Finding Relevant Cases in Large Databases of Signals, Time Series, and Clinical Data

MC Villarroel, A Saeed, GD Clifford, GB Moody, RG Mark

Harvard-MIT Division of Health Sciences and Technology, Cambridge, MA, USA

Abstract

To make effective use of web archives such as PhysioNet's multi-terabyte collections of ICU recordings, investigators need robust and flexible tools for finding data relevant to their studies. We have designed and implemented web-based visualization and search tools for this purpose. In addition to selecting recordings with desired demographic, clinical, and technical features, the search tool permits one to locate recordings containing events defined by characteristics of the recorded signals and time series, such as changes in parameters of user-specified amounts over user-specified time windows. By precomputing signal features, and by caching intermediate results, such searches can be made far more efficient than brute-force methods permit. At present, searching a collection of over 17,000 patient recordings requires about 4 seconds for an average query, thus allowing investigators to find useful subsets of very large data collections without major computing efforts.

1. Introduction

Modern intensive care units (ICUs) make use of a wide array of sophisticated instrumentation to record detailed measurements of the pathophysiological state of patients. PhysioNet [1] makes freely available over 40 collections of recorded physiologic signals and time series [2], including two large ICU databases, MIMIC [3] and its even larger successor, MIMIC II [4]. These data are stored in a variety of open flat-file formats that are readable using open-source WFDB software also available from PhysioNet.

In the MIMIC II database, each of the nearly 30,000 patient records starts at admission into the ICU and ends at discharge from the ICU. The record lengths are typically four or five days, but are sometimes several weeks. A database of such size presents significant challenges in organizing, indexing, searching, and annotating individual patient records [5, 6].

This paper presents several of the new software tools we have developed to meet these challenges. They make our large databases, as exemplified by MIMIC II, more useful to other researchers and to ourselves, and they make it possible for other interested researchers to contribute their expertise to the task of identifying and describing the important features of these and similar recordings. These tools have been designed and tested for portability, so that they can be used on any Java-capable platform, including GNU/Linux, Mac OS X, MS Windows, Solaris, and other versions of Unix.

2. Methods

2.1. Architecture

The MIMIC II database is composed of diverse data, including high frequency signals (*waveforms*, such as ECG, continous blood pressure, fingertip plethysmogram, and respiration) lower frequency time series (*trends*, such as heart and respiration rates, systolic and diastolic blood pressure, blood oxygen saturation, and core temperature), test results (*lab data*, such as blood tests, diagnostic ECGs, and radiology reports), and text (*narratives*, such as problem lists, progress reports, admitting notes, and discharge summaries). We have chosen to store waveforms and trends in web-accessible, WFDB-compatible flat files, and to store precomputed wave and trend indices, lab data, narratives, and diagnostic codes derived from narratives and lab data, in a relational database.

The middleware *data abstraction layer*, a key feature of this architecture, directly interacts with the various storage drivers and presents *virtual patients* to the user applications. Each virtual patient is an object (with properties and actions) with which a user application can interact.

Within the USA, where PhysioNet is based, all exchanges of medical data either must comply with privacy protection regulations defined by the Health Insurance Portability and Accountability Act (HIPAA) [7], or must be exempt from those regulations as a result of having been deidentified. Our goal is to deidentify MIMIC II as thoroughly as possible, and to post the portions that we can be certain have been deidentified on PhysioNet, where they can be obtained without restriction. At present, the

waveform and trend data are available in this way, but the other components can be made available only via data use agreements following HIPAA guidelines.

For this reason, applications get access to virtual patients only through a secure and encrypted channel. Waveforms are transferred in binary WFDB-compatible format, taking advantage of data compression and high throughput. Other data are accessed using an XML-based messaging system.

2.2. Application deployment

Portable, robust applications with clean and effective user interfaces and high-throughput data services present several design challenges. Browser-based solutions do not provide acceptable throughput or sufficient flexibility in user interface design, and pure browser applications are often neither portable nor robust. Users' experience with highly interactive server-based applications is often unsatisfactory because of network latency. Client-side technologies that support better interaction with server-based applications are often not portable across browsers or operating systems, and their use as components of complex, highly interactive applications makes sharing, testing, and reusing components difficult.

Our application framework uses the Java Network Launching Protocol (JNLP) [8], which provides a cross platform and browser independent technology for deploying applications to the client.

With JNLP, the user can launch the application with just a single click from a web browser. If the application is not already installed in the local computer, it will automatically download and install all necessary files and dependent libraries. Applications are cached in the user's local computer, so they can subsequently be run directly from the local copy without consuming major network resources.

Any time the user launches the application, the middleware will check for new updates and software releases. If necessary, it will automatically download and install them without the user intervention, ensuring that the most current version of the application is always present in the client computer.

A very important feature of the execution environment is that it runs inside a secure and sealed *sandbox* container that requires standard code-signing techniques when access to local resources such as the clipboard or hard disk is needed.

2.3. Concurrency and perfomance

As the set of databases available on PhysioNet grows, it is very important that the software infrastructure's response time remain acceptable for multiple simultaneous users. A multi-user, multi-threaded framework permits several asynchronous work flows, divided into (mostly) sequential processes and executed in parallel in multiple threads.

On the server side, multiple threads permit the software not only to accept requests from multiple users, but also to share common data, such as database buffers or state information, improving response time and achieving better throughput. The process of creation and termination of threads uses automated throttling thread pools [9].

2.4. Search procedure

Since the data abstraction layer provides uniform access to all components of the database, searches can be constructed without hardcoded structured query language (SQL) statements dependent on the specific formats or data storage implementation (see figure 1. Query results are appended to their respective *virtual patients*, so that interactive visualization and analysis applications can later retrieve the data from them securely.

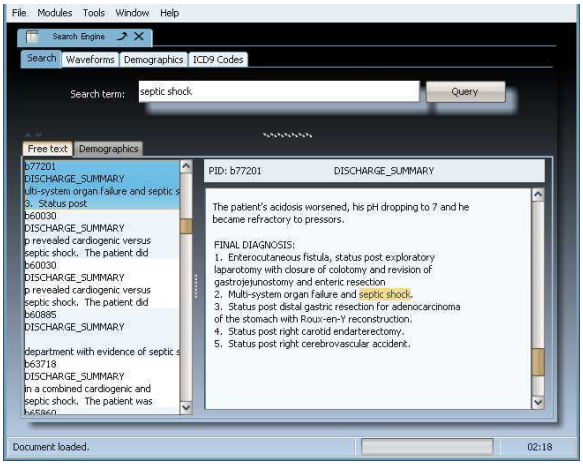

Figure 1. Search application.

Using multiple threads in the client, we also avoid some common graphical application problems such as GUI *freezes* due to asynchronous event requests such as reading data from hard disks. As shown in figure 2, when a user performs a search from the GUI, the application continues to interact with the user. Meanwhile, a separate thread is created and is added to a non-blocking queue, where it waits for the database I/O operation to complete.

Finding records of interest in a database such as MIMIC II may require searching not only for keywords, but also for features of the signals and time series, such as a change in heart rate by some user-specified absolute or relative amount, occuring over some user-specified time interval. Precomputed wavelet decomposition of the signals and time series [10] enables such searches to be performed or-

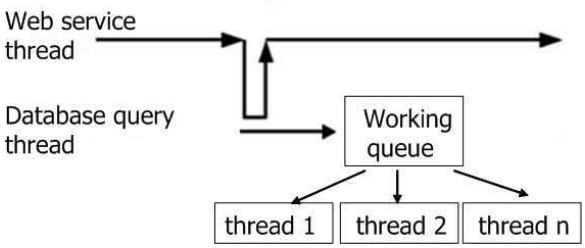

Figure 2. Concurrency in database queries

ders of magnitude more rapidly than would be possible by brute-force search.

At present, searching a collection of over 17,000 patient recordings requires about 4 seconds for a typical query. This speed permits efficient iteratively refined data mining, in which the results of a search may suggest a revised search with additional or altered criteria. Rather than investing extraordinary effort to define the ideal search criteria for a study a priori, a researcher can use the search engine to explore the range of variation of the parameters of interest, and to find useful subsets of very large data collections without major computing efforts.

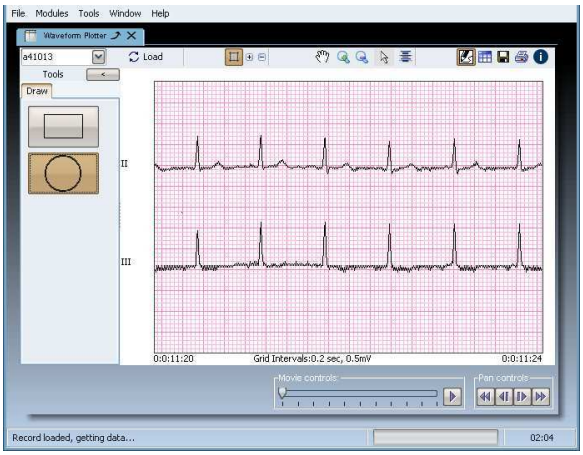

Figure 3. Waveform viewer.

2.5. Visualization

Visualization and analysis of waveforms and similar high-density data requires not only the design and implementation of highly efficient and optimized communication protocols and algorithms, but also the development of intuitive and user-friendly user interfaces [11].

Figure 3 shows an interactive waveform viewer we have designed and implemented. Its main features are:

- Standard clinical scales and grid
- Interactive pan, zoom, and plotting color customization.
- Variable-speed smooth scrolling
- Editing (notes, annotations, and drawings)
- Export/save/print capability (CSV, text, JPEG images, other formats)

As with the rest of the tools in the framework, the waveform viewer is highly integrated with the search engine, and as such, can be launched as a stand alone waveform viewer or as an embedded plotter in the search engine.

3. Future work

The middleware data abstraction layer described in this paper continues to evolve as we add new web-based tools for database exploration. Some of the features that are currently under development and integration include:

- *Interactive annotation editing*, to allow users to identify and characterize physiologic events of interest as well as persistent conditions (problem lists).
- *Searches incorporating Unified Medical Language System (UMLS) dictionaries*, to improve the effectiveness of searches in narrative data and to aid in the process of coding events [5].
- *Programming interfaces* for C, Matlab and Java to simplify using signal processing software developed for PhysioNet within the framework discussed here.
- *Searches incorporating complex temporal criteria*, such as sequences of changes in parameters, or sets of roughly simultaneous changes.
- *Search by example*, to find other records resembling some given record or set of records, with or without additional search criteria, as a step toward development of data-driven medical decision support algorithms.

4. Discussion and conclusions

Large databases such as MIMIC II present unique challenges for searching and visualizing patient records. Finding records of interest for specific studies in such large databases is made vastly easier by effective data mining tools. Our *virtual patient* model changes the way queries are performed, by presenting data gathered from diverse sources in a uniform way that avoids the need for SQL and that permits construction of queries interactively and iteratively.

While installation and maintenance of typical desktop applications requires knowledge and familiarity of the computing environment, our model allows the deployment and update of such applications to be done automatically, without user intervention.

The availability of portable, robust, efficient, effective, customizable, and easily used applications for exploring large databases of signals, time series, and clinical data makes such databases more useful to the research community at large. In particular, this permits interactive and iterative refining of data mining processes to identify patient

cohorts with cases relevant to specific studies, and to identify particular events of interest.

Acknowledgements

This work was supported in part by the U.S. National Institute of Biomedical Imaging and Bioengineering (NIBIB) and the National Institutes of Health (NIH) under Grant Number R01 EB001659, and Philips Medical Systems. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of the NIBIB, the NIH, or Philips Medical Systems.

References

[1] PhysioNet: The Research Resource for Complex Physiologic Signals. http://physionet.org/.

[2] Goldberger A, Amaral L, Glass L, Hausdorff JM, P PI, Mark RG, Mietus J, Moody G, Peng C, Stanley H. PhysioBank, PhysioToolkit, and PhysioNet : Components of a new research resource for complex physiologic signals. Circulation 2000;101(23):e215–220.

[3] Moody GB, Mark RG. A database to support development and evaluation of intelligent intensive care monitoring. Computers in Cardiology 1996;23:657–660.

[4] Saeed M, Lieu C, Raber G, Mark RG. MIMIC II: a massive temporal ICU patient database to support research in intelligent patient monitoring. Computers in Cardiology September 2002;29:641–644.

[5] Shu J, Clifford GD, Saeed M, Long WJ, Moody GB, Solovitz P, Mark RG. An Open-Source, Interactive Java-Based System for Rapid Encoding of Significant Events in the ICU Using the Unified Medical Language System. Computers in Cardiology 2004;31:197–200.

[6] Oefinger MB, Mark RG. A web-based tool for visualization and collaborative annotation of physiological databases. Computers in Cardiology September 2005;32:163–165.

[7] http://www.hhs.gov/ocr/hipaa/.

[8] Marinilli M. Java Deployment with JNLP and Webstart. Sams, 2001. ISBN 0-672-32182-3.

[9] Goetz B, Peierls T, Bloch J, Bowbeer J, Holmes D, Lea D. Java Concurrency in Practice. Addison Wesley Profesional, 2006. ISBN 0-321-34960-1.

[10] Saeed M, Mark RG. Efficient hemodynamic event detection utilizing relational databases and wavelet analysis. Computers in Cardiology 2001;28:153–156.

[11] Powsner SM, Tufte ER. Graphical summary of patient status. Lancet 1994 Aug 6;344:386–389.

Address for correspondence:

Mauricio Villarroel
Massachusetts Institute of Technology
77 Massachusetts Avenue, E25-505
Cambridge, MA 02139
maurov AT mit DOT edu