Published in final edited form as:

IEEE Trans Biomed Eng. 2012 January ; 59(1): 234–240. doi:10.1109/TBME.2011.2170986.

Design and Application of a Generic Clinical Decision Support System for Multiscale Data

Jussi Mattila*,

VTT Technical Research Centre of Finland, Tampere, Finland

Juha Koikkalainen,

VTT Technical Research Centre of Finland, P.O. Box 1300, Finland

Arho Virkki,

VTT Technical Research Centre of Finland, P.O. Box 1300, Finland

Mark van Gils [Member IEEE],

VTT Technical Research Centre of Finland, P.O. Box 1300, Finland

Jyrki Lötjönen

VTT Technical Research Centre of Finland, P.O. Box 1300, Finland

Alzheimer's Disease Neuroimaging Initiative

Juha Koikkalainen: juha.koikkalainen@vtt.fi; Arho Virkki: arho.virkki@vtt.fi; Mark van Gils: mark.vangils@vtt.fi; Jyrki Lötjönen: jyrki.lotjonen@vtt.fi

Abstract

Medical research and clinical practice are currently being redefined by the constantly increasing amounts of multiscale patient data. New methods are needed to translate them into knowledge that is applicable in healthcare. Multiscale modeling has emerged as a way to describe systems that are the source of experimental data. Usually, a multiscale model is built by combining distinct models of several scales, integrating, e.g., genetic, molecular, structural, and neuropsychological models into a composite representation. We present a novel generic clinical decision support system, which models a patient's disease state statistically from heterogeneous multiscale data. Its goal is to aid in diagnostic work by analyzing all available patient data and highlighting the relevant information to the clinician. The system is evaluated by applying it to several medical datasets and demonstrated by implementing a novel clinical decision support tool for early prediction of Alzheimer's disease.

Keywords

Clinical diagnosis; decision support systems; software architecture; supervised learning

* jussi.mattila@vtt.fi.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

I. INTRODUCTION

ADVANCES in multimodal data acquisition instrumentation have resulted in a deluge of data that have contributed significantly to scientific research of diseases [1]. It has also altered the daily clinical practice by increasing the amount of patient information that clinicians must manage. Everything from questionnaire answers to laboratory results and information obtained with sophisticated imaging methods must be considered when making diagnostic decisions. Furthermore, new knowledge about diseases is unveiled at an unparalleled rate, making the deliberate application of evidence-based medicine a challenging and time-consuming effort.

One approach to managing this complexity is to develop detailed computer-based multiscale modeling, simulation, and analysis systems. Multiscale is defined here as patient data obtained at several scales, e.g., with genetic, molecular, structural, and neuropsychological tests. Models that describe phenomena of human physiology at a particular scale may be combined, usually with considerable effort, for understanding of larger entities [2]. Physiological multiscale models are often custom-built to target a single organ, disease, or condition, and they help develop treatments, biomarkers, and even personalized disease models for use in clinical work. They have already proven useful and shall remain the focus of much of future research [3], [4].

The increasing number and scale of measurements can improve one's understanding of a system even without detailed physiological modeling. There are established machine learning methods that can classify a patient as being healthy or diseased or provide the probability of having a disease when trained with previously diagnosed patient data [5]. Recent research has introduced mathematical and statistical models, which derive composite disease indicators from quantitative multiscale and multimodal data. Their goal is to give prognoses, e.g., in the context of prostate cancer [6] or Alzheimer's disease (AD) [7]. An alternative method is to employ data-driven techniques that divide all the experimental data into components for analysis. Study of the components can provide insight into the subsystems and ultimately to the system as a whole. Such an approach, able to handle empirical patient data and implemented within a clinical decision support system (CDSS), could transform existing patient data into knowledge applicable in the clinical setting [8].

One major hurdle for the widespread use of these systems and CDSSs in general is that data collected at different clinics vary considerably. Consequently, most CDSSs for medical diagnostics are purpose-built expert systems targeting a single condition or a family of diseases, and also require a particular set of data [9]. Generic CDSSs for clinical diagnostics have also been developed, traditionally employing Bayesian inference [10], text-mining methods [11], case-based reasoning [12], or fuzzy cognitive maps (FCM) [13]. But even with the more generic CDSS systems, most require definition of disease-specific model parameters by domain experts before they can be put into use.

This manuscript describes a data-agnostic clinical decision support system, implemented as a reusable software library. The software library uses a statistical approach to analyze multiscale data and combine them into an aggregate representation interpretable by a

clinician. It supports heterogeneous patient data of virtually any type and scale and allows clinicians to study the system simultaneously as a collection of components and as a whole. The library has been designed to easily support several diseases, requiring minimal amount of configuration. The first application prototype developed using the proposed decision support library is a CDSS tool for early diagnosis of AD. The statistical methods are validated using data from several medical datasets and the clinical applicability of our proposed system is demonstrated by evaluating the implementation of the CDSS tool.

The main contributions of this work are the description of the generic decision support software library, the statistical method behind it, and evaluations of classification and computational performance of the proposed system using several medical datasets. A more thorough analysis of the statistical method and its relationship to established machine learning methods with regards to AD is available in [14].

II. MATERIALS AND METHODS

A. Evaluation of Disease State

In this work, a data-agnostic statistical disease modeling method has been developed. It combines heterogeneous multiscale data to compute a value in the interval $[0,1]$, indicating a patient's disease state, i.e., the location or rank based on data, in relation to previously known control (healthy) and positive (disease) populations. It is intended to be used mainly with quantitative features, such as standardized questionnaire answers, laboratory analysis results, automatically quantified biomedical data, and outputs of personalized disease model simulations. It can be considered a supervised classifier, where patient data are compared to previously diagnosed data. In its development, equal emphasis was given to classification accuracy and to clinical interpretability of the results.

Given the heterogeneous patient data from a single test at a single time point, e.g., an individual neuropsychological test or laboratory analysis results of a blood sample, as $x_1, x_2, \dots, x_n$, we define the n -variable scalar valued disease state index (DSI) function as a weighted mean

$$\text{DSI}(x_1, x_2, \dots, x_n) := \frac{\sum_{i=1}^n \text{Rel}(i) \text{Fit}(x_i)}{\sum_{i=1}^n \text{Rel}(i)} \quad (1)$$

where $\text{Rel}(i)$ is a *relevance* function providing the weighting between $[0,1]$ for variable i and $\text{Fit}(x_i)$ is a *fitness* function providing a nonlinear transformation of value x_i into *fitness* space $[0,1]$.

A *fitness* function computes the location, i.e., rank, of an individual variable x_i relative to values of the same variable in two different populations, denoted as controls C_i and positives P_i . Our system currently supports scalar, ordinal, and categorical (including boolean) variables, but could be extended to support others, such as value lists and complex values, by deriving appropriate *fitness* functions. Let us consider a scalar variable where the

progression of a disease tends to increase its value (see Fig. 1). For these, *fitness* is defined as a monotonically increasing function

$$\text{Fit}(x_i) := \frac{L_P(x_i)}{L_P(x_i) + R_C(x_i)} \quad (2)$$

where $L_P(x_i)$ is the left integral of probability density function (PDF) for positive class values P_i and $R_C(x_i)$ is the right integral of PDF for control class values C_i . Derivation of the *fitness* function can be conducted in an analogous manner for ordinal variables. For a categorical variable $x_i \in \{\Omega_1, \dots, \Omega_n\}$, we use as *fitness* the conditional probability of the subject belonging to the positive population in the case of observing $\Omega = x_i$.

The weighting factors of DSI, i.e., *relevancies* of variables, are determined by the variables' ability to correctly classify between the known classes C_i and P_i , and are independent of the patient data. *Relevance* is defined for scalar and ordinal values that increase with disease progression as

$$\text{Rel}(i) := \max\{0, L_C(x_i^*) + R_P(x_i^*) - 1\} \quad (3)$$

where $L_C(x_i^*)$ is the left integral of PDF for control values C_i and $R_P(x_i^*)$ is the right integral of PDF for positive values P_i at the decision threshold x_i^* (shown in Fig. 1). For categorical variables, *relevance* is the classification accuracy of training cases given the category of the independent variable.

To combine data from multiple tests and/or multiple scales, DSI values obtained from (1) are recursively inserted back into (1) as new variables, using several levels of recursion for granularity. Recursive evaluation provides *fitness*, *relevance*, and DSI values for a tree of data, where the leaves and branches represent multiple scales but converge to a common root describing the whole system. This tree of data can be rendered for quick visual interpretation of multiscale data, using colors and shapes to quickly distinguish patient state and the relevance of all tests and variables. The nodes can also be ordered according to *relevance* to show the most important features at the top (see Fig. 2).

In summary, DSI uses available multiscale data to model the state of having a disease. It does so first with the individual measurement values, then transforms the values nonlinearly to a common classification space and combines them within that space to obtain aggregate results. The recursive computation produces classification results at multiple levels of abstraction, which can be visualized using a tree hierarchy.

B. Decision Support Library

We have developed a software library implementing the DSI computational method and supporting features using the C# language (see Fig. 3). The library is context independent, and thus is applicable to several domains.

Since the DSI can use any available multiscale data, the library supports accessing multiple data repositories with a layered approach. Data access implementations, called persistence stores^(a) in Fig. 3, are free to connect with data sources in any way that is needed, e.g., through an object relational mapping (ORM) service, web services, or simply reading a flat text file. An interface defines how the persistence stores can transfer data to and from the library.

A data definition layer^(b) comprises descriptions of entries (e.g., types of tests done to a patient) and feature values (types of individual data points) within those entries. Definitions are application-specific metadata and must be configured in source code or by Extensible Markup Language (XML) when initializing the library for use. In addition to all features existing at the leaf nodes, the organization of the DSI tree hierarchy is also described within this layer. The actual data that are analyzed are contained within another layer^(c), where all the subjects, entries, and feature values are represented by matching object instances, as described in Table I.

Performing DSI computations requires the library to construct control and positive classes in a generic manner, using entities from one or more persistent stores that provide training data. For this, we have developed a rule-based grouping system^(d), where a grouping rule interface is called to check whether a training entity belongs to a particular class, e.g., to healthy controls or Alzheimer's disease patients. A CDSS tool using this decision support library is aware of the context and is responsible for defining the group forming rules, e.g., "if diagnosis equals AD, assign patient to group AD." A graphical user interface (GUI) component is available in the decision support library to allow interactive modification of the rules that have been implemented so far. If necessary, new rule implementations can be created. They are able to use all available patient information when deciding whether he or she is to be included in a training class or not.

After applying grouping rules, entities in control and positive classes are known^(e). Now, the library must collect all types of values from the entities in a generic manner. For this, we have developed a sampling system^(f), where sampling policies control how data from a single entity are chosen for training. One can, e.g., use the mean of all scalar values for a particular feature or pick the value that was obtained most recently. As with the grouping rules, the sampling policy implementations can be configured with a GUI component and new ones can be implemented in source code if complex sampling policies are necessary. Custom grouping and sampling may be used, e.g., for personalized healthcare, where stratification is employed to collect feature values with age and gender constraints.

Now, having the training data^(g), data from the patient we are studying, and with the definition layer^(b) describing the feature hierarchy, the library has all the necessary information for evaluating the DSI^(h). Training data obtained through grouping and sampling is organized in the tree hierarchy where the leaves contain actual measurement values for the training set. *Fitness* and *relevance* are evaluated at the leaf level, DSI and *relevance* values in internal nodes are computed recursively, and, finally, a total DSI value for the whole dataset at the root of the DSI tree is obtained.

The library provides implementations of GUI components for displaying DSI trees⁽ⁱ⁾, data distributions^(j), entry timeline^(k), and entry details^(l). These are implemented on top of the logic tier using Windows Presentation Foundation (WPF) platform.

C. Data Access Implementations

Currently, there exist two implementations of persistence stores^(a) for accessing patient information to be used with the decision support library. One of them uses an entity-attribute-value (EAV) scheme, which is a common methodology for database design in healthcare applications, thanks to its applicability to storing heterogeneous and sparse patient data [15]–[17]. EAV is well suited for querying data of individual patients, but it is well known to be inefficient for bulk queries, which are needed for collecting large quantities of training data [18]. These require the use of a normalized database where the patient and all record types are represented by their own tables [19]. Unfortunately, this is a conflicting requirement for the decision support library, which strives to be a generic one, accepting any kind of data from any clinic to be incorporated into it. To overcome the conflicting requirements, a normalized database and persistence store generators have been developed to go along with the library. They are based on C# language features, such as partial classes and reflection [20], with Entity Framework 4 (EF4) [19] and Text Template Transformation Toolkit (T4) [21] engine used for generating all the necessary constructs without hard-coding any data descriptions. Reflection is a mechanism in object-oriented programming languages that is used for examining, instantiating, and using unknown types. Partial classes allow splitting class definitions to several source files. It is often used to combine machine generated source code in one file with manually written source code in another.

The process of generating normalized databases utilizes the data definition layer^(b), which is also used within the library for describing the CDSS data organization. A T4 script reads the data definitions and automatically transforms this metadata to database generation commands, which can be executed to create a new database containing data tables adhering to the given data definitions. With the database structure in place, one can create, using EF4, an object relational mapping (ORM) that allows writing and reading data in the database tables. More specifically, the EF4 tooling environment builds a conceptual model of the database by inspecting its structure and generates the necessary code for transferring data between the database and an application using the data. The EF4 generated conceptual model uses strongly typed C# classes, again working against the requirement of providing a generic decision support library. With strongly typed classes, it is normally required to explicitly declare the type of the class before using it, which in this case is impossible since the database structure is unknown to the library. To overcome this, another T4 script is used for generating partial class definitions that augment the EF4-generated conceptual model classes. The partial class definitions add functionality that allows the augmented object instances to be created and manipulated, using reflection, in a manner that can be considered weakly typed. Through these mechanisms and with information from the data definition layer^(b), generic implementations of persistence stores are able to access normalized databases and transform patient data contained within those into the data structures (entities/

patients, entries, and feature values) used by the decision support library. Finally, there are tools to populate persistence stores^(a) with data from other persistence stores as necessary.

Together, the decision support library and data access implementations form a data-agnostic end-to-end system, which can generate and populate appropriately designed databases based on the data definitions and provide evidence-based decision support using the statistical DSI method.

D. Evaluation of the Proposed CDSS

Data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.ucla.edu, accessed September 2, 2010). Primary goal of ADNI has been to measure the progression of mild cognitive impairment (MCI) and early AD using biomarkers, and clinical and neuropsychological assessment. MCI is a heterogeneous state of cognitive decline, with multiple possible outcomes and increased risk of AD [22]. ADNI recruited approximately 400 people with MCI to be followed for 3 years, in addition to recruiting 200 normal elderly individuals and 200 AD patients.

From the MCI patients recruited to ADNI, this study included those whose last clinical diagnosis during the study was still MCI or had converted to AD, forming the classification groups of stable MCI (SMCI, $n = 190$) and progressive MCI (PMCI, $n = 154$, average time to getting AD diagnosis: 19 months), respectively. Using baseline measurements alone, we tested our method's ability to predict conversion to AD using sparse multiscale measurement data that included neuropsychological tests, magnetic resonance imaging data, molecular test data, and genetic test data (see Table II).

The ability of DSI to predict AD was compared to three reference classifiers; support vector machine (SVM), Naïve Bayes, and Logistic Regression (LR). All methods were given exactly the same data. Data preprocessing, parameter search, and feature selection was done for the reference classifiers to attain the best performance possible. The generic DSI method has been designed not to require preprocessing of any kind, and was used as such. Ten iterations of 10-fold cross-validation were done to obtain robust performance metrics.

In addition to the MCI dataset, we tested the DSI method with three other medical datasets (Pima Indian Diabetes, Cleveland Heart Disease, and Hepatitis) available online [23]. Performance with these datasets was compared to publicly available benchmark results [24]. It is not possible to make a completely objective comparison between the benchmark values and the DSI method since many of the reported values are expressed only as a single number giving the classification accuracy, without standard deviation or information about the validation process. Also, some benchmark results were computed only after excluding subjects with missing values. To robustly assess the DSI method, ten iterations of 10-fold cross-validation were performed with all available data and compared against the best benchmark method whose standard deviation was available, and against the average of benchmark methods that performed better than a simple majority classifier, i.e., one that assigns every case to whichever class is in the majority in the training set.

Applicability of the software library was demonstrated by developing a CDSS tool for early prediction of AD. The complexity of implementation work was evaluated qualitatively and the computational performance of the interactive DSI method was measured quantitatively on a laptop PC with Windows XP SP3, 2 GB of memory, and a 2.4 GHz dual core processor.

III. RESULTS

A. Classification Performance

With the MCI dataset from ADNI, the DSI method performed on a level similar to established machine learning methods, as seen in Table III.

Results obtained with other medical datasets show that the DSI method tends to perform slightly worse than the best benchmark methods, but similar to the average of them. With the diabetes dataset, the best benchmark method was SVM. For heart disease data, the maximum was obtained with a 28-nearest neighbors (k-NN) classifier, using Euclidean distance, and trained only with a subset of features. With the hepatitis dataset, accuracy was best with an 18-NN classifier, this time using Manhattan distance. Results of these evaluations are listed in Table IV.

B. Implementing the CDSS Tool

Relying on the generic decision support library for much of the necessary functionality, a prototype of a CDSS tool for early prediction of AD was developed. The prototype uses two persistence stores that connect to local databases, one using EAV scheme that provides MCI patients for analysis, and a normalized database for accessing training data. Definitions of entries and feature values are described in an XML file and provided to the decision support library during initialization of the application.

The tool provides a comprehensive overview of all available patient data to clinicians. GUI components from the library visualize entries, the DSI tree, and data distributions on a single screen. The patient details panel and the rendering of brain MRI images were custom built for this application. From the user interface, clinicians can select entries to see the data in more detail, select nodes from the DSI tree to see patient and training data distributions, change classification groups, and change included features to customize classification. In summary, the tool allows mining of multiscale patient data and evidence-based study of their relation to known Alzheimer's disease profiles.

The software library facilitated rapid implementation of the CDSS prototype. Taking it into use required configuration of persistence stores and data definitions, providing the necessary sampling and grouping rules, and finally wiring the GUI components into the application.

C. Computational Performance of the DSI Implementation

Training of the DSI model and computation of the initial set of DSI values was done for all patients sequentially, taking on average 860 ms/patient (standard deviation 74 ms). Re-evaluation of DSI values after user initiated exclusion or inclusion of a feature was virtually instantaneous, consistently taking less than 1 ms. Grouping and sampling of training data, including the necessary queries to the database, took on average 10 s. This is done only once

after the application launches, but could be performed again if the training data are changed while running the application.

IV. DISCUSSION

To the authors' knowledge, there are no other CDSS tools or decision support libraries for clinical diagnostics developed with a similar philosophy, i.e., using any available sparse and unprocessed patient data, and not requiring manual tuning or decision parameters defined by clinical experts. To use the decision support system presented here, one only needs data definitions, which can in several cases be derived in a straightforward manner, using the structure of the original data. Data hierarchy definitions can be modified manually if a particular organization is preferred. Computer-based methods for organizing the data hierarchy could also be developed, possibly grouping features automatically along the dimensions of a disease, e.g., effect to motor dysfunction or to delayed recall performance. Further studies are required to assess the effect of different hierarchy structures on the classification accuracy of the statistical DSI method behind the library.

The generic clinical decision support library was found to be a good basis for developing a CDSS tool for early diagnosis of AD. Features of the library aim to support clinical requirements, e.g., they accommodate workflows where patient data are collected sporadically. The statistical methods are not computationally intensive, and could be further optimized with parallelization. Computational performance of the decision support library is more limited by access to training data. Retrieving bulk patient data for training sets in a generic manner was made feasible by developing tools and defining processes that can be used for creating and populating normalized databases from existing electronic datasets.

The DSI method behind the decision support library was able to provide values for quickly interpretable visualizations of multiscale data without compromising prediction accuracy. The visualizations were designed to be transparent, i.e., to clearly disclose the origin of the derived values, since even accurate diagnostics obtained with a black box classifier are not very easily applied in clinical practice. Compared to the reference classification methods, the DSI also emphasizes clinical inter-pretability by 1) providing information about all subsystems of different scales (e.g., genetic, molecular, structural, and neuropsychological) individually and also as a part of the whole, 2) computing a rank of the patient data in relation to diagnosed populations instead of maximizing class separation, which leads to 3) consistency in output that should reflect the magnitude of changes in the raw data. In addition to highlighting important details to clinicians, the DSI and *relevance* values can facilitate building of expert systems.

Classification accuracy of the DSI was found comparable to benchmark methods when applied to various medical datasets, even though it is designed not to require feature selection or searching of optimal classifier parameters. In other words, the generic DSI method obtained classification accuracies close to the best benchmark results, which were manually tuned to work with the given data as well as possible. The relatively low classification accuracies with MCI data are in line with other studies [25] and underline the fact that data alone are not enough for reliable prediction of conversion from MCI to AD at

an early phase of the disease. This is also true for ADNI data, partly due to a relatively short follow-up time and also due to errors in the diagnoses which have not been confirmed pathologically. Correlation between features was also considered. It appears that the tree hierarchy and the recursion resulting from it partially nullify issues due to correlation. For datasets with a large number of features, we have implemented a method that explicitly addresses correlation by applying principal component analysis (PCA) to the leaf nodes of the data hierarchy. In the evaluation datasets, this did not, however, increase classification accuracy.

Healthcare is slowly moving towards electronic health records. Eventually, patient data could be automatically loaded for analyses inside a tool such as this. A clinician diagnosing a patient would not need to observe hundreds of individual measurements at different scales, available from several sources. Instead, they could see all available data at once, hypothesize a disease, and immediately see which data are relevant in that context and which point toward the disease. This could save both time and frustration from information overload. For now, manual work is needed, either entering patient records into the tool, implementing a custom persistence store implementation, or implementing a data adapter which reads existing electronic sources of a particular clinic into a database supported by the library. This limits the presented solution to specialist clinics in the immediate future. The authors also acknowledge that routinely collected clinical data contain more artifacts and missing information than research data that affect the performance of the methods. Therefore, there are plans for future studies using less well-curated patient data from realistic sources.

The main disadvantage of the presented DSI method and the decision support library implementation is that in addition to the patient measurements for analyses, they require properly validated datasets for control and disease cases. This training data could be local to a particular clinic, but could also be collected regionally or nationally, greatly decreasing the burden of creating validated training datasets. The authors believe that data obtained in research studies should be a good starting point for compiling the initial training datasets.

Another limitation of the proposed system is that currently the library has proper support for two-class problems only. Future research will address how these methods are appropriately applied when multiple diseases are in consideration, which is a clinically important requirement for differential diagnostics.

V. CONCLUSION

In this manuscript, the design and implementation of a generic decision support system was presented. It is implemented as a reusable software library employing a statistical disease state modeling method, which is able to robustly analyze heterogeneous multiscale patient data with minimal preprocessing. The context-agnostic data access, analysis, and visualization methods allow the library to be rapidly applied in several contexts. When presented with a new problem or data, there is no searching of parameters, handling of missing values, or development of new user interfaces. As long as definitions of the data and the data itself are provided to the library, it can organize available values and construct interactive views that provide analyses of the recently defined information to clinical

decision makers. The ultimate goal is to provide evidence-based decision support for clinicians during diagnostic work. Application of the decision support library was demonstrated by developing a prototype CDSS tool for early prediction of AD. We are currently evaluating the prototype at two memory clinics in Europe, comparing it to traditional diagnostic methods. We are also applying the DSI method and the decision support library to several other datasets to assess their robustness more comprehensively.

Acknowledgment

The authors thank participants of project PredictAD, funded partially by the 7th Framework Program by the European Commission under the ICT theme Virtual Physiological Human (Grant Agreement 224328). Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.ucla.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in analysis or writing of this report. A complete listing of ADNI investigators can be found at: http://adni.loni.ucla.edu/wpcontent/uploads/how_to_apply/ADNI_Authorship_List.pdf.

This work was supported in part by the 7th Framework Program by the European Commission (<http://cordis.europa.eu/ist>; EU-Grant-224328-PredictAD; From Patient Data to Personalized Healthcare in Alzheimer's Disease). Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904).

REFERENCES

- [1]. Chen H, Fuller S, Friedman C, and Hersh W, Medical Informatics: Knowledge Management and Data Mining in Biomedicine. New York: Springer, 2010.
- [2]. Southern J, Pitt-Francis J, Whiteley J, Stokeley D, Kobashi H, Nobes R, Kadooka Y, and Gavaghan D, "Multi-scale computational modelling in biology and physiology," *Prog. Biophys. Mol. Biol.*, vol. 96, pp. 60–89, 2008. [PubMed: 17888502]
- [3]. Noble D, "Modeling the heartFrom genes to cells to the whole heart," *Science*, vol. 295, pp. 1678–1682, 2002. [PubMed: 11872832]
- [4]. Deisboeck TS and Stamatakos GS, Multiscale Cancer Modeling. London: CRC Press, 2010.
- [5]. Alpaydin E, Introduction to Machine Learning, 2nd ed. Cambridge: MIT Press, 2009.
- [6]. Madabhushi A, Agner S, Basavanahally A, Doyle S, and Lee G, "Computer-aided prognosis: Predicting patient and disease outcome via quantitative fusion of multi-scale, multi-modal data," *Computerized Med. Imaging and Graphics*, to be published.
- [7]. Ye J, Chen K, Wu T, Li J, Zhao Z, Patel R, Bae M, Janardan R, Liu H, Alexander G, and Reiman E, "Heterogeneous data fusion for alzheimer's disease study," in *Proc. 14th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, Las Vegas, 2008.
- [8]. Bennett C and Dobb T, "Data mining and electronic health records: Selecting optimal clinical treatments in practice," in *Proc. 2010 Int. Conf. Data Mining Las Vegas*, Jul. 2010.
- [9]. Greenes RA Ed., Clinical Decision Support the Road Ahead. New York: Elsevier, 2007.
- [10]. Horrocks JC, McCann AP, Staniland JR, Leaper DJ, and de Dombal FT, "Computer-aided diagnosis: Description of an adaptable system, and operational experience with 2, 034 cases," *British Med. J.*, vol. 2, pp. 5–9, 1972.
- [11]. Barnett GO, Cimino JJ, Hupp JA, and Hoffer EP, "DXplain. An evolving diagnostic decision-support system," *The J. Amer. Med. Assoc.*, vol. 258, pp. 67–74, 1987.
- [12]. Watson I and Marir F, "Case-based reasoning: A review," *The Knowledge Eng. Rev.*, vol. 9, pp. 327–354, 1994.
- [13]. Stylios CD, Georgopoulos VC, Malandraki GA, and Chouliara S, "Fuzzy cognitive map architectures for medical decision support systems," *Appl. Soft Comput.*, vol. 8, pp. 1243–1251, Jun. 2008.

- [14]. Mattila J, Koikkalainen J, Virkki A, Simonsen A, van Gils M, Waldemar G, Soininen H, and Lotjonen J, "A disease state fingerprint for evaluation of alzheimer's diseases," *The J. Alzheimer s Dis*, to be published. Available: <http://iospress.metapress.com/content/kg54325631131n10/>.
- [15]. Beck P, Truskaller T, Rakovac I, Cadonna B, and Pieber TR, "On-the-fly form generation and on-line metadata configuration—A clinical data management Web infrastructure in Java," *Stud. Health Technol. Inform*, vol. 124, pp. 271–276, 2006. [PubMed: 17108536]
- [16]. Brandt CA, Deshpande AM, Lu C, Ananth G, Sun K, Gadagkar R, Morse R, Rodriguez C, Miller PL, and Nadkarni PM, "TrialDB: A web-based clinical study data management system," in *AMIA Annu. Symp. Proc*, 2003, p. 794. [PubMed: 14728299]
- [17]. Nadkarni PM, Brandt C, Frawley S, Sayward FG, Einbinder R, Zeltermann D, Schacter L, and Miller PL, "Managing attribute—Value clinical trials data using the ACT/DB client-server database system," *J. Am. Med. Inform. Assoc*, vol. 5, no. 2, pp. 139–151, 1998. [PubMed: 9524347]
- [18]. Chen RS, Nadkarni PM, Marengo L, Levin FW, Erdos J, and Miller PL, "Exploring performance issues for a clinical database organized using an entity-attribute-value representation," *J. Amer. Med. Inf. Assoc*, vol. 7, pp. 475–487, 2000.
- [19]. Lerman J, *Programming Entity Framework: Building Data Centric Apps with the ADO.NET Entity Framework.*, Sebastopol, CA, O'Reilly Media, 2010.
- [20]. Albahari J and Albahari B, *C#40 in a Nutshell: The Definitive Reference*, Sebastopol, CA, O'Reilly Media, 2010.
- [21]. Vogel P, *Practical Code Generation in .NET: Covering Visual Studio 2005, 2008, and 2010* (Addison-Wesley Microsoft Technology Series). Boston, MA: Pearson Education Inc., pp. 249–284, 2010.
- [22]. Petersen RC, Roberts RO, Knopman DS, Boeve BF, Geda YE, Ivnik R, Smith G, and Jack CR, "Mild cognitive impairment: Ten years later," *Arch. Neurol.*, vol. 66, pp. 1447–1455, 2009. [PubMed: 20008648]
- [23]. Frank A and Asuncion A, UCI Machine Learning Repository, [Online] Available: <http://archive.ics.uci.edu/ml>, 2010.
- [24]. Duch W, Comparison of Classification Results, [Online] Available: <http://www.is.umk.pl/projects/datasets.html>, 2011.
- [25]. Llano DA, Laforet G, and Devanarayan V, "Derivation of a new ADAS-cog composite using tree-based multivariate analysis: prediction of conversion from mild cognitive impairment to Alzheimer disease," *Alz. Dis. Assoc. Dis*, vol. 25, pp. 73–84, 2011.

Fig. 1. Probability density functions of C_i and P_i , the resulting *fitness* (with examples at test outcome values a and b), and the optimal classification threshold x_i^* .

Fig. 2.

DSI tree visualizations for two patients, one healthy, one with AD. Larger node sizes indicate higher *relevance* (i.e., better discrimination of training classes), with irrelevant features omitted. Shades of red indicate similarity of the patient data to the disease population, shades of blue similarity to healthy.

Fig. 3.

Tiers, layers, and components of the generic decision support library, also showing the main direction of data flow.

TABLE I**LIBRARY RUNTIME DATA STRUCTURES**

Instance	Contains	Purpose	Example usage
Entity	Entries	Object of interest	A patient
Entry	Features	Data container	Blood analysis results
<i>Features</i>			
Text	free text	Value container	Verbatim answer
Scalar	double	Value container	Blood pressure
Nominal	category	Value container	Multiple choice question
Ordinal	position	Value container	# of words remembered

Exact composition of entries and features are described in the definition layer. E.g., an entry definition provides a list of feature values that it can contain, while a nominal feature definition defines the list of allowed values.

TABLE II

ENTRIES AND THEIR FEATURE COUNTS FOR EARLY DIAGNOSIS OF AD

Entry	Features	Available ^c	Description
<i>Neuropsychological Tests</i>			
MMSE	30	100%	Mini-Mental State Examination
ADAS	13	99%	AD Assessment Scale
<i>Biomedical Imaging</i>			
MRI	13	89%	Volumes of structures from brain MRI ^a
<i>Molecular Tests</i>			
CSF	2	52%	Amyloid- β and Total Tau from CSF ^b
<i>Genetic Tests</i>			
APOE	2	100%	Alleles of apolipoprotein E

^aMagnetic resonance imaging,^bCerebrospinal fluid,^cPercentage of patients for whom the data existed in the ADNI database at baseline.

TABLE III

CLASSIFICATION PERFORMANCE WITH ADNI MCI DATASET

Method	AUC ^a	Accuracy	Sensitivity	Specificity
DSI	0.75 ± 0.08	0.68 ± 0.08	0.70 ± 0.12	0.66 ± 0.10
SVM	0.75 ± 0.08	0.67 ± 0.07	0.64 ± 0.11	0.69 ± 0.11
Bayes	0.76 ± 0.08	0.67 ± 0.07	0.65 ± 0.12	0.69 ± 0.11
LR	0.69 ± 0.09	0.62 ± 0.07	0.73 ± 0.10	0.53 ± 0.11

Table shows means and standard deviations (SD) over ten iterations of 10-fold cross-validation.

^aArea under curve from receiver operating characteristic (ROC).

TABLE IV

CLASSIFICATION ACCURACY WITH BENCHMARK DATASETS

Dataset	Controls / Positives	DSI ^a	Benchmark Maximum ^b	Benchmark Average ^c
Diabetes	500/268	0.75±0.04	0.78±0.04	0.74±0.03
Heart disease	164/139	0.81±0.06	0.85±0.01	0.79±0.06
Hepatitis	123/32	0.84±0.08	0.90±0.01	0.85±0.04

^aTable shows class counts, or means and standard deviations (SD) of classification accuracy from ten iterations of 10-fold cross-validation,

^b10-fold cross-validation,

^cor all methods beating the majority class classifier.