Knowledge-Based Syndromic Surveillance for Bioterrorism

Mark A. Musen[1], Monica Crubézy[1], Martin O'Connor[1], and David Buckeridge[1,2]

[1]Stanford Medical Informatics, Stanford University Stanford, CA 94305-5479
[2]VA Palo Alto Health Care System, Palo Alto, CA

Abstract

Syndromic surveillance requires the acquisition and analysis of data that may be "suggestive" of early epidemics in a community, long before there is any categorical evidence of unusual infection. These data are often heterogenous and often quite noisey. The processs of syndromic surveillance poses problems in data integration; in selection of appropriate reusable problem-solving methods, based on task features and on the nature of the data at hand; and in mapping integrated data to appropriate problem solvers. These are all tasks that have been studied carefully in the knowledge-based systems community for many years. We demonstrate how a software architecture that suppports knoweldge-based data integrationa and problem solving facilitates many aspects of the syndromic-surveillance task. In particular, we use reference ontologies for purposes of semantic integration and a parallelizable blackboard architecture for invocation of appropriate problem solving methods and for control of reasoning. We demonstrate our results in the context of a prototype system known as the Biological Spacio-Temporal Outbreak Reasoning Module (BioSTORM), which offers an end-to-end solution to the problem of syndromic surveillance.

The New Trend: Syndromic Surveillance

In recent years, public-health surveillance has become a priority for national security and public health, driven by fears of possible bioterrorist attacks. Authorities argue that early detection of nascent outbreaks through surveillance of "pre-diagnostic" data is crucial to prevent massive illness and death (Pavlin 1999). The need for improved surveillance and the increasing availability of electronic data have resulted in a blossoming of surveillance-system development (Bravata et al. 2004). Most recently developed systems use electronically available data and statistical analytic methods in an attempt to detect disease outbreaks rapidly. In general, the emphasis is on the interpretation of noisy, non-definitive data sources, such as diagnosis codes from emergency-room visits, reports of over-the-counter and prescription drug sales, reports of absenteeism, calls to medical advice personnel, and so on.

For example, the Real-time Outbreak Detection System (RODS; Tsui et al. 2003) allows for automated transmission and analysis of administrative diagnostic codes and other data directly from hospital information systems at many emergency rooms in the greater Pittsburgh area. The Electronic Surveillance System for the Early Notification of Community-based Epidemics (ESSENCE; Lombardo et al. 2003) monitors disease codes assigned for outpatient visits by military personnel and their dependants across the United States and throughout the world. More recently, the CDC began development of the BioSense system to monitor data from many sources including DOD and VA facilities, laboratory systems, and over-the-counter pharmaceutical sales. By the summer of 2003, public health authorities had already deployed more than 100 different surveillance systems in the United States, all relying on electronically available data to detect disease outbreaks rapidly (Buehler et al. 2003).

In most situations, surveillance data that are available electronically are not collected for the expressed purpose of monitoring the public's health. Recently deployed surveillance systems tend to rely on data collected for administrative and business purposes. For example, many systems follow healthcare utilization records collected to enable billing, or pharmaceutical sales records collected for inventory and marketing purposes. Because these data sources are not collected with surveillance in mind, they often are biased in various ways. In addition, because public health agencies do not control the data collection, the data rarely conform to a standard format. Different data sources can represent the same concepts differently, and different data sources can also represent different concepts in a superficially similar manner. When incorporating data sources into a surveillance system, the differences in structure and concept representation must be reconciled. Semantic reconciliation is especially important so that analyses across data sources can integrate conceptually diverse data and can reason about those data in a consistent manner.

Knowledge-Based Syndromic Surveillance

To meet the complex operational and research needs of surveillance applications, we have developed a prototype system known as the Biological Spatio-Temporal Outbreak Reasoning Module (BioSTORM; Buckeridge et al. 2003).

BioSTORM is a computational framework that brings together a variety of data sources and analytic problem solvers with the goal of meeting the performance demands of emerging disease-surveillance systems. The system addresses the following goals: (1) to acquire and curate data from diagnostic and pre-diagnostic sources; (2) to provide a knowledge-based infrastructure to integrate and experiment with alternative data sources and problem solvers; and (3) to support development and evaluation of problem solvers for temporal and spatial analysis.

As shown in Figure 1, the BioSTORM system has four main components, each described in the remainder of this section: (1) a data-source ontology for describing the features of specific data sources and data streams to be used for analysis; (2) a library of statistical and knowledge-based problem solvers for analyzing biosurveillance data; (3) an intelligent mediation component that includes (a) a data broker to integrate multiple, related data sources that have been described in the data-sources ontology and (b) a mapping interpreter to connect the integrated data from the data broker to the problem solvers that can best analyze those data; and (4) a control structure, known as RASTA, that deploys the problem solvers on incoming streams of data.

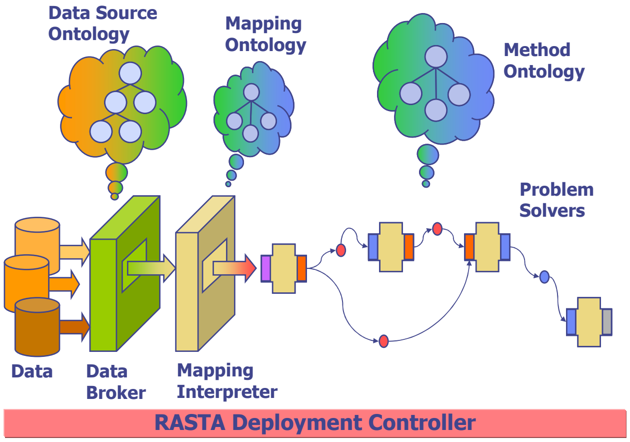

Figure 1. Overview of deployed BioSTORM components showing data being fed through the Data Broker and Mapping Interpreter to a set of problem-solving methods. The RASTA deployment controller orchestrates the deployment of problem-solving methods (PSMs) and the flow of data to those PSMs via the Data Broker and the Mapping Interpreter. The Data Source and Mapping Ontologies are used by the broker and mapping interpreter to construct semantically uniform streams of data for the deployed PSMs. The Method Ontology is used by RASTA to configure sets of PSMs into analytic strategies to perform analysis on those data streams.

A Data-Source Ontology for Describing and Contextualizing Data Streams

Public-health surveillance data are diverse and usually distributed in various databases and files with little common semantic or syntactic structure. Thus, these data can be difficult to represent in a way that enables their consistent analysis by reusable analytic methods. We have developed a data-sources ontology that provides a means for describing extremely diverse data in a coherent manner and that facilitates reasoning and processing of those data (Pincus and Musen 2003). Our ontology provides a domain-independent semantic structure for raw data to assist in the integration of data from disparate sources. More precisely, the data-sources ontology provides an approach to data integration that combines the semantic rigor of creating a global ontology with the flexibility and level of detail that comes from devising customized, local ontologies specifically for each data source of interest.

The data-sources ontology aims to make data self-descriptive by associating a structured context with each potential data source. A developer describes the context of data from a particular data source by filling in a template with relevant details about that data source. The data-sources ontology provides a predefined taxonomy of data attributes to describe this context. To describe individual data elements, developers use terms that we have adopted from the Logical Identifier Names and Codes (LOINC). The LOINC approach, which clinical pathologists use to contextualize results reported by clinical laboratories, describes a piece of data along five major semantic axes. We have generalized the LOINC axes from their specific role in reporting clinical laboratory results to a generic set of descriptors for many different types of data.

This systematic, template-directed process allows developers to create a customized local model of each data source that shares a common and consistent structure, space of attributes, and set of possible attribute values with all other similarly created models. The BioSTORM system processes the data-sources ontology to access relevant context information about incoming data streams to interpret and analyze those data appropriately.

We have used the data-sources ontology successfully to develop descriptions for a number of data sources, including San Francisco Emergency 911 dispatch data and patient data from the Palo Alto VA medical center, as well as data related to reportable diseases, in collaboration with the CDC. The data-sources ontology was able to capture individual-level primitive data (e.g., signs and symptoms, laboratory tests) as well as observable population-level data (e.g., aggregated syndrome counts, school absenteeism). The ontology offers descriptions of generic data sources (e.g., 911 dispatch data), with instances of the generic descriptions that describe the specific data fields of particular data sources (e.g., the 911 dispatch data available in San Francisco). The Data Broker component (see Figure 1) is a software module that uses the data-sources ontology (1) to perform queries on the BioSTORM data feeds (which may be tab-delimited files, XML documents, or relational databases with sundry formats) and (2) to transform the queried data into an internal canonical representation that ultimately may be mapped to the problem-solving methods that will perform the necessary data analysis.

A Library of Outbreak-Detection Problem-Solving Methods

Next-generation surveillance systems will require a variety of analytic methods, ranging from traditional statistical techniques that operate on low-level data such as raw disease counts to knowledge-based approaches capable of reasoning about qualitative data and unusual patterns. In addition, there is a need for methods capable of making correlations among different kinds of data, and of aggregating and abstracting data into information about populations, spatial regions, or temporal intervals. In addition to implementing a set of methods that can address this range of capabilities, there is a need to model explicitly the performance characteristics, data requirements, and assumptions of each method to facilitate system modification, both to enhance portability and reuse of methods, and to allow users to understand system functions and results. We have already developed a small library of computational methods to address the analysis of multiple, varying types of data for detecting abnormal patterns (Buckeridge et al. 2003).

All surveillance analysis methods that we have defined are associated with a method ontology that defines the classes of data and knowledge on which a given method operates (Gennari et al. 1994). The method ontology thus makes explicit the data requirements of a problem-solving method (like the signature of a subroutine in a programming language), and thereby enables data sources to be mapped to analytic methods using the BioSTORM Data Broker and Mapping Interpreter. The method ontology facilitates the interoperation of analytic methods by identifying appropriate interactions between methods and different types of data (e.g., so that a univariate, atemporal method is not used to interpret multivariate temporal data). Overall, our framework for surveillance algorithms provides a structure for incorporating existing surveillance algorithms into our system and for establishing the knowledge requirements of those methods. By making the characteristics of each method explicit, our framework facilitates the identification of methods suitable for a specific subtask in the overall task decomposition.

A Mediation Component for Integrating Data and Problem Solvers

Our data-sources ontology provides a consistent mechanism for describing sources and elements of data in a way that allows these data be used concurrently by surveillance methods. However, surveillance methods that operate on these data may have many different input requirements. Some algorithms operate at the population level, whereas others work at the level of individuals; many algorithms expect time series data at varying granularities; certain algorithms require spatial data at several levels of aggregation. An infrastructure for outbreak surveillance therefore needs to integrate data in different combinations and at varying levels of abstraction, depending on the type of analysis to be performed.

Because each problem solver in our library has a declarative method ontology, our system makes explicit the type of data that each method is capable of processing. At the same time, we have devised a standard mechanism to mediate data from multiple sources to various methods at run time. This data-integration mechanism involves two components, a Data Broker and a Mapping Interpreter, that operate at the data level and at the ontological level, respectively.

We devised the Data Broker as a software component that uses the data-sources ontology to allow problem solvers to read data from a variety of sources transparently at runtime. The Data Broker queries the data-sources ontology for the description of a particular data source and constructs a stream of uniform data objects from the raw data. First, the Data Broker accesses and retrieves data in their original location, based on the metadata describing the low-level data classes in the data-sources ontology. Second, the Data Broker formats and groups the data as specified in the data-sources ontology and packages the data with the appropriate context annotations to create data objects that are syntactically uniform in format and semantically unambiguous. The objects supplied by the Data Broker can then be fed directly to problem solvers that need to operate on them.

Some problem solvers can operate readily on the data supplied by the Data Broker. However, many of the surveillance methods in our library expect data in a format, conceptualization, or level of granularity that is often different from the detailed, lower-level data elements provided by the Data Broker. In this case, supplying problem solvers with data in the appropriate representation requires mapping and transforming data from the format that the Data Broker outputs to whatever representation is expected by the problem solvers.

We have designed an ontology of generic mapping relations that defines the relationships between the input requirements of a method and the corresponding domain, or data, constructs (Crubézy and Musen 2003). This mapping ontology enumerates the types of transformations that enable data sources and data elements to match the particular input–output specifications of different problem solvers. For each category of data, specific mapping relations define the transformation of data elements into runtime inputs of problem solvers. These transformations range from the simple renaming of domain-specific elements to the corresponding terms used by the method to the composition of lexical or numerical expressions to match method terms. For example, when configuring a method to aggregate different data streams, where each stream reported on different 911 dispatches, we created a set of mappings to transform the contents of the different data streams into individual events as required by the aggregation method.

After creating a knowledge base of data-to-method mapping relations, incoming data elements must be translated into a set of input instances for use by the particular method. A Mapping Interpreter applies the mapping relations to the data instances to produce the specific data passed to the method. The Mapping Interpreter processes the mapping relations for each data group and each problem solver, and then generates streams of individual events that are ready to be processed by the problem solvers. The Mapping Interpreter therefore reformats and reconceptualizes the original data to conform to the input requirements of each problem solver. As a result of employing both the Data Broker and the Mapping Interpreter, problem solvers are provided with streams of semantically consistent data that conform to the problem solvers' input requirements. Together, the Data Broker and the Mapping Interpreter provide a semantic wall between analytic methods and the raw data on which they operate. Our approach makes it possible for the analytic methods to make meaningful computations over disparate types of surveillance data without the need for major reprogramming every time that we provide BioSTORM with a new data source or that we develop a new problem-solving method for the library.

A Control Structure for Deploying Surveillance Methods

As a central piece of the prototype BioSTORM infrastructure, we developed a control structure, known as RASTA, that coordinates the flow of data from their raw representations to appropriate problem solvers, via the Data Broker and Mapping Interpreter. The control structure unifies disparate data, possibly distributed among various sources, into semantically uniform data streams, maps and supplies these streams to multiple problem solvers, and deploys these problem solvers to conduct surveillance. The control structure is based on the Linda model (Carriero and Gelernter 1989) to provide a distributed means for problem-solver deployment (Buckeridge et al. 2004b). Linda is based on a logically global, associative object memory called a tuple space. Linda provides interprocess communication and synchronization facilities via the insertion and removal of tuples from this tuple space. It effectively implements parallelism with a small number of simple operations on the tuple space to create and coordinate distributed processes. Linda lends itself to producer–consumer algorithms that use distributed data structures, where processes generate data to be consumed by other processes. As a result, this model is very well suited for enabling opportunistic problem-solver deployment for biosurveillance.

Conclusion

Although Bio-STORM is a laboratory prototype, the system demonstrates an end-to-end solution to many of the problems associated with data acquisition, data integration, and data analysis for public health surveillance. The architecture builds solidly on long-standing work in AI concerning the use of ontologies for semantic integration, the deployment of reusable problem-solving methods, the mapping of problem-solving methods to domain ontologies, and parallelization methods for distributed problem solving. We do not claim that BioSTORM offers a fundamental advance in AI. Rather, we claim that BioSTORM demonstrates how established AI methods can lead to the rapid development of a robust computational approach to the analysis of large volumes of disparate, noisy data that is extremely flexible and highly maintainable.

When novel data sources are identified, developers can edit the data sources ontology to incorporate the new data streams in a straightforward manner. When new analytic methods are developed, they can be encoded and added to the RASTA control system quite easily. Unlike the countless existing systems for syndromic surveillance, BioSTORM does not require reprogramming each time a new data source or a new analytic algorithm becomes available; instead, developers simply edit the associated ontologies.

The ease with which the ontology-based approach accommodates changes to the system has implications that extend beyond system maintenance. The major difficulty with current deployed systems for syndromic surveillance is that not one of them has been rigorously evaluated. The homeland-security community has taken it on faith that these systems are useful (Bravata et al. 2004). The BioSTORM architecture offers a modular framework into which developers can drop new problem-solving methods and new data sources, and then measure system performance. The architecture thus makes it straightforward to evaluate the relative contributions of new data sources and new analytic approaches. Although we are eager to deploy BioSTORM in actual surveillance settings, we believe the system may have more immediate payoff in serving as a test bed in which to evaluate new surveillance methods and data streams. To date, the tremendous enthusiasm for deploying syndromic surveillance systems has not been matched by commensurate interest in measuring the effectiveness of alternative approaches.

Acknowledgements. This work has been supported by the Defense Advanced Research Projects Agency.

References

Bravata, D.M., McDonald, K.M. et al. (2004). A critical evaluation of existing surveillance systems for illnesses and syndromes potentially related to bioterrorism. Annals of Internal Medicine 140(11):910-922.

Buckeridge, D.L., Musen, M.A. et al. (2003). An analytic framework for space-time aberrancy detection in public health surveillance data. Proceedings of the American Medical Informatics Association Annual Symposium, Washington, D.C.: 120-4.

Buckeridge DL, Burkom H. et al. (2004a). Evaluation of syndromic surveillance systems: Development of an epidemic simulation model. In: Sydromic Surveillance: Reports from a National Conference; 2003; New York, NY: MMWR 2004;53(Suppl). 137-143.

Buckeridge DL, O'Connor MJ, Xu H, Musen MA (2004b). A knowledge-based framework for deploying surveillance problem solvers. Proceedings of the 2004 International Conference on Information and Knowledge Engineering (IKE'04). June 21-24, 2004, Las Vegas, NV.

Buehler, J. W., Berkelman, R.L. et al. (2003). Syndromic surveillance and bioterrorism-related epidemics. Emerging Infectious Diseases 9(10): 1197-204.

Carriero, N. and Gelernter, D. (1989). How to write parallel programs: a guide to the perplexed. ACM Computing Surveys 21(3).

Crubézy, M. and Musen, M.A. (2003). Ontologies in support of problem solving. Handbook on Ontologies. S. Staab and R. Studer, Springer-Verlag: 321-341.

Gennari, J.H., Tu, S.W., Rothenfluh, T.E. and Musen, M.A. (1994). Mapping domains to methods in support of reuse. International Journal of Human-Computer Studies 41: 399-424.

Lombardo, J., Burkom, H. et al. (2003). A systems overview of the Electronic Surveillance System for the Early Notification of Community-Based Epidemics (ESSENCE II). Journal of Urban Health 80(2 Suppl 1): i32-42.

Pavlin, J. A. (1999). Epidemiology of bioterrorism. Emerging Infectious Diseases 5(4): 528-30.

Pincus Z. and Musen, M.A. (2003) Contextualizing heterogeneous data for integration and inference. In: Proceedings of the American Medical Informatics Association Annual Symposium, Washington, D.C.: 514-8.

Tsui, F. C., Espino, J.U. et al. (2003). Technical description of RODS: a real-time public health surveillance system. Journal of the American Medical Informatics Association 10(5): 399-408.