known that it is an expensive operation (i.e., almost behaves like Brute-force) to find out which order of the input set gives an optimal set, we decided to use this algorithm as a kind of low boundary in our evaluation.

D. Optimization Function

We used the Pearson correlation coefficient function [17] to compute correlation values between the two time series: a set of OTC product weekly sales and weekly ED visits. The Pearson correlation coefficient equation is written as Equation 1:

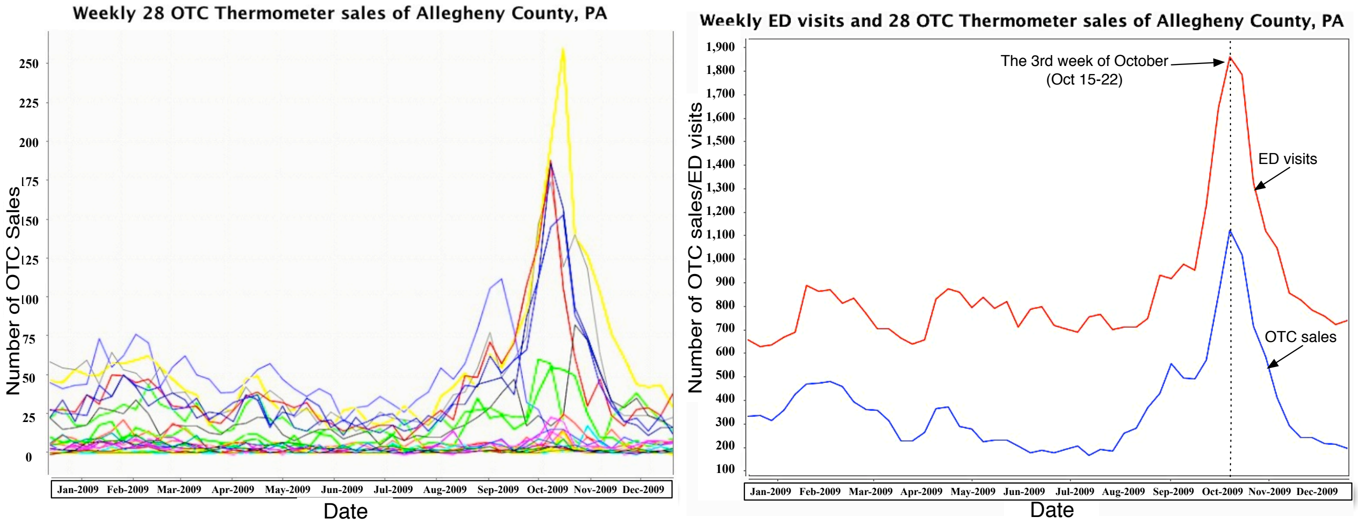
$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

In Equation 1, r is a measure of the correlation coefficient (linear dependence) between two variables (time series) X and Y written as x_i and y_i (where i = 1, 2, ..., n), n is the sample data size, and $\bar{x}$ and $\bar{y}$ are the mean values of two samples from the data.

E. Evaluation

We evaluated the three search algorithms by comparing the correlation coefficient values (CCVs) computed by the Pearson correlation coefficient function and run time. In our evaluation, we generated time series of thermometer sales for Allegheny County, Pennsylvania for the year 2009. To reduce the impact of noisy data, we applied two different filtering processes in our search queries.

1) Product Level Filtering: The product level filtering excluded thermometer products that had less than a specific number of days of sales over the study period. For example, filtering at the threshold 10 days, we generated a dataset (Dataset 1) that included 28 OTC thermometer products. By varying the threshold from 10 to 70 days, we generated additional datasets with 26 OTC thermometer products (Dataset 2 with threshold 20 days), 25 OTC thermometer products (Dataset 3 with threshold 30 days), 23 OTC thermometer products (Dataset 4 with threshold 50 days), and 20 OTC thermometer products (Dataset 5 with threshold 70 days). The filtering at threshold 40 and 60 days generated same datasets with Dataset 3 and Dataset 4 respectively. The results obtained those different datasets will be further discussed in Section III-A.



(a)

(b)

Fig. 2. Weekly OTC Thermometer sales and ED visits for Constitutional syndrome of Allegheny county, PA in 2009

2) Store-Product Filtering: The store-product filtering not only excluded sales data from any store that had less than a specific number of days of sales over the study period as a threshold, but also excluded the products stated in the product filtering process. For example, by setting each store sales threshold at 60 days (found 118 stores out of 208), and single product sales threshold at 10 days, we generated a dataset with 26 OTC thermometer products. By varying the store sales threshold from 60 to 100 and product sales threshold from 10 to 50, we generated additional datasets with 25 OTC thermometer products (Dataset 2 with store threshold 70 and product threshold 20 days, 96 stores found out of 208), 22 OTC thermometer products (Dataset 3 with store threshold 80 and product threshold 30 days, 86 stores found out of 208), 21 OTC thermometer products (Dataset 4 with store threshold 90 and product threshold 40 days, 75 stores found out of 208), and 17 OTC thermometer products (Dataset 5 with store threshold 100 and product threshold 50 days, 49 stores found out of 208). The result obtained with those different datasets will be further discussed in Section III-B.

III. Experiment Results

We conducted our evaluations on an Apple iMac computer (3.06 GHz Intel dual cores CPU, 4 GB RAM). The iMac computer served as a client computer that queried the distributed system described in Section II-B. The time cost of each search algorithm was measured by excluding the querying and filtering process. In this section, we report results for each of the two filtering processes: (1) product-level filtering, and (2) store-product filtering.

A. Product Level Filtering

The parameters we used for querying a total 596 of OTC thermometers in the distributed system include:

- Query period: from Jan 1, 2009 to Dec 31, 2009

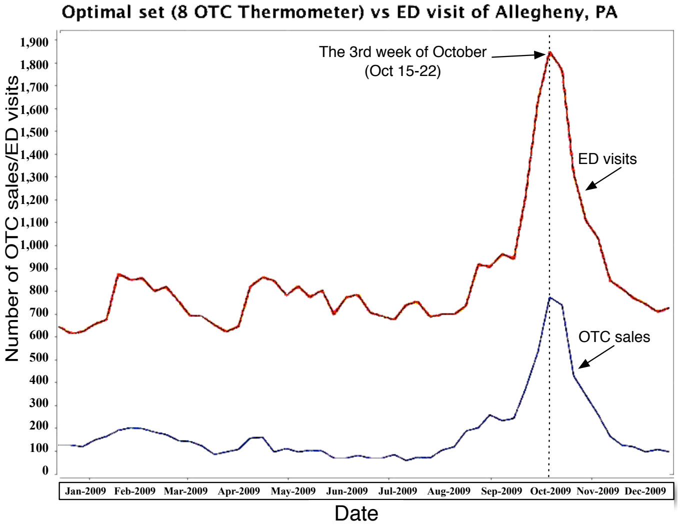


Fig. 3. Weekly time series of an optimal set of 9 OTC thermometer sales and ED visits for constitutional syndrome of Allegheny County, PA in 2009. The optimal OTC product set was obtained from product level filtering and the Brute-force search.

- Geographic area: Allegheny County, PA
- Aggregate time level: Daily
- Breakdown by: product ID, i.e., Universal Product Code (UPC)

After applying the product-level filtering, we identified a set of 28 (from 596) OTC thermometer products. We further aggregated daily data to weekly data as shown in Figure 2. We then evaluated the 3 search algorithms in terms of optimal output results and how they performed based on the results of the Pearson correlation coefficient function. The correlation value of aggregated time series of 28-product set with ED visits for Constitutional syndrome was 0.91, and the both time

TABLE I
THE LIST OF OTC PRODUCTS IN OPTIMAL SET OF 9 (OUT OF 28) AND THEIR INDIVIDUAL CCVS

Individual OTC Product	Correlation coefficient value
Product 4	0.9414
Product 9	0.0967
Poduct 10	0.8715
Product 11	0.7744
Product 14	0.1244
Product 16	0.5624
Product 18	0.9381
Product 19	0.8854
Product 24	0.9140

TABLE II
COMPARISON OF THE THREE ALGORITHMS WITH DIFFERENT OTC PRODUCT SETS BASED ON THE PRODUCT LEVEL FILTERING (CCV⋆: PEARSON CORRELATION COEFFICIENT VALUE; SIZE†: SIZE OF THE OPTIMAL OUTPUT SET WITH A CORRESPONDING SEARCH ALGORITHM)

Input Dataset		Brute-force search output		Greedy search output		Knapsack search output	
	Number of OTCs	CCV⋆	Size†	CCV⋆	Size†	CCV⋆	Size†
Dataset1	28	0.9592	9	0.9592	9	0.9586	11
Dataset2	26	0.9592	9	0.9583	9	0.9592	10
Dataset3	25	0.9589	7	0.9589	7	0.9285	15
Dataset4	23	0.9589	7	0.9589	7	0.9567	11
Dataset5	20	0.9586	6	0.9586	6	0.9565	9

series peaked the 3rd week of October (Oct 15-22), 2009, as shown in Figure 2b. The red (top) line in Figure 2b is the ED visit time series, while the blue (bottom) line is 28 OTC thermometer sales aggregated time series.

1) Brute-force Search Results: The Brute-force search algorithm identified an *optimal* set of 9 (out of 28) OTC products with the best correlation value at 0.9592. Table I shows the individual correlation coefficient values (CCVs) for each of the 9 OTC products with ED visits for constitutional syndrome. The individual product CCVs ranged from 0.0967 to 0.9414. Figure 3 shows the aggregated time series of the optimal set of OTC (9) thermometer products obtained from product filtering and ED visits for Constitutional syndrome. Both time series peaked the 3rd week of October (Oct 15-22), 2009.

2) Comparison between three Search Algorithms: Table II shows the results for the three search algorithms using different OTC product sets with aggregated sales of 28, 26, 25, 23, and 20 products. The relationship of those 5 input datasets in Table II is a smaller dataset is the subset of a larger dataset, and they were generated using the product-level filtering method introduced in Section II-E1. The smaller output set may not necessarily be the subset of larger output set.

In Table II, the 1st column represents the input datasets and the 2nd column has the number of OTC products (size) in each dataset. The 3rd, 5th, and 7th columns have the correlation coefficient value of output product set computed respectively by the three different algorithms, while the 4th, 6th, and 8th columns show the size of those different output sets.

TABLE III
THE LIST OF OTC PRODUCTS IN OPTIMAL SET OF 8 (OUT OF 26) AND THEIR INDIVIDUAL CCVS

Individual OTC Product	Correlation coefficient value
OTC 4	0.9425
OTC 9	0.1316
OTC 10	0.8723
OTC 11	0.7921
OTC 12	0.8673
OTC 14	0.1242
OTC 16	0.3949
OTC 24	0.9120

B. Store-Product Filtering

The store-product filtering applied store-level filtering in addition to the product-level filtering. During the experiment, we found 118 stores out of 208 qualified for store-level filtering criteria (stores with sales data $>=$ 60 days). The total number of thermometer included in this study as a result of the filtering process was 26 out of 596 products. Figure 4a shows the time series of each of the the individual 26 OTC products, while Figure 4b has a time series of ED visit for constitutional syndrome and a aggregated time series of 26 OTC thermometer sales.

1) Brute-force Approach Results: The Brute-force search algorithm identified an optimal set of 8 (out of 26) OTC products that had a best correlation value of 0.9612. Table III shows individual correlation values (CCVs) for each of the 8 products with ED visits for Constitutional syndrome. The individual product CCVs ranged from 0.1316 to 0.9425. Figure 5 shows the aggregated time series of the optimal set of OTC (9) thermometer products obtained from store-product filtering and ED visits for Constitutional syndrome. Both time series peaked the 3rd week of October (Oct 15-22), 2009.

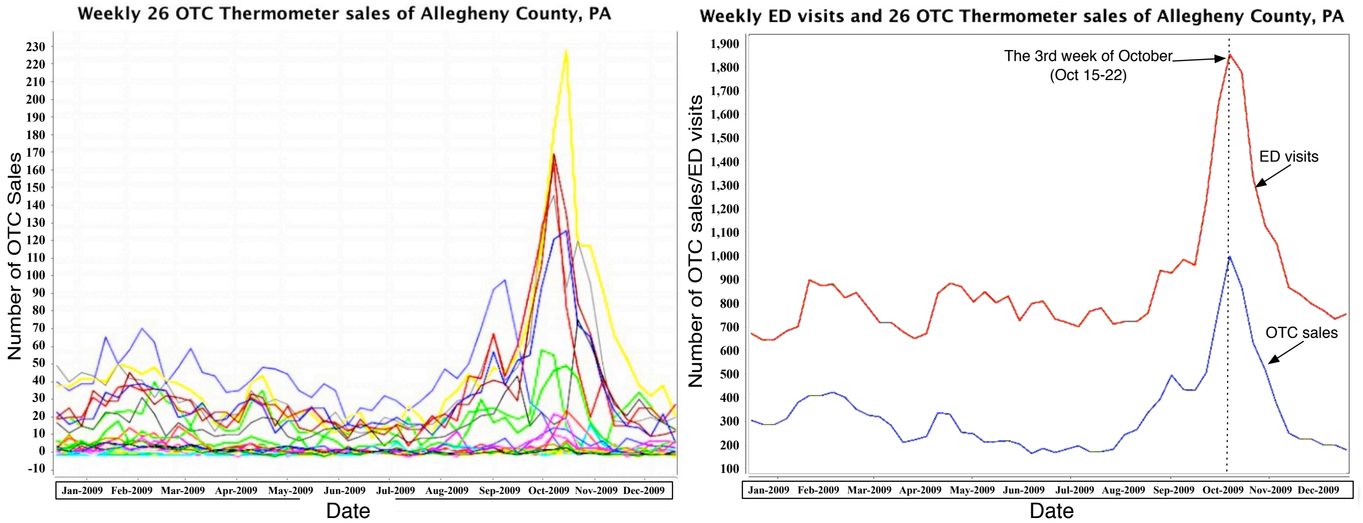
2) Comparison among the three Search Algorithms: Table IV shows the results for the three search algorithms using different OTC product sets with aggregated sales of 26, 25, 22, 21 and 17 products, respectively. The relationship of those 5 input datasets in TableIV is a smaller dataset is the subset of a larger dataset, and they were generated using the store-product filtering method introduced in Section II-E2. The smaller output set may not necessarily be the subset of larger output set.

In Table IV, the 1st column represents the input datasets and the 2nd column lists the number of OTC products (size) in each dataset. The 3rd, 5th, and 7th columns have the correlation coefficient values of optimal product set computed respectively from the three different algorithms, while the 4th, 6th, and 8th columns list the size of those different output sets.

Table V shows the run-time comparison between 3 algorithms with Dataset 1 that has 28 OTC products shown in Table II.

IV. DISCUSSION

By limiting our study dataset to (1) the geographic area of Allegheny County in PA; (2) a subset of OTC products (596 out of 9,000+); (3) a specific ED visit for Constitutional



(a)

(b)

Fig. 4. Weekly 26 OTC thermometer sales and ED visits Constitutional syndrome of Allegheny county, PA in 2009

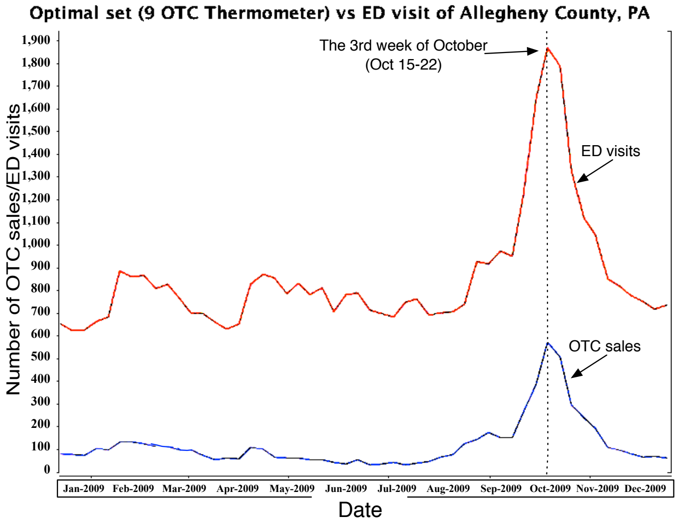


Fig. 5. Weekly time series of an optimal set of 8 OTC Thermometer products and ED visits for Constitutional syndrome of Allegheny County, PA in 2009. The optimal OTC product set was obtained from store-product filtering and the Brute-force search.

TABLE IV
COMPARISON OF THE THREE ALGORITHMS WITH DIFFERENT OTC PRODUCT SETS BASED ON THE STORE-PRODUCT FILTERING (CCV⋆: PEARSON CORRELATION COEFFICIENT VALUE ; SIZE†: SIZE OF OPTIMAL OUTPUT SET WITH THE CORRESPONDING SEARCH ALGORITHM)

Input Dataset		Brute-force search output		Greedy search output		Knapsack search output	
	Number of OTC	CCV⋆	Size†	CCV⋆	Size†	CCV⋆	Size†
Dataset1	26	0.9612	8	0.9612	8	0.9522	12
Dataset2	25	0.9595	10	0.9595	10	0.9581	11
Dataset3	22	0.9569	9	0.9569	9	0.9569	9
Dataset4	21	0.9561	7	0.9561	7	0.9501	7
Dataset5	17	0.9480	8	0.9480	8	0.9433	7

TABLE V
RUN TIME COMPARISON AMONG THE THREE SEARCH ALGORITHMS WITH SET OF 28 OTC PRODUCTS

	Brute-force Search (seconds)	Greedy Search (seconds)	Knapsack Approach (seconds)
CPU Time	150	< 1	< 1

syndrome category, we provided a framework for an effective search of *optimal* OTC product sets that correlate with ED visit data, which could facilitate disease outbreak detection.

The Brute-force search serves as a gold standard for optimal OTC product sets that correlate with ED visits. Since this approach is computationally intensive, that makes the scalability is limited. Thus, we need a better approach which is computationally less expensive and can so overcome the scalability issue.

The greedy search method provided fairly good results, and was both efficient and easy to implement. We also assume we are able to scale it to large dataset. The results for greedy search, such as those shown in Table II and Table IV were the same as those for the brute-force search result.

The Knapsack solution was as efficient as the greedy search in terms of run time, but more complex to implement. It also returned fairly good results, however, because of the data property of time series the output result is not reliable. The Knapsack solution guarantees to find the optimal solution if adding an item into the knapsack either increases or decreases value not depending on the subset in the knapsack. If we add one OTC each time and correlate with ED visits, we may get an increased CCV for a subset in the knapsack but may get a decreased one for another subset. With this behavior, the Knapsack solution will not necessarily yield an optimal solution.

Although the greedy search and Knapsack search algorithms may not be able to identify such a compact set of OTC products that has optimal CCV as shown in the result, the result of greedy search is more close to the Brute-force search result Setction III-A and Section III-B, and indicates it is more reliable than the Knapsack search algorithm for solving our specific problem. In terms of time complexity, these two algorithms required much less run-time ($<$ 1 second) than the brute-force search (150 seconds) as shown in Table V. Run-time is computed on the client by excluding the data retrieval process.

The results demonstrate the need for the search for an optimal product set. The CCV from the initial OTC set comprising 28 products through the product-level filtering was 0.91 whereas the CCV from the compact set that comprised 9 products obtained from the Brute-force search was 0.9592. We found that store-product level filtering increased the CCV over the product level filtering. The difference between the CCVs of optimal product sets from product level filtering and store-product filtering was 0.002. Although adding more filtering criteria had a small amount of improvement for our current data set, it still be necessary for scaled dataset.

V. Conclusions and Future Work

Syndromic Surveillance system is an outbreak detection system that uses various individual and population health related temporal data to identify disease outbreaks or health events, and to monitor the health status of a community. Selection of large volume of health care indicators traditionally been done manually, and the performance of such Syndromic Surveillance system is not efficient enough to detect the outbreak real-time. In order to provide epidemiologies efficient data selection and enable them to perform data processing and computational analysis on a distributed system is the main purpose of this work.

Identifying an optimal set of OTC products that correlates with ED visits facilitates disease outbreak detection. With the leverage of an in-memory search across multiple machines (or nodes), Elastic Search allows fast and ad-hoc queries from a large data set, which is an ideal setup for a surveillance system and machine learning algorithms. Our large NRDM dataset comprising 1.2 billions records would serve as a good resource for machine learning experiments using Elastic Search.

Our results demonstrate that using a distributed search engine, a search algorithm and an optimization function could facilitate the identification of optimal sets of health-related events from large sets of collaboratively obtained public health data. The proposed approach filters the noisy OTC products and identifies optimal product sets to facilitate more timely and accurate detection of disease outbreaks.

For future work, we would like to improve our search algorithms that can output optimal results even with scaled dataset, in a reasonable amount of time. Also, we would like to apply larger, such as statewide or nationwide, datasets with different OTC products and ED syndrome categories using the proposed framework.

Acknowledgment

This research was funded by grant SAP #40000012020 from the Pennsylvania Department of Health and grant from the National Retail Data Monitor.

We thank Mehmud Abliz for his significant contribution on designing the greedy search algorithm in our framework. Also, we thank Leon Lai for his hard work on maintaing the the UPC update procedure where our quarterly renewed OTC product list comes from.

References

[1] Syndromic Surveillance (SS), Available: http://www.cdc.gov/ehrmeaningfuluse/syndromic.html, Accessed Aug 14, 2012

[2] F. Tsui, J. U. Espino, Technical Description of RODS: A Real-time Public Health Surveillance System, JAMIA, Sept. 2003;10(5) 399-408

[3] W. R. Hogan, Early Detection of Pediatric Respiratory and Diarrheal Outbreaks from Retail Sales of Electrolyte Products, JAMIA, 2003,10(6)

[4] R. Li, G. L. Wallstrom, W. R. Hogan, A Multivariate Procedure for Identifying Correlations between Diagnoses and Over-the-counter Products from Historical Datasets, AMIA 2005 Symposium Proceedings

[5] BioSense fact sheet, Availlable: http://www.cdc.gov/biosense, Accessed Sep 22

[6] L. Lazarus R, Kleinman KP, Dashevsky I, et al. Using automated medical records for rapid identification of illness syndromes (syndromic surveillance): the example of lower respiratory infection. BMC Public Health. 2001;1:9

[7] F. C. Tsui, J. Espino, Key design elements of a data utility for national biosurveillance: event-driven architecture, caching, and Web service model. Annual Symposium Proceedings/AMIA Symposium 2005, 739-43

[8] J. Que, F. Tsui, Rank-based spatial clustering: an algorithm for rapid outbreak detection, JAMIA, Feb. 2011;18 (218-224)

[9] A.Dugas, Y. Hsieh, Google Flu Trends: Correlation With Emergency Department Influenza Rates and Crowding Metrics, CID, January 8, 2012

[10] NRDM National Retail Data Monitor a Public Health Surveillance Tool, (RODS) Laboratory, http://rods.health.pitt.edu/NRDM.htm

[11] W. Chapman, J. Dowling, M. Wagner, Classification of Emergency Department Chief Complaints Into 7 Syndromes: A Retrospective Analysis of 527,228 Patients, Annals of Emergency Medicine, November 2005, Volume 46

[12] J. Que, F. Tsui, Spatial and Temporal Algorithm Evaluation for Detecting Over-The-Counter Thermometer Sale Increases during 2009 H1N1 Pandemic, Journal of Public Health Informatics, Vol.4, No. 1, 2012

[13] R. Villamarin, G. Cooper , F. Tsui, et al. Estimating the incidence of influenza cases that present to emergency departments. Emerging Health Threats Journal. 2011; 4: s57

[14] Why ElasticSearch?, Available : http://www.elasticsearch.com, Accessed Aug 10,2012

[15] ElasticSearch A Distributed RESTful Search Engine, Available : https://github.com/elasticsearch/elasticsearch, Accessed Jul 20, 2012

[16] T. H. Cormen, Introduction to Algorithms Third Edition, 2009

[17] B. Rosner, Fundamentals of Biostatistics 7th edition, 2010

[18] Centers for Disease Control and Prevention. Overview of influenza surveillance in the United States. Available: http://www.cdc.gov/flu/weekly/overview.htm, Accessed Aug 1, 2012

[19] G. L. Wallstrom, W. R. Hogan, Unsupervised clustering of over-the-counter healthcare products into product categories, Journal of Biomedical Informatics 40 (2007) 642648