Knowledge-Based Trend Detection and Diagnosis

by

Ira Joseph Haimowitz

S. B. Mathematics (1988)
Massachusetts Institute of Technology

M. Phil. Computer Speech and Language Processing (1989)
Cambridge University

Submitted to the Department of
Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the
Massachusetts Institute of Technology
May 1994

Signature of Author _____
Department of Electrical Engineering and Computer Science
May 12, 1994

Certified by _____
Peter Szolovits
Professor, Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by _____
Frederic R. Morgenthaler
Chair, Department Committee on Graduate Students

Knowledge-Based Trend Detection and Diagnosis

by

Ira Joseph Haimowitz

Abstract

This thesis presents a knowledge-based approach to diagnostic process monitoring. The cornerstone of this work is the representation and detection of multivariate trends in process data. The trend representation, called a *trend template*, denotes a time-varying pattern in multiple variables common to a diagnostic population. Each pattern contains representations for landmark events and a set of phases, each temporally uncertain. The phases are represented by a partially ordered set of temporal intervals. Bound to each interval are constraints on real-valued functions of measurable parameters. The constraints are low-order polynomial regression models, with either qualitative or quantitative coefficient estimates. A computer program called *TrenDx* diagnoses trends by matching process data to a set of competing trend templates within a clinical context. The matching score of a trend template hypothesis is based on the mean absolute percentage error between the regression models and the data. TrenDx not only maintains alternate hypotheses of different trends, but also optimizes over different chronologies within each trend description. Therefore TrenDx can report both *what* the most significant trend is and *when* events and phases take place within that trend.

The thesis describes how TrenDx can be extended to complete an architecture for automated diagnostic process monitoring. A faulty trend is judged significant if over time it matches process data better than the expected trend. Significance of a faulty trend may trigger an alarm, switch the clinical context, or filter data for an intelligent display.

TrenDx has been applied to diagnosis of trends in two medical domains. The program diagnoses trends in pediatric growth from heights, weights, bone ages, and sexual staging data. TrenDx also detects trends in intensive care unit patients from hemodynamic and respiratory data. The techniques of TrenDx are intended as general purpose, and may be applicable to other diagnostic monitoring applications such as industrial process control, telecommunications, economics, and finance.

Thesis supervisor: Peter Szolovits

Title: Professor, Electrical Engineering and Computer Science

Acknowledgments

My path to a doctoral thesis was a long and arduous one. I have many to thank for guiding me down this path and helping me enjoy the view as I journeyed. Below is but a partial list.

My advisor, Professor Peter Szolovits, has long supplied thoughtful, valuable advice on this thesis. For eight years he has been a role model during my initiation into the world of scientific research.

Dr. Isaac Kohane has been a constant source of advice on both temporal reasoning and in pediatric endocrinology. He has dedicated long hours to writing papers and incorporating the Temporal Utility Package into TrenDx.

Dr. Howard Shrobe read and listened to several accounts of this work, and helped me place this thesis within broader research contexts.

Phillip Phuc Le has offered computer programming support, assistance in all TrenDx experiments, and excellent feedback on germinating ideas.

The Division of Endocrinology at Boston Children's Hospital generously invited me to observe their growth clinic and use patient data. They encouraged me regarding the practical potential of my work. Dr. John F. Crigler Jr., Dr. Samir Najjar, and Dr. Joseph Majzoub diagnosed growth cases for one of the experiments. Khuram Faizan and Nadya Adjanee prepared a database of abnormal growth patients which I used in some experiments.

Dr. Norman Sherry kindly donated some of his old pediatric growth records for training and testing TrenDx.

The Dipartimento di Informatica e Sistemistica of the Universita di Pavia, Italy, welcomed me during September, 1992, and enlightened me to different approaches to automated diagnostic monitoring. Mario Stefanelli, Ricardo Bellazzi, Marco Ramoni, and Alberto Riva were generous with their time and advice.

Members past and present at the Clinical Decision Making Group at M.I.T. have made my graduate student life continually stimulating and enjoyable. My office mates through the years; Alex Yeh, Elisha Sacks, Dennis Fogg, Kevin Zalondek, and Yeona Jang; were ever ready to discuss anything from knowledge representation to election year politics. Tze-Yun Leong has been my comrade and friend throughout this pre-doctoral odyssey.

My family has given me unbounded love and support throughout my academic career. They share with me the sense of accomplishment this thesis has brought.

Most especially, my wife Barbara Haimowitz has been a devoted companion, statistics advisor, and best friend. Her steadfast love and belief in me have made this achievement possible.

This research has been supported (in part) by National Institutes of Health grant R01 LM 04493, National Library of Medicine training grant 1T15LM07092, and by a U.S. Office of Naval Research Graduate Fellowship.

לדעת חכמה ומוסר להבין אמרי בינה׃
לקחת מוסר השכל צדק ומשפט ומישרים׃

That men may know wisdom and instruction,
understand words of insight,
receive instruction in wise dealing, righteousness,
justice, and equity.

Proverbs Chapter 1, verses 2-3.

Contents

1 Introduction 17

1.1 Trend Detection is Important for Diagnosis 18

1.2 Diagnostic Process Monitoring 19

1.3 Aims of this Research 21

1.3.1 . Monitoring Sensitive to Clinical Condition and Time Period 21

1.3.2 .Multivariate Trends 21

1.3.3 .Reporting of Key Temporal Events 21

1.3.4 .Use in Domains with Scant Training Data 22

1.3.5 .Monitoring Sparse and Dense Data 22

1.3.6 .Monitoring Where Pathophysiology is Poorly Understood 22

1.3.7 .Compatibility with Workflow 23

1.4 TrenDx: Knowledge-Based Trend Detection and Diagnosis 23

1.5 Guide to This Thesis 24

2 Application Domains 25

2.1 Pediatric Growth Monitoring 25

2.1.1 .Diagnosis as Trend Detection 28

2.1.2 .Special Properties of The Growth Chart Domain 28

2.1.3 .An Example Case 28

2.1.4 .Clinically Significant Patterns 29

2.2 Intensive Care Unit Monitoring 31

2.2.1 .Example of Context-Sensitive ICU Monitoring 31

2.2.2 .Goals 33

2.3 Other Potential Application Domains 33

3 Trend Templates 35

3.1 Desiderata for Trend Representation35

3.1.1 .Requirements on Expressiveness 35

3.1.2 .Requirements for General Purpose Use 36

3.2 Core Representation Plus Flexible Framework 37

3.3 Model of Measured Parameters and Data 37

3.4 Example Trend Template 38
3.5 Language Definition 39
3.5.1. Temporal Constraints 39
3.5.2. Representing Temporal Distances 40
3.5.3. Value Constraints 41
3.5.4. Restricted Value Constraint Language 43
3.6 Expressiveness of Trend Templates 44
3.6.1. Growth Trend Template Combines Measurements 45
3.6.2. Trends Indexed by Current Time 46
3.6.3. Trends Indexed by a Trend Interpretation 46
3.6.4. Hierarchical Trend Templates 47
3.6.5. Limitations of Trend Templates 47
3.7 Diagnosis with Trend Templates 48
3.7.1. Semantics and Complexity 49
3.7.2. Cascaded Theory of Diagnosis 49

4 TrenDx 51
4.1 Instantiating Monitor Sets for a Process 51
4.1.1. Temporal Anchoring 51
4.1.2. Temporal Contexts 53
4.2 Structure of a TrenDx Hypothesis 54
4.3 Processing a Data Cluster 54
4.3.1. Example of TrenDx Processing 56
4.3.2. Space of Alternate Chronologies 58
4.3.3. Computing Consistent Temporal Contexts 59
4.4 Scoring a Hypothesis 60
4.4.1. Matching Data to Value Constraints 60
4.4.2. Combining Scores for a Hypothesis 61
4.4.3. Distribution Assumptions on $\{Y_t\}$ 62
4.4.4. Nearest Satisfier Value Constraint Model 63
4.4.5. Importance of Removing Outliers 65
4.4.6. Discriminatory Power of Low-Order Polynomials 65
4.5 Pruning Hypotheses by Beam Search 66
4.5.1. Larger Beam Means Sounder Reasoning 66
4.5.2. Prediction Granularity: Temporal Soundness vs. Efficiency 68
4.6 TrenDx Output 68
4.7 Computational Complexity of TrenDx 69
4.8 Optimizations of Temporal Reasoning for Monitoring 70
4.8.1. Reducing Queries to the Temporal Reasoner 71
4.8.2. Reducing Time Spent Per Query 71

5 TrenDx Results on Clinical Data 73

5.1 Pediatric Growth 73
5.1.1. Trend Templates 74
5.1.2. Method 76
5.1.3. Results 78
5.1.4. Discussion 82
5.1.5. How to Improve Monitoring Performance 82
5.2 Intensive Care Unit 83
5.2.1. Trend Templates 83
5.2.2. Method 84
5.2.3. Results 85
5.3 Discussion 88

6 Completing the Monitoring System 91

6.1 Detecting When a Trend is Significant 91
6.1.1. Thresholds 92
6.1.2. Accumulators 93
6.1.3. TrenDx 94
6.2 Generating Alarms 95
6.3 Data Visualization 95
6.4 Triggering Alternate Monitor Sets 97
6.5 Monitor Set Guidelines 97
6.5.1. Basic Principles 97
6.5.2. Rules 98
6.5.3. Critiquing Programs 100
6.6 Learning Trend Template Time Parameters 100
6.7 Monitoring in a Large Clinical Setting 102

7 Alternative Representations and Algorithms 103

7.1 Constraint-Based TrenDx 103
7.1.1. Value Constraints 103
7.1.2. Trigger Sets 104
7.1.3. Expressiveness of Representation 105
7.1.4. Constraint-Based Matching 106
7.1.5. Exploratory Clinical Trial 107
7.1.5.1Methods 107
7.1.5.2Results 108

7.1.6. Comparing Regression-Based and Constraint-Based TrenDx 109
7.2 Other Chronology Search Strategies 109
7.2.1. Parallel Hypothesis Evaluation 110
7.2.2. Persistence Heuristic 110
7.2.3. Repeated Iteration of Chronologies 111

8 Related Work 113

8.1 System Identification 113
8.1.1. Regression Analysis 114
8.1.2. Time-Series Analysis 115
8.1.3. Hidden Markov Models 116
8.1.4. Neural Networks 117
8.1.5. Shortcomings 118
8.2 Statistical Process Control 118
8.3 Representations of Temporal Patterns 120
8.3.1. Process Histories 120
8.3.1.1 Qualitative Histories 120
8.3.1.2 Steady State Representation 121
8.3.2. Predicates on Parameter Values Over Time 121
8.3.3. Temporal Uncertainty of Boolean Propositions 123
8.3.4. Trends as Fuzzy Membership Functions 124
8.4 Knowledge-Based Monitoring Algorithms 126
8.4.1. Model-Based Process Monitoring 126
8.4.2. State-Based Process Monitoring 127
8.4.3. Temporal Abstraction of Data to Qualitative States 127
8.4.4. Temporal Truth Maintenance 129

9 Conclusions 131

9.1 Summary 131
9.1.1. Combining Models in Two Axes Can Discriminate Trends 131
9.1.2. Framework for Diagnostic Monitoring 131
9.1.3. Flexible Representation and Pattern Matcher 132
9.1.4. Broad Applicability 132
9.2 When TrenDx is Applicable 132
9.3 Future Work 133
9.3.1. Large-Scale Clinical Trial 133
9.3.1.1 Issues in Evaluating Monitoring Systems 134
9.3.1.2 Proposed Method 134
9.3.2. Other Application Domains 136
9.3.3. Toolkit for Trend Detection 136
9.3.4. Data Uncertainty 136
9.3.5. Recommending Tests 137

9.3.6. Multiple Disorders 137
9.3.7. Integrating Text and Graphics 138
9.3.8. Subjective Trend Evaluation 139
9.4 On Modeling Natural Phenomena 139

Appendix A Pediatric Growth Monitor Sets 141
A.1 Boy Growth Monitor Set 141
A.2 Girl Growth Monitor Set 148

Appendix B Intensive Care Unit Monitor Sets 157
B.1 Blood Pressure Initial Classification Monitor Set 157
B.2 Handbagging Monitor Set 158

Appendix C Discriminatory Power of Value Constraints 161
C.1 Experimental Method 161
C.2 Results of Initial Experiment 162
C.3 Result of Increasing Sensor Noise 163
C.4 Result of Decreasing Net Change Over Interval 164
C.5 Discussion 165

References 167

Subject Index 183

Figures

Figure 1	Decreasing trend of parameter within the normal range.	19
Figure 2	Male patient with constitutional delay of growth.	26
Figure 3	Height and weight points of the growth patient interpreted as Z-scores.	29
Figure 4	One hour of four signals plotted from an intensive care unit patient.	32
Figure 5	Trend Template for a hemodynamic fault during oxygen handbagging.	38
Figure 6	Low-order polynomial regression models in trend template value constraints.	44
Figure 7	Portion of male average normal growth trend template.	45
Figure 8	Trend template for a recent decreasing blood pressure.	46
Figure 9	Trend template for adequate baroreceptor response to falling blood pressure.	47
Figure 10	Monitor set for initial classification of blood pressure.	48
Figure 11	Anchoring a trend template landmark to the patient history.	52
Figure 12	The temporal context hierarchy of TrenDx.	53
Figure 13	Portion of the average normal growth trend template.	56
Figure 14	Alternate hypotheses of the average normal growth trend template.	57
Figure 15	Continued matching to two normal growth hypotheses.	57
Figure 16	$O(T^2)$ placements of T data clusters in a maximally uncertain trend template interval	58
Figure 17	When a value constraint's qualitative parameter estimate fails, TrenDx matches to the nearest constraint satisfier	64
Figure 18	Outlying data may drastically alter the OLS quadratic fit.	65
Figure 19	Branching and pruning of TrenDx chronologies for a single trend template.	67
Figure 20	TrenDx with a small beam width may be susceptible to noisy signals.	67
Figure 21	TrenDx output. One fault template continually matches better than normal template.	69
Figure 22	Trend template for male average normal growth.	74
Figure 23	Trend template for male constitutional delay.	76
Figure 24	Growth data of constitutional delay patient.	78
Figure 25	TrenDx results on constitutional delay patient.	79
Figure 26	Growth patient from general pediatrician, diagnosed normal.	80
Figure 27	TrenDx results on general pediatrics patient	81
Figure 28	Monitor set for oxygen handbagging.	84
Figure 29	TrenDx results on ICU patient data. beam size = 4; grain = 0.	85
Figure 30	TrenDx comparative results on ICU data, beam sizes 1, 2, and 4.	86

Figure 31 TrenDx reports of phase transitions in best scoring hemodynamic fault hypothesis with beam size = 4. 87

Figure 32 TrenDx results with prediction grain of 1 minute and beam size of 4. 88

Figure 33 TrenDx results on ICU patient data. 91

Figure 34 Proposed monitor set for onset of a fault trend. 94

Figure 35 Intelligent data display based on significance of a fault trend. 96

Figure 36 A monitor set resulting in best hypotheses that throw away data. 99

Figure 37 Portion of a constraint-based trend template for male average normal growth. .. 104

Figure 38 Trigger sets on a value constraint in a constraint-based trend template. 105

Figure 39 Trend template for AST pattern in hepatic necrosis. 105

Figure 40 Constraint-based TrenDx prunes hypotheses where data do not all satisfy the value constraints. 106

Figure 41 Use of a persistence heuristic to limit branching of chronologies 111

Figure 42 Normal and abnormal tank filling trends for neural network learning. 117

Figure 43 An X-bar control chart. 119

Figure 44 Part of a disorder context in the Skeletal Dysplasia Diagnostician 122

Figure 45 Fuzzy membership function for "adequate oxygenation." Reproduced from [Steimann 1994] with permission of author. 125

Figure 46 Temporal abstraction. 128

Figure 47 TCS process for TrenDx chain of reasoning about a monitor set. 130

Figure 48 Subject test form for large-scale growth experiment. 135

Figure 49 Portion of possible future TrenDx trend alarm. 139

Figure 50 Distributions from which random data were generated. 161

Tables

TABLE 1 TrenDx matching results on tertiary care patients. 79

TABLE 2 Updated TrenDx results incorporating general pediatric patients. 81

TABLE 3 Matching low-order polynomial models to data valued within [40, 76]. White noise is Normal(0,1). 163

TABLE 4 Matching similar data as above, with white noise as Normal(0, 3), to the three value constraint models. 164

TABLE 5 Matching similar data as in Table 3; data valued within [50, 59]; with white noise as Normal(0, 1). 164

1 *Introduction*

In many domains experts can examine a series of time-ordered data generated by some process and judge the health of that process. An economist inspects the last year of employment, inflation, and other economic data and describes the economy as recessive or recovering. A coach observes an athlete's statistics for the past several games and evaluates that player as "hot" or "in a slump." A pediatric endocrinologist examines heights and weights over a child's lifetime and can judge if he or she is growing normally.

These experts have knowledge that lets them link patterns in multiple measured parameters[1] to diagnoses of the process. In this thesis we shall say that the experts know *trends:*

- A *trend is a* clinically significant pattern in a sequence of time-ordered data.

Trends may denote different types of patterns. The trend "rapidly increasing temperature" is a *symptom.* The trend "poor growth due to undernutrition" is a class of *disorders.* The trend "stable heart rate" may indicate a *response to therapy.* Although these classifications use medical terms, they apply to other processes that may be diagnosed or treated, including an economy, an industrial process, or an athlete's performance.

One way for a computer program to detect trends as well as domain experts do is for the program to "know" what patterns are important. The thesis of this research is that a computer program with knowledge of time-varying constraints on measured data can be used for automated trend detection. Given the same process data as an expert in some application domain, the program may detect trends comparably to the expert.

1. Unfortunately, the term *parameter* has two widely used meanings, both relevant to this work. One can speak of the "parameter" of a mathematical model, as in the coefficients of a polynomial. One can also speak of a measurable "parameter" of a process, as in height or temperature. The text will distinguish these meanings with the surrounding context and clarifying synonyms.

This research aims to prove this thesis with a representation called *a trend template*, denoting a time-varying pattern in multiple variables common to a diagnostic population. Each pattern contains representations for landmark events and a set of phases, each temporally uncertain. The phases are represented by a partially ordered set of temporal intervals. Bound to each interval are constraints on real-valued functions of measurable parameters. A trend diagnosis program called *TrenDx* interprets data of a process by matching them to constraints in trend templates. The program maintains alternate hypotheses not only of different trends, but also of different ways the process may have varied over time within a trend description.

1.1 Trend Detection is Important for Diagnosis

In diagnostic knowledge based systems, programs are written that can reason abductively from symptoms (or "findings") in a patient to the disorders that cause them. However, the vast majority of work in this field treats symptoms as fixed in time. These may take boolean forms, as in "chest pain = true" or one of a series of qualitative categories for a measurable parameter, as in "serum sodium = low."

Such a stationary representation of findings is insufficient for a process (such as a medical patient) being monitored over any period of time. An expert monitoring a process has notions of how the measured parameters should vary over time under the current hypothesis.When the measurements vary from the expected, that expert may consider an alternative diagnosis.

One way for a computer program to behave similarly is for it to represent the expected *trend*: how the relevant data should change over time. Other researchers have noted the discriminatory power of adding temporal information to diagnostic classes. [Kohane 1987, page 6] illustrates through examples in liver disease diagnosis that "the ability to use a large variety of temporal information permits a drastic pruning of the problem space - the number of diagnoses to be considered." [Szolovits and Pauker 1976] discovered that adding the coarse temporal descriptions "distant-past," "recent-past," "now," and "future" to symptoms entered into the Present Illness Program allowed more accurate diagnosis of acute versus chronic diseases. This research aims to apply this insight to diagnosing trends from primary data.

Given the need to represent data variation over time, the question then arises of how the data should be interpreted. [Lundbye-Christensen, Winkel et al. 1991] have noted problems of merely using reference intervals to classify each time-stamped datum as low, normal, or high. If a population interval is used, the assumption is that the patient is always from the same population, and if the interval is inferred from previous data, the assumption is that the patient's condition has

not changed over time. The authors call for time-varying descriptions of data patterns for different patient populations.

Furthermore, merely checking laboratory values against a reference interval can lead to ignoring a trend where the parameter is markedly decreasing, increasing, or periodically fluctuating within that range. Some general pediatricians make this error in monitoring growth charts. A height centile that steadily decreases from 75 percent to 25 percent over the last two years is still within a range of "normal" centiles. However, that decrease may strongly indicate either an endocrinological or nutritional disorder. This general pattern is illustrated in the figure below.

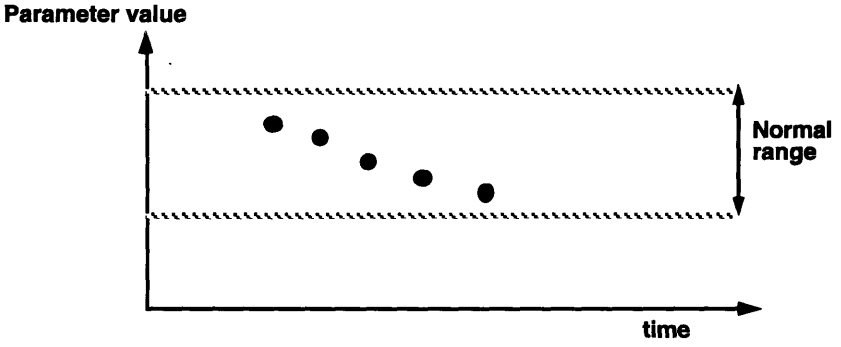


Figure 1 Decreasing trend of parameter within the normal range.

In this diagram the data values all are within the pre-established normal range, yet together they can indicate a trend such as "steadily decreasing within normal range." A computer program can detect such specific trends *only if they are represented explicitly.*

1.2 Diagnostic Process Monitoring

This thesis contributes to the discussion of designing and implementing automated systems for *diagnostic process monitoring.* In a typical diagnostic process monitoring system, a human expert (in many domains termed an *operator*) judges the state of a process from the values of measurable parameters of that process. The expert has in mind at any one time a set of hypotheses of how the process may be behaving. If there is no known fault[2] in the process and parameter values are as expected one hypothesis is probably "normal operation." Even with a known disorder D, if data are as expected then "normal operation for disorder D" should be a hypothesis. When data are not as expected for the current process state, the expert

2. In this generic description of process monitoring the terms "fault," and "disorder" are interchangeable. Typically "fault" is used in factory or power plant monitoring while "disorder" is used in medical diagnosis. In medicine the "process" is a patient.

may have to reconsider or even eliminate current hypotheses in favor of other faults.

Diagnostic process monitoring can be described as a cycle of several steps that repeats every time new data are gathered. Say at time t_0 the monitor maintains a set of hypotheses H_0 about the process. New data arrive at time t_1, later than t_0. Then a typical monitor[3] will undergo these steps:

1. The monitor interprets parameter values into other variables that offer more diagnostic information. These values may be abstractions of the data into qualitative states such as "low," or "increasing." They may be quantitative functions of the parameter values at t_1 and possibly also earlier values.
2. The monitor predicts for hypotheses in H_0 what variable values are expected at t_1. These predictions are usually constraints encoded in, or generated from, the hypotheses' representations.
3. For each hypothesis in H_0 the monitor matches the interpreted values in step one to the predictions of step two. The matching may involve reasoning about uncertainty in either the predictions or in the variable values.
4. Based on the success of the match in step three, the monitor updates the list of plausible hypotheses to a new set H_1. H_1 includes those hypotheses in H_0 that sufficiently matched the data at t_1, minus those that have been pruned because they did not sufficiently match. H_1 also includes hypotheses that have been triggered based on the match.
5. The monitor ranks the elements of H_1 according to how well they match the patient's data at time t_1.
6. If the highly ranked hypotheses in H_1 warrant immediate attention, the monitor acts externally. In *open-loop* monitoring the monitor notifies relevant human experts with an alarm. In *closed-loop* monitoring the monitor acts directly upon the monitored process, perhaps by adjusting an input or output control.
7. In either open or closed loop monitoring, the monitor may give an explanation of why its conclusion was reached. The explanation may include particular segments of the measured data that support its conclusion.

3. Some monitoring architectures may bypass or re-order one or more of these steps. As we shall see TrenDx interprets the values (step one) only during hypothesis matching (step three). Other systems including HELP [Warner 1979] send alarms (step six) based on data interpretation (step 1).

1.3 Aims of this Research

1.3.1 Monitoring Sensitive to Clinical Condition and Time Period

A main goal of this work is to establish a framework of context sensitive monitoring, in which measurements are interpreted in the appropriate clinical settings and during the time periods when monitoring those measurements provide useful information. To illustrate clinical context, in a patient whose height, weight, and sexual development indicate normal growth, there is no apparent need to monitor levels of serum growth hormone or thyroid hormone because these expensive measurements are only indicated when the other data suggest a disorder. Similarly, an acutely rising heart rate which in isolation may be worrisome, is not as problematic during intense exercise.

To illustrate how time periods affect the monitoring of various signals, during December a department store manager typically expects a short term rise in net receipts. In other months the manager would find this increase a very pleasant surprise. In order to achieve temporal context, diagnostic monitors must embed constraints on data values within temporal constraints specifying during what times each value pattern is expected. Thus the value constraints are only applied to the monitored data where they are temporally applicable.

1.3.2 Multivariate Trends

Also necessary for achieving proper temporal context is representing trends of multiple parameters together, particularly those parameters that naturally respond to each other. If a therapy on an intensive care unit patient acutely changes that patient's blood pressure, one can expect a healthy neural reflex to cause almost immediately a change in heart rate in the opposite direction. If these two measurements are constrained together in the trend for that therapy, the change in heart rate will not be seen as abnormal, but instead as an expected response in light of the therapy. The net result of this combination of parameters can be a reduction in the number of alarms during interventions or therapies. Instead, the alarms that are sounded can reflect deviations from the overall expected multivariate pattern.

1.3.3 Reporting of Key Temporal Events

As mentioned above, explanation of monitoring conclusions may require partial descriptions of the data. These descriptions are inherently related to the temporal constraints in a trend representation. For example, a user of a hemodynamic monitoring system may want to see the increasing blood pressure data in concert with an alarm. Thus the monitoring system needs to compute the beginning (and possibly end) of the time interval over which rising blood pressure was detected.

Additionally, a monitoring system may be asked to describe the times of landmark events in a process. When this event is expected in the future, we term this *prediction.* For example, in pediatric growth diagnosis it is important to predict the onset of puberty in a patient with growth disorders, to know how aggressively to treat the patient. The same type of query may also be asked about a past landmark point, for the purpose of summarizing the history of a patient.

1.3.4 Use in Domains with Scant Training Data

This research aims to produce techniques which are generally applicable in a variety of application domains. In some domains where traditional time-series analysis and signal processing have been applied, there is a wealth of training data to learn the many parameters of the trend models. Such domains include speech processing and stock market trend analysis. However, in some domains, such as pediatric growth monitoring, large training sets of abnormal patient populations have not been compiled. In these applications, a trend detection system must infer presence of trends mainly from the process data being monitored.

1.3.5 Monitoring Sparse and Dense Data

Another need for general applicability is to design a system that may detect trends in both *dense* and *sparse* data. Note that data may be called sparse if either:

- it is sampled at a frequency below the Nyquist limit, and thus the underlying pattern cannot be completely identified, or
- the sample period is long enough so that decisions must be made well before the next sampled data.

In domains with dense or oversampled data, an automated monitor may liberally filter and smooth the data before matching the signals to a trend model. Examples are intensive care unit hemodynamic and respiratory data, sampled every fifteen to twenty seconds; and speech data, sampled in milliseconds. However, in domains characterized by sparse data, an expert may deliberate over the significance of data at a particular time slice, should that data contribute to an important local trend. Examples of domains with sparse data are pediatric growth, with new samples every few months to every year; and state and federal macroeconomics, with new samples every month to every quarter year.

1.3.6 Monitoring Where Pathophysiology is Poorly Understood

Yet another requirement for general applicability is usability in domains where the underlying physiologic process is not well understood. In such domains it is extremely difficult to model the application domain by a set of differential equations, either quantitative or qualitative. Even in application domains characterized by incomplete understanding, domain experts can still accurately diagnose trends.

Pediatric endocrinologists detect trends in growth charts although they cannot describe a pathophysiologic model for how current heights, weights, and pubertal data affect similar measurements in the future. Macroeconomists can diagnose if a national economy is recessive or recovering although they have problems accurately predicting next month's economic indicators. Given this lack of understanding, or a lack of training data, one may reasonably consider a knowledge-based approach to trend detection.

Even if a pathophysiologic model can be specified, knowledge-based representations of significant trends may still improve automated monitoring. Should a monitored process not meet the model's assumptions, model predictions can vary widely from actual data. Knowledge-based patterns of prototypical normal and faulty processes can better account for inter-process variation, yielding higher diagnostic sensitivity. An incompletely specified model may generate multiple behaviors with unimportant distinctions. Without additional knowledge, a program must match process data to all the generated behaviors. If a knowledge engineer specifies which patterns are clinically significant, the number of patterns the data is matched to is reduced, improving monitoring efficiency.

1.3.7 Compatibility with Workflow

From the description in section 1.2 one can see that an automated monitor can run *without any input from humans*; process data may be read directly from a database or from the process itself. In a hospital, the data could come from an electronic patient record or the clinical event stream. Automatic data access greatly facilitates acceptance of monitoring systems into clinical practice. In one pediatric intensive care unit a monitoring system was not accepted because "special skills are needed to use the keyboard and operate the computer. This is particularly relevant where temporary staff often have to be drafted to man the bays" [Stodley, Walker et al. 1992].

Therefore a goal of this research is to produce a program that *does not* require user input of symptoms or laboratory values. The program will process whatever data are available in the on-line patient record and generate the best diagnoses and alarms based on those data.

1.4 TrenDx: Knowledge-Based Trend Detection and Diagnosis

This thesis presents a prototype computer program called *TrenDx* for automated trend detection during diagnostic process monitoring. The program matches process data to representations called *trend templates* that define faults as multivariate patterns common to a diagnostic population. These patterns include landmark events, and a partially ordered set of temporal intervals with uncertain endpoints that represent phases of the trend. Bound to each temporal interval are value con-

straints on real-valued functions of measurable parameters. In this way trend templates embed traditional signal processing and time-series analysis techniques in the appropriate temporal and clinical context. TrenDx uses the temporal uncertainty of trend templates to optimize over alternate chronologies of assigning data to phases of the trend. The program simultaneously matches process data to competing trend templates to determine which pattern within a clinical context best describes the process behavior.

TrenDx has been applied to diagnosis of trends in two medical domains. The program can diagnose trends in pediatric growth from examining heights, weights, sexual development measures, and other data. TrenDx can also recognize trends in intensive care unit patients by examining hemodynamic and respiratory data. The techniques of TrenDx are intended as general purpose, and may be applicable to other diagnostic monitoring applications such as industrial process control and finance.

1.5 Guide to This Thesis

This thesis includes nine chapters. The next three chapters continue to lay the foundations. Chapter two introduces the application domains most thoroughly studied, pediatric growth monitoring and intensive care unit monitoring, and suggests other application areas. Chapter three defines with examples the knowledge representation for clinically significant trends called trend templates. Chapter four presents the trend detection and diagnosis program TrenDx that matches time-ordered data from a process to trend templates that compete within a clinical context.

The following three chapters show how the foundations can be applied and extended. Chapter five gives the results of applying TrenDx to patient cases in pediatric growth and intensive care unit monitoring. Chapter six discusses how to supplement TrenDx to form a more complete monitoring system. Chapter seven presents other versions of trend templates and TrenDx that share the framework of the earlier chapters, yet make different choices about particular representations or algorithms.

The final two chapters set this work within a larger research picture. Chapter eight discusses related work both in statistics and in artificial intelligence. Chapter nine concludes by discussing when TrenDx is applicable, and listing ideas for continued research.

2 Application Domains

In recent years an increasing number of hospitals have put the majority of laboratory results on-line. As a result more data are reported than clinicians can possibly examine carefully. Inpatient settings, particularly intensive care units, have a plethora of high frequency data for relatively few clinicians. Outpatient clinics have less data but an overload of patients that often prohibits clinicians from considering trends in longitudinal patient data.

Consequently there is a need in clinical medicine for automated monitoring systems that can detect and notify clinicians of trends in patient data that may otherwise go unnoticed. These automated notices can result in earlier diagnoses and possible prevention of subsequent morbidity and mortality.

The hospital information system HELP [Warner 1979] can detect abnormal events in on-line patient data and communicate them via electronic mail to that patient's clinicians. HELP can detect events such as dramatic changes of consecutive laboratory parameter values and prescriptions for potentially toxic antibiotics. Studies have shown that these automatically generated alerts reduce rates of postoperative wound infection and of inappropriate antibiotic therapy [Evans 1991].

A primary goal of this thesis is creating intelligent monitors having a similar impact on medicine. Our two application domains, pediatric growth monitoring and intensive care unit monitoring, could benefit from these monitors. These domains each have clinicians overloaded with longitudinal data. They differ in the number and frequency of measurements. These two domains were chosen in part to demonstrate broad applicability of TrenDx.

2.1 Pediatric Growth Monitoring

Monitoring the growth of a child is extremely important in assessing the child's overall health: many classes of disorders evidence themselves first in abnormal growth patterns. Therefore growth measurements are taken regularly. Infants are measured for length, weight, and head circumference every two or three months; older children have height and weight measured at least once a year. If a growth disorder is suspected, a child may be measured more frequently.

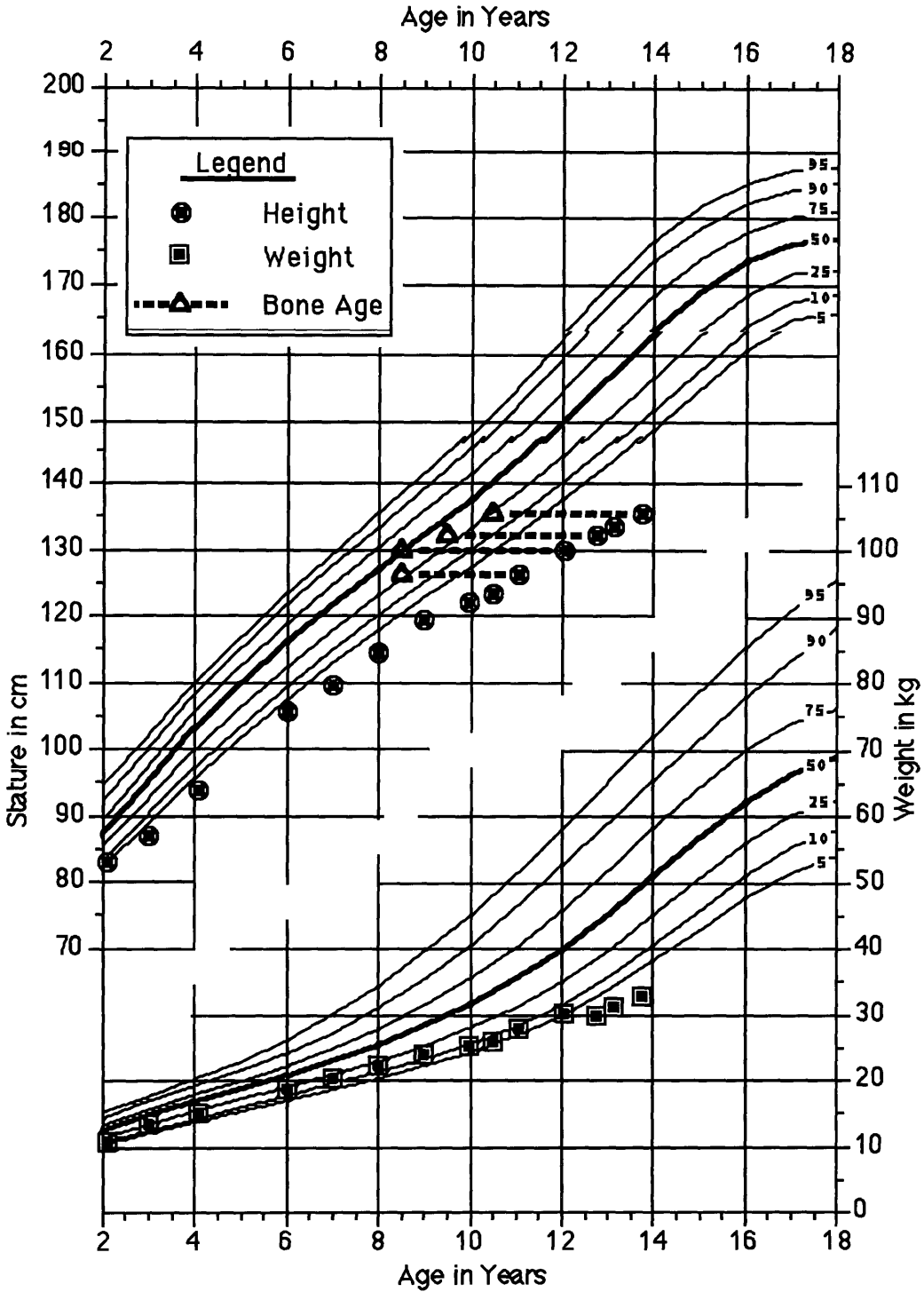


Figure 2 Male patient with constitutional delay of growth. Data courtesy of Boston Children's Hospital. Graphics courtesy of Phillip Phuc Le.

Currently the principal tool pediatricians use to monitor the growth of their patients is the growth chart. Figure 2 shows the growth in height and weight of a boy as his age progresses. A set of curves representing the mean and various centiles for heights and weights of male children studied by the National Center for Health Statistics (NCHS) [Hamil, Drizd et al. 1979] are pre-plotted on the chart. Each centile curve describes the proportion of the male children in the U.S.A. of the same age who are taller or shorter than children whose height falls on that curve. For instance, as calculated from the normal distribution, all children whose height is on the 95th centile curve are taller than 95% of male children in the U.S.A. of the same age. Paper charts typically come with lines for the 5th, 10th, 25th, 50th, 75th, 90th, and 95th centiles for height and weight.

Growth charts with such centiles exist for a variety of subpopulations, including separate charts for boys and girls in infancy (birth to 36 months) and older childhood (2 - 18 years), boys and girls with early or delayed puberty [Tanner and Davies 1985], girls with the genetic disorder Turner Syndrome [Lyon, Preece et al. 1985] and boys and girls with Down Syndrome [Cronk, Crocker et al. 1988]. When a child is suspected to have a certain disorder, expert pediatricians plot the child's height and weight points on the growth chart, if any, for the population of children with that disorder, to examine if the data support that diagnosis

Several other measurements besides height and weight are useful for pediatric growth assessment. *Bone age*, expressed in years of age, measures the development of the bones of a child and is evaluated by visual comparison of an x-ray of the left wrist with a standard atlas of normal x-rays [Greulich and Pyle 1959]. *Height age*, also expressed in years of age, is calculated as the age at which a child's height would fall on the 50th centile line of the NCHS standards for the appropriate gender. For the assessment of puberty, when growth accelerates, pediatricians monitor signs of sexual development. Many pubertal measurements are expressed as *Tanner stages*, numerical scores from 1 (no pubertal development) to 5 (full maturity). Measurements in Tanner stages include pubic hair and breast development for girls, and pubic hair, penile development, and testicular development for boys. Another important sexual development parameter is the age of menarche (first menstrual cycle) in girls.

Also important in growth assessment is the child's family history, particularly the heights and puberty onset times of parents and grandparents. Generally, a child with two short (or tall) parents will also be short (or tall), and children usually have approximately the same pubertal onset age as their parent of the same sex.

If a hormonal disorder is strongly suspected in a patient, then physicians may order a test for serum growth hormone or serum thyroid hormone. However, these tests are both expensive and time-consuming (growth hormone is measured in an inpatient test) and are only administered when the other growth measurements strongly indicate an endocrine problem.

2.1.1 Diagnosis as Trend Detection

In the growth clinic of Boston Children's Hospital, pediatric endocrinologists sometimes show each other only a patient's growth chart and ask for possible disorders. These experts can diagnose disorders from the growth chart because *some types of growth disorders show characteristic, separable trends in height and weight.* These trends are expressed verbally mostly as patterns of faithfulness of points to their centile channels. These patterns usually contain temporal relations. If a pre-pubescent child is "following the centile channel" in a population, then that child is typically growing well for that population. Children that "lose centiles" in weight while "maintaining centiles" in height show signs of undernutrition.

For more serious classes of diseases, trends in height and weight may be too ambiguous to yield a complete diagnosis. In this case the expert will request additional growth data that distinguish one disorder from another. For example, both children with growth hormone deficiency and with constitutional delay of growth (delayed puberty) will lose height and weight centiles at some age. However, those with constitutional delay have a fixed delay of bone age behind chronological age, while in growth hormone deficient children this gap increases with time if the disorder is untreated.

2.1.2 Special Properties of The Growth Chart Domain

As a process monitoring domain, pediatric growth charts are distinct from some other domains in at least two ways. First, the sampling frequency of a child's height and weight is irregular and quite slow. Therefore an automated growth monitor should not require data to be sampled regularly or frequently. Another special property of growth charts is that height data are relatively clean. Height measurements have been reported to have a coefficient of variation of 0.19 [Voss, Walker et al. 1989], and a measurement error of three millimeters when done by "skilled and careful hands" [Tanner 1990, page 7].

2.1.3 An Example Case

The growth data illustrated in Figure 2 are from a boy seen at the endocrinology clinic at Boston Children's Hospital. This patient was brought into the clinic at age 11 for consultation about possible growth disorders. His regular pediatrician had noticed a particularly sharp drop against the NCHS standards when the patient was 10.5 years old. This was of concern as children with a normal tempo of growth tend to grow on or parallel to the standard curves. The endocrinologists at Children's agreed that the boy did not appear to exhibit an average growth pattern, and that one of the likely hypotheses was constitutional delay of growth, a normal variant of growth marked by delayed puberty and bone maturity well behind the

patient's chronological age. Another, less likely hypothesis was growth hormone deficiency.

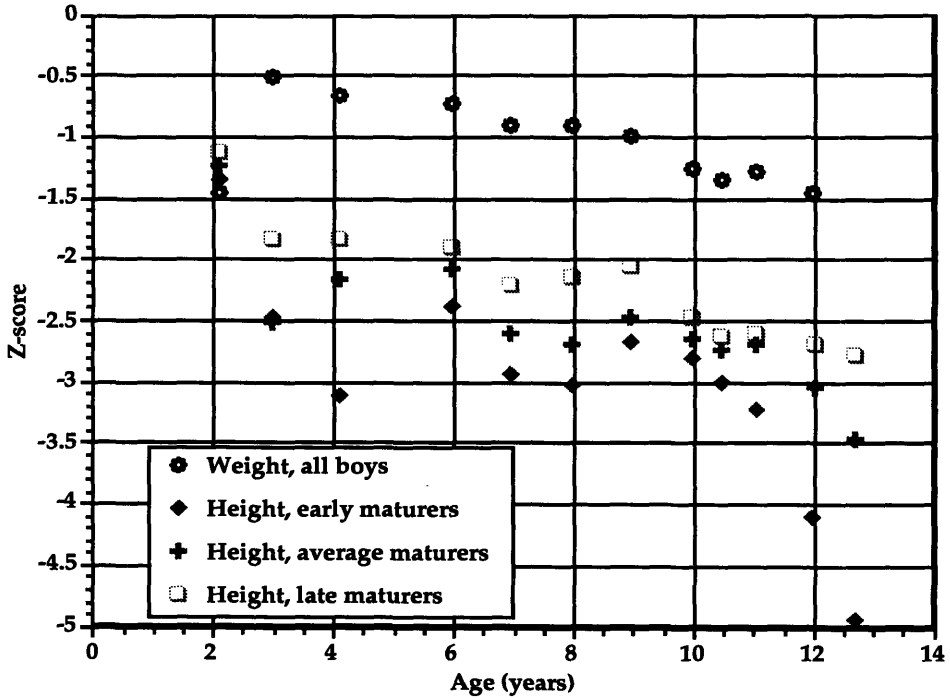


Figure 3 Height and weight points of the growth patient interpreted as standard deviations (Z-scores) with respect to different populations.

In the above figure the patient's heights and weights are plotted as *Z-scores*. A *Z-score* is the number of standard deviations from the mean that a height or weight is for a child of the same age, gender, and diagnostic population. In using the Z-score we assume height and weight are close to normally distributed per gender, population, and age. This is widely accepted for height, and is a debatable but sometimes reasonable assumption for weight. In this patient the height Z-scores with respect to the early and average maturers drop much sooner than the Z-scores with respect to the late maturers. The height Z-scores in the different populations clearly diverge after age eleven. Consequently, during this age range one can best discriminate rate of pubertal development based on height. The Z-scores for populations of different pubertal development were computed from the standards in [Tanner and Davies 1985].

2.1.4 Clinically Significant Patterns

Specialized textbooks and pediatric endocrinologists both describe in text the patterns followed by children growing normally and abnormally. The text description of male average normal growth consists in part of four constraints:

C1. From birth until age 2 - 3 years, the patient establishes his centiles for height and weight. During this time height and weight centiles vary in the same way from their original centiles.

C2. From then until the onset of puberty, the patient stays close to the same centiles in height and weight, with respect to the population of average pubertal developers.

C3. Throughout the growth of the child, bone age is approximately equal to chronological age.

C4. Puberty begins between age 10 and age 13, and is measured by when the qualitative testicular stage of the boy changes from stage 1 to stage 2.

The text description of constitutional delay of growth in boys differs only in that puberty begins at some moment between age 12 and 16, that the bone age is delayed several years, and in that the reference population is children having delayed puberty rather than puberty at the average age.

The text descriptions are characterized by uncertainty both in the *value* of relevant variables, and in the *times* over which these values vary. There is uncertainty in the time of the onset of puberty, and uncertainty of the age at which constraint C1 ends and constraint C2 begins. There is uncertainty in a child staying "close to the same centiles" and in the bone age being "approximately equal to" height age.

This uncertainty is due to experts characterizing a common pattern for a large diagnostic population. In domains where there is little knowledge of the underlying physiologic process, as in human growth, patterns like these are the primary tools experts use to diagnose disorders, and to recommend further tests.

In pediatric growth monitoring both false positive and false negative diagnoses could be reduced with more careful monitoring of heights and weights on the growth chart. False negatives include craniopharyngiomas, brain tumors that can grow to damage the pituitary gland and may later cause blindness. These tumors are often not detected soon enough because "the patient with craniopharyngioma may complain only of growth failure and may not have headaches or visual impairment" [Kaplan 1990, page 37]. A leading pediatric endocrinology text recommends that "any child with unexplained growth retardation, obesity, and delayed skeletal maturation should be screened for the presence of craniopharyngioma" [ibid, page 37]. False positives include healthy children with delayed puberty or short parents that are sent to specialists for consideration of growth hormone treatment even though there is no endocrinological disorder. The controversy over whether such patients merit growth hormone treatment has made the popular press [Werth 1991].

General pediatricians suffer from *data overload* in monitoring growth of their patients. The physician generally has fewer than ten minutes with a patient during a checkup. This is not enough time to carefully analyze the entire sequence of

time-ordered heights and weights. I have observed in pediatric clinics paper patient records with several growth charts, each with a single visit plotted.

2.2 Intensive Care Unit Monitoring

In intensive care unit (ICU) monitoring there is also data overload: eight or more patients are in an ICU, and each patient is monitored with dozens of hemodynamic and respiratory variables sampled several times per minute [Berk and Sampliner 1990, chapters 6 and 7]. It is impossible for nurses to steadily monitor even minutes worth of continuous data from an individual patient. In pediatric growth monitoring new data (i.e. heights and weights) arrive several months or a year apart and each datum takes on diagnostic importance. In the ICU filtering techniques can be applied more liberally.

Our goal in ICU monitoring is developing context-sensitive monitors whose use will significantly reduce the high false positive rates typically produced with simple threshold monitors. TrenDx will monitor the adequacy or failure of external interventions on ICU patients, and normality or abnormality of physiological mechanisms in these patients. Each intervention or mechanism consists of multivariate patterns over several phases. An automated monitor must apply specific filters and value constraints appropriate for each phase of the intervention or mechanism.

2.2.1 Example of Context-Sensitive ICU Monitoring

In Figure 4 below are one hour of ICU data from an 8 month old girl with adult respiratory distress syndrome [Nichols, McCloskey et al. 1992]. Four signals are plotted from 12:00 a.m. to 1:00 a.m.: heart rate taken from the electrocardiogram (ECG), mean arterial blood pressure, oxygen saturation, and fraction of inspired oxygen (FIO_2). Data were compressed by reporting values only upon changes. Usually, the patient received oxygen via the ventilator, FIO_2 at 50%. Approximately once every two hours, the patient was ventilated by the nurse squeezing a hand bag filled with 100% oxygen, so that a bronchodilator could be delivered in aerosol form. One such handbagging session was from 12:22 a.m. until 12:31 a.m. As illustrated in the figure, the change to hand-bagging was marked by an immediate rise of FIO_2 from 50 to 100, remaining at 100 during hand bagging. Within a minute after hand-bagging began, O_2 saturation rose sharply to 100%. These two changes are expected in such a handbagging session. During such hand-bagging it is preferable that the patient's hemodynamics remain stable. However, in this patient mean arterial blood pressure dropped from about 12:26 a.m. to 12:31 a.m, and ECG-measured heart rate rose steadily from approximately 12:27 until 12:30 a.m. These two changes are not usually expected. This pattern in these four parameters occurred during each of the six handbagging sessions for this patient over a twelve-hour period.

One possible explanation for this hemodynamic fault is that the oxygen handbagging increased pressure in the chest cavity. This could have depressed the patient's vena cava and compromised her venous return to the heart, resulting in the falling blood pressure. The heart rate increase may have been a normal baroreceptor reflex to the falling heart rate. Whatever the explanation, this hemodynamic fault is worthy of a clinician's attention.

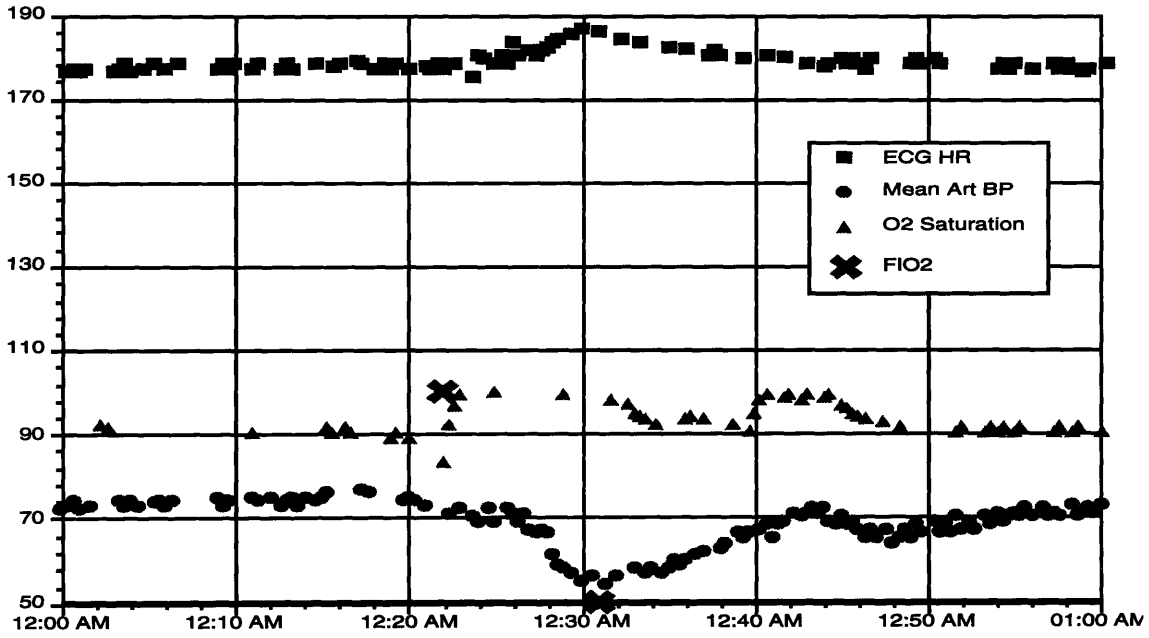


Figure 4 One hour of four signals plotted from an intensive care unit patient.

A set of threshold monitors, each checking for single parameter deviation from pre-set ranges, would send as many as four alarms during the hand-bagging period. Two of the four may be viewed as false positives. In the context of handbagging, the FIO_2 and O_2 saturation changes are normal. The other two changes may be presented as a single alarm for a hemodynamic fault during oxygen handbagging. During some of the handbagging sessions with this patient the nurse did in fact suspend automatic alarms. A monitoring system that monitors multivariate trends and is aware of the handbagging session can reduce the number of alarms from four to one.

A reliable reduction of redundant and false positive alarms is definitely needed in an intensive care unit. The reduction could clearly improve monitoring practice. When nurses shut off or ignore automated alarms they risk missing alerts to serious conditions. Monitoring of industrial power plants and industrial processes could also benefit from context-sensitive alarms with higher specificity.

2.2.2 Goals

Our goals include developing intelligent monitors that, in a handbagging scenario similar to this example, will:

- Detect when the nurse has begun handbagging the patient with 100% oxygen.
- Closely monitor relevant variables once handbagging has been detected.
- Accept the FIO_2 and O_2 saturation changes as normal.
- Alert the nurse as mean arterial pressure decreases and rising heart rate increases of a hemodynamic fault.

Such a set of monitors will demonstrate the potential of TrenDx for reducing redundant and false positive alarms in the intensive care unit.

2.3 Other Potential Application Domains

In other monitoring domains experts can recognize multivariate, temporally uncertain trends suggest a class of diagnoses. The important trends and diagnoses usually change based on clinical context. Here we briefly suggest a few domains.

In *macroeconomics*, researchers have investigated detailed time series models for accurately predicting future levels of single variables, including gross national product [Stock and Watson 1988]. However, multivariate diagnostic monitoring is prevalent. For instance, a rise in interest rates by the United States Federal Reserve Board during an economic recovery[1] leads to close monitoring for combined trends in new housing starts, inflation, and unemployment rate. A related, higher frequency monitoring domain is *financial trading*. Here too, researchers have focused on accurate prediction of single variables (a good introduction is in [Hutchinson 1994, chapter 1]). Diagnostic monitoring may be useful in judging if particular stocks react as expected to a merger announcement, if New York trading patterns closely follow trends in Tokyo and London, or if multiple stock prices fall shortly after the aforementioned interest rate increase.

Knowledge-based diagnostic monitoring of multivariate trends may be applied in domains having a schemata of process connections. Both *industrial process monitoring* and *power plant monitoring* are two examples. As mentioned in the previous section, these domains can benefit from automated alarms with low false positive rates. In chapter 8 we describe a few knowledge-based approaches to monitoring in these domains; most have focused on representing univariate trends. Monitoring for multivariate patterns following well-understood faults may

1. Such interest rate raises were a major American concern in early 1994.

improve monitoring performance. For example, leaky tanks or valves may be represented via the expected time-varying changes of tank levels, temperatures and pressures. Another domain with a detailed process schemata is *telecommunication monitoring*. Using this schemata and experience of experts, knowledge engineers may be able to encode expected responses over time to a broken cable or failed generator.

In subsequent chapters we define the trend template knowledge representation for clinically significant trends. We also define the algorithm TrenDx that matches process data to trend templates. Throughout these chapters we illustrate our techniques with examples from pediatric growth and intensive care unit monitoring.

3 *Trend Templates*

A *trend template* represents a pattern of multivariate data shared by a diagnostic population in a particular clinical context. The representation combines constraints in both value and time. The value constraints limit functions of measured parameters to fit statistical models expressed as functions of time. The time constraints limit the extent of the value constraints to the appropriate time intervals, each having uncertain endpoints and duration. Together these constraints provide more discriminatory power between different diagnoses than either could provide alone. Trend templates are similar to patterns verbally expressed by experts in some monitoring domains.

Throughout this chapter we will illustrate trend templates pictorially. The pictures illustrate all of the essential temporal and value information in a trend template. However, the actual representation language for trend templates is implemented in the Common Lisp Object System (CLOS). Appendices A and B give the actual encoded trend templates in pediatric growth and intensive care unit monitoring.

3.1 Desiderata for Trend Representation

This work aims to produce a general purpose knowledge representation for use in detecting multivariate trends. Here we list some of the requirements in designing such a representation.

3.1.1 Requirements on Expressiveness

Our representation must be expressive enough to encode trends for alternative diagnostic populations. Below are some corresponding restrictions.

- A represented trend must allow variation among processes with the same diagnosis. A monitoring program matching data to these trends can have a high *sensitivity.*
- The representation language must let a modeler highlight the differences in trends that compete within a clinical context. A

monitoring program matching to these trends could then have a high *specificity.*

- As discussed in section 1.3.1, the trends should include the application of constraints on measurements during the appropriate time periods. Thus value constraints need to be embedded within temporal constraints.
- To allow the aforementioned variation within a diagnostic category, the representation must allow uncertainty in both the temporal and the value constraints. Consider our description of the first phase of normal growth in section 2.1.4. The interval lasts from birth until between 2 and 3 years. During this interval there should be a "similar" variation in height and weight centiles. The uncertainty along two axes allows for inter-patient variation within this population.

3.1.2 Requirements for General Purpose Use

For a trend representation language to be of general purpose use in diagnostic monitoring, it must meet several additional requirements.

- For facilitating both knowledge engineering and explanation, the representation should have its linguistic primitives at the knowledge level [Newell 1981] of a process. This level includes as primitives measurements, data, landmark events, and phases.
- While in some domains the courses of trends have been studied in detail, in other domains they must be discovered from training data. Thus the parameters in the trend models must be readily estimable from data.
- To be useful in many application domains, the representation language should be compatible with the natural expression of trends in many domains. Certainly intensive care unit physicians, pediatricians, stock market analysts, and power plant operators describe their meaningful trends using different time scales and different detail of measurement changes. Yet there is much commonality. Process phases, stability, rates of change, and concurrent variation of multiple signals are pervasive notions in trend detection. Our goal is to include these common components or trends, yet allow specialized representations as a particular application warrants.

3.2 Core Representation Plus Flexible Framework

The aforementioned requirements are not easily concurrently satisfied. Generally, the more expressive a representation language, the better the discriminatory power between competing descriptions. However, the richer the representation language, the more a modeler must rely on either expert experience or training data to represent a diagnostic population. If one expands the representation language by adding numerical parameters to estimate, and these parameters have no intuitive meaning, the representation may fall below the knowledge level. Given these trade-offs, the decision about linguistic expressiveness should not be uniform, but rather *application dependent.*

This work defines a trend representation called the *trend template* that includes a fixed core model of measurements, data, and time constraints, supplemented by a flexible framework of value constraints. This thesis for the most part describes a particular choice of value constraint language for trend templates: low-order polynomials with qualitative or quantitative parameter estimates. This value constraint language gives a knowledge engineer some latitude for discriminating between competing trends. The resulting trend template language supports a straightforward maximum likelihood algorithm for estimating temporal distances. In domains where large data sets of many competing trends are available, users may extend the value constraint language to include more detailed regression or time series models. A modeler should choose a value constraint language as detailed as necessary for discriminating between competing trends. In chapter 7 we shall describe other alternatives for trend template value constraint languages.

3.3 Model of Measured Parameters and Data

A parameter p is a type of measurement. Inherent to a parameter are its name, its unit of measurement, and several attributes potentially useful for trend detection. The *persistence* of a parameter is a function of a process, that when applied on a particular process gives a pair of temporal distances defining how much time into the past (t_p) and the future (t_f) that value may be soundly projected.

$$\text{Persistence}(p)\text{: Process} \rightarrow (t_p, t_f). \qquad \textbf{(EQ 1)}$$

The persistence is process specific because it may depend on, for example, the age, gender, and clinical history of a medical patient. Similarly, the *measurement error* of a parameter is also a function of the process, that when applied to a process yields the standard deviation of the error in taking the measurement on the patient.

$$\text{Error}(p)\text{: Process} \rightarrow \text{m} \qquad \textbf{(EQ 2)}$$

A datum d is an instance of applying a parameter to a process. A datum may be thought of as an ordered quadruple:

$$d = (Pr, p, t, v) \quad \text{(EQ 3)}$$

where Pr is the process the parameter p is measured on, t is the calendar time when the measurement is made, and v is the value. We assume data are time stamped.

We can use properties of the parameter p to extend the estimates for the value of d. If Persistence(p) applied to Pr at time t is (t_p, t_f) then we may estimate that p has value v for Pr from time (t - t_p) to (t + t_f). By incorporating the measurement error of p we can make a broader estimate. If Error(p) applied to Pr at time t is m, and we assume that measurements of p follow a normal distribution, then our 95% confidence interval of the value of p for Pr is (v - 1.96m, v + 1.96m) during any time from (t -t_p) to (t + t_f). This straightforward model of persistence has proven sufficient for this project. More thorough work on projecting a single measurement to obtain probabilistic estimates of future values can be found in artificial intelligence [Dean and Kanazawa 1989] and Kalman filtering [Chui and Chen 1991].

3.4 Example Trend Template

Below is an example of a trend template representing a multivariate trend in four parameters during a hemodynamic fault in the clinical context of oxygen handbagging (as described in section 2.2).[1]

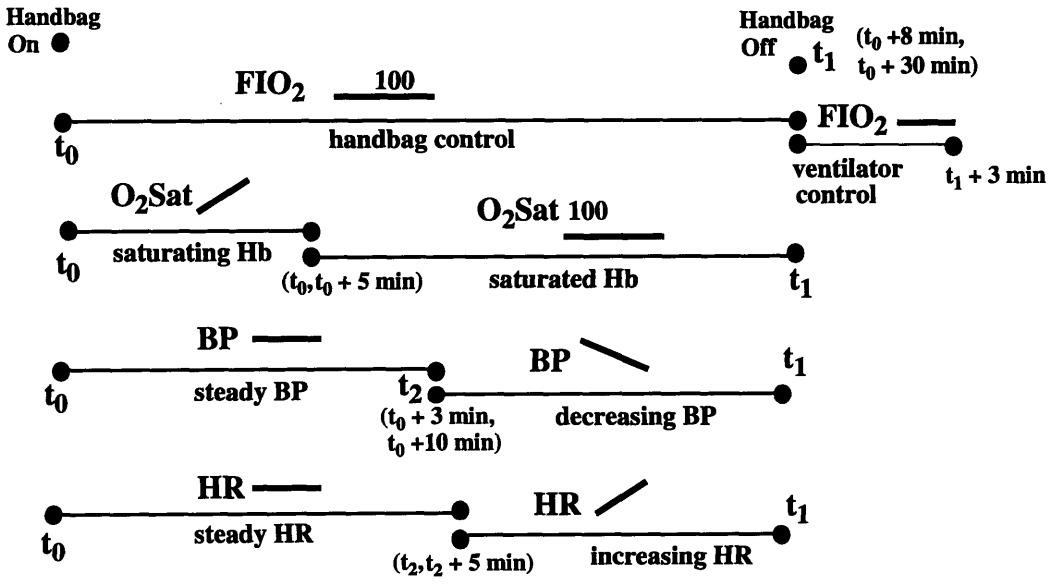


Figure 5 Trend Template for a hemodynamic fault during oxygen handbagging.

1. A patient may experience other hemodynamic fault patterns during handbagging. For example, blood pressure may rise while heart rate falls. The other faults, if worth sending alarms for, may be represented as a competing trend templates. The pattern shown is of particular interest as it *may* be caused by improper squeezing of the handbag.

This trend describes the multivariate pattern expected in nearly all patients with this hemodynamic fault. We shall refer to this example while defining the trend template language.

3.5 Language Definition

Trend templates represent uncertainty both in the *times* of changes in the measured parameters and in the *values* measured during those changes. The power of trend templates to discriminate between competing trends comes from the combination of both the temporal and value constraints.

3.5.1 Temporal Constraints

The temporal component of a trend template include *landmark points* and *intervals*. Landmark points represent significant events in the lifetime of the monitored process. Landmark points may be linked with time ranges `(min max)` expressing the minimal and maximal times between them. The trend template for the hemodynamic fault includes two landmark points: `Handbag On` and `Handbag Off`. The trend template constrains `Handbag Off` to take place 8 to 30 minutes after `Handbag On`, which is a wide enough uncertainty bound to include each handbagging session of the training patient.

Intervals represent periods or phases of the process that are significant for diagnosis or therapy. Each interval consists of a begin point and end point, both having temporal uncertainty expressed in one of the following forms:

- offsets of the form `(min max)` from a landmark point, or
- offsets of the form `(min max)` from another interval begin or end point.

Because intervals are intended to represent meaningful phases during a process, a knowledge engineer may assign names to each interval, and then encode the temporal distances on interval endpoints in terms of the interval names. In the growth domain, phases such as infancy, pre-puberty, and growth spurt are naturally represented by intervals with these names. In the above intensive care unit template, some intervals are named based on the qualitative change in a measurement. If an interval is not given a name, it may be referred to by an internal number that the program automatically generates.

The hemodynamic fault trend template consists of eight intervals. The changes in four parameters are each represented in a pair of intervals. Temporal relations between these intervals establish a pattern that is fairly specific to this particular population response. The top two intervals denote that, during handbagging, the fraction of inspired oxygen (FIO_2) remains constant at 100 percent, and that for three minutes after handbagging, FIO_2 is constant at some unspecified value.

Given the time of `Handbag On` for a particular patient, the trend detection program TrenDx can use these two intervals to estimate that patient's time of `Handbag Off`. TrenDx estimates based on when FIO_2 has changed from 100 percent. The next two intervals describe the rise and stabilization of oxygen saturation of hemoglobin. During `saturating Hb`, O_2 saturation is linear and increasing; during `saturated Hb`, O_2 saturation is constant at 100 percent. The two other pairs of intervals constrain the responses of blood pressure and heart rate. Each parameter first has a steady phase, beginning at the same time as `Handbag On`. During these phases, both parameters are constant. A second phase of decreasing BP, beginning 3 to 10 minutes after `Handbag On`, constrains BP to be linear and decreasing. This phase ends at `Handbag Off`. A second phase of increasing HR, beginning 0 to 5 minutes after the begin point of `decreasing BP` constrains HR to be linear and increasing. The temporal relations between these intervals insure that as TrenDx matches process data early in the template, the program constrains the expected match to data in the future.

3.5.2 Representing Temporal Distances

Trend templates represent time using the Temporal Utility Package (TUP) [Kohane 1987]. The features of TUP that made it useful for representing trend templates are:

1. Representation of both time points and time intervals; intervals include a begin point and an end point.
2. Representing uncertain ranges of temporal distances.
3. Reasoning about alternate temporal worlds.

Other temporal reasoners with these features are probably useful for representing trend templates and for reasoning by TrenDx.

For our two application domains we have used a second as the time unit. All statements of temporal relations between two points have the following form:

```
(create-relation p1 p2 :lb n1 :ub n2)
```

where p1 and p2 are time points, and the n1 and n2 are numbers denoting the lower and upper bounds on the numbers of seconds between p1 and p2.

For example, two time constraints on the hemodynamic fault trend template's fourth interval, `saturated Hb` are encoded as the two statements:

```
(create-relation Handbag-On (begin saturated-Hb)
            :lb 0 :ub (minutes 5))
(create-relation Handbag-off (end saturated-Hb)
            :lb 0 :ub 0)
```

where `(minutes n)` multiplies n by the number of seconds in a minute, and `begin` and `end` are functions that retrieve the begin point and end point, respectively, of an interval.

A special macroexpression in the trend template language describes two trend template intervals as *consecutive phases*. The expression

```
(consecutive-phase Int1 Int2)
```

denotes that the endpoint of the interval named Int1 is instantaneously before the begin point of the interval named Int2. This macroexpression is translated into the statement:

```
(create-relation (end Int1) (begin Int2)
            :lb *epsilon* :ub *epsilon*)
```

where *epsilon* is a fixed number describing a time period several orders of magnitude smaller than any temporal distance in any trend template. The consecutive phase intervals are used by TrenDx for efficiency of matching (see section 4.8), and are useful for critiquing the usefulness of a trend template (see section 6.2). The above trend template for hemodynamic fault contains four pairs of consecutive phases; each pair is placed together in the diagram.

Whenever an trend template interval *Int* is defined, the temporal reasoner automatically creates a relation representing that the end of *Int* is after the begin of *Int*. Therefore this relation need not be explicitly encoded in a trend template.

The knowledge engineer should at first encode all temporal distances in a trend template as wide, universally agreed upon ranges. In section 6.6 we present a maximum likelihood algorithm for estimating these temporal bounds from training data. We discuss in section 4.6 how the trend detection program TrenDx restricts this temporal uncertainty for an individual process in order to optimize the fit of the process data to this trend.

3.5.3 Value Constraints

The trend diagnosis program TrenDx generates hypotheses of how process data match a trend template. A hypothesis assigns each datum to a unique trend template interval. Thus one may speak of "the data assigned to an interval" in a hypothesis.

The value component of a trend template is a set of *value constraints,* each belonging to a unique interval in the trend template. Each value constraint is a parameterized statistical model describing how data assigned to the interval may vary. More precisely, let *hyp* be a TrenDx hypothesis consisting of a trend template *TT* and an assignment of data to the intervals of *TT*. Let *I* be an interval of *TT* and let D(*I*, *hyp*) be the data assigned to *I* in *hyp*. Each value constraint consists of two main components:

1. a function F that maps the data D(*I*, *hyp*) to a time-indexed real-valued sequence $\{Y_t\}$.
2. a linear regression model describing the pattern of $\{Y_t\}$.

The function F computes its sequence $\{Y_t\}$ primarily in one of two ways. One method is to return the same numerical attribute of each datum for each measurement. Examples are the sequence of values of all temperature data assigned to *I*, and the sequence of centiles of all height data assigned to *I*. The other method is to return all values of a multivariate function, evaluated over sets of measurements whose persistence ranges (section 3.3) overlap. An example is the sequence of blood serum pH's calculated using the Henderson-Hasselbach equation on input sequences of partial pressure of carbon dioxide (PCO_2) and serum bicarbonate concentration ($[HCO_3^-]$) measurements:

$$pH_t = 6.1 + \log \frac{[HCO_3^-]_t}{0.0301 \cdot (PCO_2)_t} \qquad \text{(EQ 4)}$$

Following such an initial calculation, the function F may process the resulting sequence yet further by filtering or smoothing the data, and removing outliers. The final results of F is the sequence $\{Y_t\}$.

The parameterized linear regression models specify that the sequence $\{Y_t\}$ fits a linear combination of fully-specified statistical models of time. This general class of models includes, but is not restricted to:

- Y_t is any polynomial function of time:

$$Y_t = \left(\sum_{i=0}^{N-1} a_i t^i \right) + \varepsilon_t, \qquad \text{(EQ 5)}$$

  where N must be specified, and the ai may be specified or unknown.

- Y_t is a linear combination of fully-specified exponential or logarithmic functions of time:

$$Y_t = \left(\sum_{i=0}^{N-1} a_i e^{b_i t} \right) + \varepsilon_t, \text{ or } Y_t = \left(\sum_{i=0}^{N-1} a_i \log (b_i t) \right) + \varepsilon_t \qquad \text{(EQ 6)}$$

  where N and the b_i must be specified, and the a_i may be specified or unknown.

- Y_t is a linear combination of fully-specified trigonometric functions of time. For example:

$$Y_t = \left(\sum_{i=0}^{N-1} a_i \sin(b_i t) \right) + \varepsilon_t, \quad \text{(EQ 7)}$$

where N and the b_i must be specified, and the a_i may be specified or unknown.

- Y_t satisfies an autoregressive model (of order k):

$$Y_t = \left(\sum_{i=1}^{N} a_i Y_{t-i} \right) + \varepsilon_t \quad \text{(EQ 8)}$$

where N must be specified, the samples Y_t must form a time series[2], and the a_i may be specified or unknown.

Each of these models includes an error term ε_t, to represent variability in Y_t not accounted for by the model.

The knowledge engineer building trend templates is encouraged to use, where appropriate, these or other linear regression models within value constraint definitions. As part of its matching process, TrenDx calculates the minimal mean squared error estimates for the unspecified parameters for each set of data assigned to an interval. Note, however, that where equation parameters are specified, the knowledge engineer must either learn these parameters from a training set of data, or be completely confident of these estimates, say from a clinical trial.

3.5.4 Restricted Value Constraint Language

The primary set of linear regression models used in this research is a subset of the general space of models described above. The models used are polynomials of degree 0, 1, and 2, with qualitative or quantitative constraints on some subset (perhaps empty) of the polynomial coefficients. A *qualitative constraint* is a member of the set {+, -}, representing that the parameter is positive or negative.[3] A *quantitative constraint* is either a single numerical value or a numerical range `[min max]` of values. In Figure 6 below are the seven qualitatively distinct elementary regression models used in value constraints. These seven models are sufficient to roughly distinguish between different types of behaviors.

Constant models with quantitative parameter constraints can be used to represent steady states. An interval of normal human temperature may constrain temperature to be constant at 37 degrees Celsius. We have already seen fully saturated

2. In order to form a time series the sequence $\{Y_t\}$ must be equally spaced in time. If not, a time series may be generated by including all the $\{Y_t\}$ plus other values found by (linear) interpolation.

3. In this research a parameter estimate of 0 is considered a quantitative estimate.

hemoglobin modeled with a constant O_2Sat of 100. A constant model without a numerical estimate represents quiescence at an unknown level.

Linear models with quantitative slope constraints can help distinguish clinically distinct trends. For example, blood pressure loss due to handbagging may have a slope in the range -1 mm Hg to -3 mm Hg per minute, whereas blood pressure loss due to internal hemorrhaging may have a slope in the range -10 mm Hg to -20 mm Hg per minute. Linear models with qualitative slopes constraints can roughly distinguish responses, as seen in the hemodynamic fault template.

Quadratic models with qualitative constraints are useful for representing trends having a sharp increase or decrease followed by a stabilization. They can better fit data showing a nonlinear response than can a linear model. When qualitative or quantitative constraints on quadratic coefficients are not derivable, a knowledge engineer may better characterize a quadratic trend with *qualitative constraints on the first and second derivatives*. The four combinations of these quadratic qualitative derivative constraints are shown in Figure 6,c.

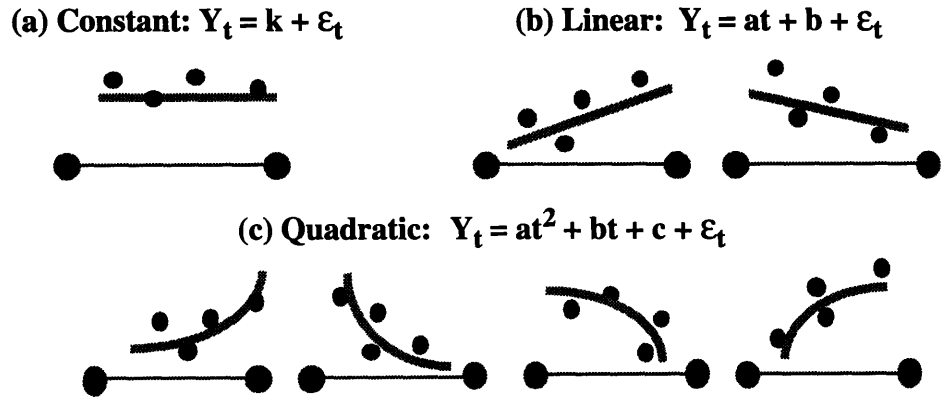


Figure 6 Low-order polynomial regression models in trend template value constraints.

3.6 Expressiveness of Trend Templates

Trend templates may express a fairly broad class of multivariate patterns. In the following sections we illustrate the utility of value constraints with more complex functions. We also show trend templates indexed not by a fixed moment in the patient history but by times that must be calculated: the current time, and the time indexed by another trend. Additionally, descriptions of trend templates in a variety of medical domains including monitoring diabetic patients and temperature during the menstrual cycle may be found in [Kohane and Haimowitz 1993].

3.6.1 Growth Trend Template Combines Measurements

In Figure 7 below is a portion of the male (average pubertal tempo) normal growth trend template, which aims to capture the natural language trends expressed in section 2.1.4. This partial template includes two landmark points: `birth`, denoting the patient's birth date, and `puberty onset`, denoting the time at which the child begins puberty. `Puberty onset` occurs 10 to 13 years after `birth`. The temporal links between landmark points establish a time scale for the a process being monitored. In the diagram, `birth` is illustrated as at time 0 and a time scale is labeled in "years." These arbitrary labels are for illustration; the trend template representation requires no "zero point" and it allows mixing of time scales (seconds, hours, minutes) within the same template. Two intervals are shown in this partial template: `establish centiles`, representing the first phase of growth where centiles are established, and `pre-puberty`, the period when a child grows within centile ranges. The two intervals are consecutive phases; the transition point between the phases is temporally uncertain at between two and three years after `birth`.

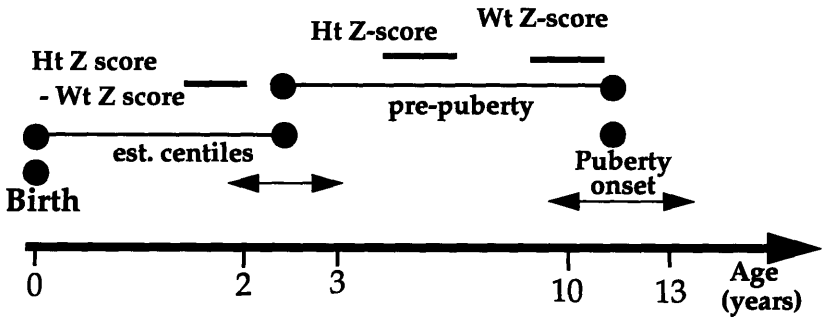


Figure 7 Portion of male average normal growth trend template.

The value constraints in this normal growth template are based on height and weight *z-scores* (see section 2.1.3), Computing the z-score of a height or weight involves comparing that datum to curves of height and weight centile standards from [Hamil, Drizd et al. 1979; Tanner and Davies 1985]. The standard curves are approximated with pre-computed eight and ninth order polynomials. In the above template, the standards used are for males developing at the average tempo of puberty. Other, competing templates use standards of males developing at early or late tempo of puberty.

The value constraints use z-scores to approximate the aforementioned natural language trends. The interval `establish centiles` models that height and weight centiles vary together by constraining that the difference between the height z-score and weight score is constant. The interval `pre-puberty` models that height and weight remain in centile channels by constraining the height z-scores to

remain constant, and also the weight z-scores to remain constant. In section 5.1 we shall describe this entire growth trend template and show its being matched to patient data.

3.6.2 Trends Indexed by Current Time

Some trends cannot be indexed by a particular point in the patient history but instead need to be continually evaluated against recent or current data. A trend template may contain a special landmark point called Now that denotes the time of the currently processed datum.

The trend template in Figure 8 below may be used to detect a short term trend of falling blood pressure over the last five minutes. The landmark point `-5 min.` is temporally constrained as exactly five minutes before Now. A pair of consecutive phases constrain blood pressure to first be constant and then linear and decreasing. The transition point between the phases is between one and five minutes before the current datum time.

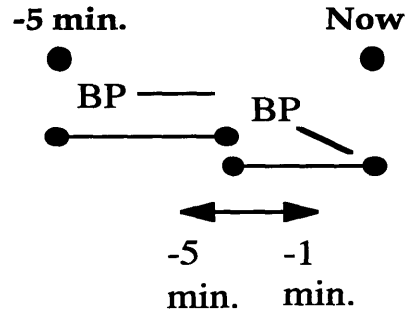


Figure 8 Trend template for a recent decreasing blood pressure.

Similarly, one may define a trend template that examines data both in the recent past and in the near future. A trend template incorporating a landmark of Now is continually instantiated using the method of section 4.1.

3.6.3 Trends Indexed by a Trend Interpretation

Other trends are best indexed by a diagnostic result, including another trend interpretation. Below is a trend template describing an adequate baroreceptor reflex to falling blood pressure [Guyton 1991, page 199]. The lone landmark point is `onset-falling-BP`, representing the time at which blood pressure begins falling in a patient. The intervals and value constraints represent that within zero to five minutes after the blood pressure begins to fall, heart rate begins to rise. This trend

template may be instantiated for a patient, based on the results of matching to the trend template in Figure 8.

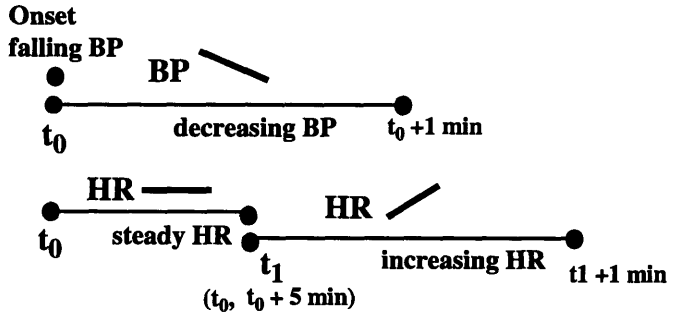


Figure 9 Trend template for adequate baroreceptor response to falling blood pressure.

3.6.4 Hierarchical Trend Templates

A straightforward extension of the current representation is allowing trend templates to include value constraints which are pointers to other trend templates. This results in hierarchical trend templates, and a more space-efficient trend representation. For example, the above baroreceptor response trend template may be referred to by name within larger trend templates that include a fall in blood pressure. Hierarchical trends have also been developed using a finite state machine representation in [Nelson and Hadden 1992].

3.6.5 Limitations of Trend Templates

The trend template representation limits the expressiveness of multivariate trends. Most of these limitations are motivated by the TrenDx matching algorithm, which optimizes a multiple regression fit.

Value constraint models cannot be more complex than linear combinations of fully specified models of time. For example, if the coefficients a_i or b_i are unknown in either the exponential (EQ 6) or sinusoidal (EQ 7) models, these models could not be used in a trend template.

Trend templates cannot express functional relations between parameters in different value constraints or different intervals. For example, one cannot model that the slope of a linear model in one value constraint is double the slope of a linear model elsewhere. This would be useful in a trend like "temperature increasing twice as fast as before."

Trend templates cannot express temporal uncertainty rooted in probability or many other belief distributions. For example, one cannot say for the trend template in Figure 9 above that time t_1 is normally distributed with mean t_0 + 2.5 minutes.

A particular trend template models a trend accurately enough so that appropriate data match that template better than others. Therefore, depending on the application, a trend template model need not be extremely detailed. The reader should consider whether the above limitations are a hindrance to representing distinct trends in each diagnostic context. We have not found the limitations problematic in our research.

3.7 Diagnosis with Trend Templates

A *monitor set* is a set of trend templates forming a clinical context for monitoring. The trend templates within a monitor set are viewed as a partition of trends that may occur in a particular clinical context. The members of a monitor set are concurrently matched against the same patient data by the trend diagnosis program TrenDx. In Figure 10 is a monitor set for checking initial recent trends in blood pressure. The three member trend templates check for a recent increasing, decreasing, or steady blood pressure over the past five minutes.

In a diagnostic setting one trend template within a monitor set is the *expected* or *normal model* while the other trend templates are *fault models*. The fault models are those that if matched well warrant attention by the person or system observing the device. In section 6.2 we describe techniques for generating alarms based on scores of fault models. The monitoring program TrenDx cannot combine models of single faults to produce models of multiple faults. A knowledge engineer must define a distinct trend templates for each multiple fault of interest

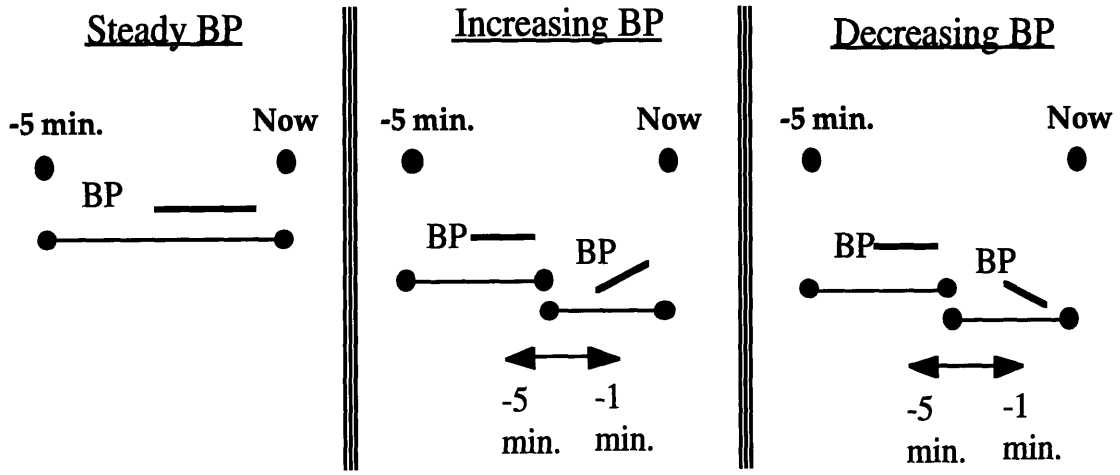


Figure 10 Monitor set for initial classification of blood pressure.

Monitor sets are prescribed for each patient based on his or her clinical condition. For example, an intensive care unit patient may automatically be prescribed

monitor sets for recent trends in blood pressure and other critical parameters. A patient being given a particular drug or treatment may be assigned a monitor set that discriminates between adequate and various adverse reactions to the treatment. A clinician may prescribe to a diabetic patient several monitor sets that track reactions of serum glucose and insulin to various meals, insulin doses, and exercise sessions.

The trend template in Figure 5 for a hemodynamic fault is part of a monitor set that monitors patient response to oxygen handbagging. That complete monitor set, and the results of matching the member templates to clinical data, are presented in section 5.2.

3.7.1 Semantics and Complexity

One should not consider building a trend template in isolation but instead building several competing templates as part of a monitor set for each clinical context. Thus the choices of what parameters to track and what intervals to include in a given trend template are in fact dependent upon what must be represented in the competing trend templates within the same monitor set. This representation issue of defining multiple objects together is recurrent in artificial intelligence. Knowledge engineers of rule-based systems must consider how the rules may interact during forward chaining. Designers of frame-based knowledge bases (e.g. [Haimowitz 1988]) must represent the differences between sibling concepts so that a classifier can distinguish to which group an instance belongs.

The trend templates belonging to a monitor set represent *all* of the patterns of interest in a clinical context. This is a drastic reduction in the complexity of trends that may otherwise be inferred from data. This complexity would otherwise be at least of order $(2^{I \times P})$, for I intervals in a trend template, and P parameters being tracked.

A knowledge engineer may use the trend template representation language to create monitor sets with superfluous intervals or value constraints, or trend templates within monitor sets with inconsistent constraints. To discourage this practice we present in section 6.2 guidelines for optimal monitor sets.

3.7.2 Cascaded Theory of Diagnosis

As we describe in the next chapter, TrenDx matches process data to each active monitor set to determine which trend predominates. A monitor designer may use TrenDx within a monitoring system supporting a cascaded theory of diagnosis. First TrenDx matches to monitor sets which represent a context of *initial classification*. In these monitor sets, trends in key measurements suggest whether a problem exists warranting further exploration. All processes in an application may be assigned initial classification monitor sets. For example, any intensive care unit

patient may be automatically prescribed monitor sets for recent trends in blood pressure (see Figure 10), as well in other parameters.

If the initial classification suggests a possible fault, the monitoring system triggers other monitor sets to further discriminate the diagnosis. The triggering mechanism is described in section 6.4. The trend templates in these monitor sets represent trends in more detailed faults. In a given application, triggering new monitor sets for more detailed diagnoses may proceed for several levels.

Other monitor sets are also triggered during diagnostic monitoring. These include the *various responses expected*, should the initial classification deduce a fault trend. These response monitor sets may be triggered to track the process response to that fault, or to repairs aimed at fixing the fault. In this way the baroreceptor reflex trend template (Figure 9) may be part of a monitor set that can be activated if the `decreasing BP` trend of Figure 10 predominates. Monitor sets are also triggered in response to interventions and therapies applied to the process under observation. For example, in medical domains, one must encode monitor sets for adequate and various adverse reactions to drugs and other treatments.

A knowledge engineer building monitor sets must understand his or her monitoring setting, including the expertise of the clinicians or operators who will receive TrenDx's alarms or and explanations. A primary objective of this research is reducing data overload. Completely automated diagnosis is not necessarily an objective. Beware needless creation of monitor sets or trend templates within them. If the operator has some expertise in distinguishing between ultimate etiologies once presented with a generic fault pattern, then trend templates need not model the ultimate etiologies. Furthermore, if during therapies or intervention an operator capably tracks a limited, relevant subset of measurements, then monitor sets for such interventions may not be required.

The next chapter presents in detail how TrenDx matches process data to trend templates. The matching algorithm is suited to the representation language described in this chapter.

4 TrenDx

TrenDx diagnoses trends by matching time-ordered process data to the competing trend templates in each monitor set assigned to that process. TrenDx begins matching by instantiating each trend template for the monitored process. TrenDx then computes all temporal worlds in which the currently interpreted data may be assigned to intervals of the trend template. Each temporal world represents a different hypothesis for the same trend template. For each hypothesis, TrenDx assigns the data to the appropriate trend template intervals and computes the matching scores of the relevant value constraints. The value constraint scores are combined to an overall error score for each hypothesis. Finally, the top hypotheses for each trend template are maintained via a beam search. The output of TrenDx is a list of the top hypotheses for each trend template within a monitor set, with the score of each hypothesis.

4.1 Instantiating Monitor Sets for a Process

TrenDx instantiates a monitor set for a process by temporally relating each trend template to the process history. TrenDx can then infer the possible temporal locations of process data in each trend template.

4.1.1 Temporal Anchoring

Figure 11 illustrates how TrenDx temporally links a trend template to the patient history. The lower portion of the figure illustrates the temporal relations from part of the normal growth trend template first discussed in section 3.6.1. The upper portion of the figure depicts the temporal relations from the history and data of a particular patient. Before the trend template is instantiated, these two sets of temporal relations are completely separate. TrenDx instantiates a trend template for a process by linking these two sets of temporal constraints with a new temporal relation bridging the two sets. In this case, TrenDx links as equal in time the `birth` landmark point of the trend template and birth date in the patient history.

We may call the `birth` landmark point an *anchor point* because it is this point which is temporally linked to an episode in the patient (more generally, process) history. An anchor point may be any point in a trend template: a landmark point, an interval begin point or an interval end point. The linking temporal relation between the history point and the trend template point need not be temporal equality, but may be a range `(min max)` of times.

In this example, the episode in the process history, the patient's birth date, was immediately accessible. More generally, the process event to which the instantiated trend template is linked may involve some calculation. For the templates in the blood pressure monitor set of section 3.7, the anchor point is the landmark Now and the event is the time of the patient datum being currently processed. For the baroreceptor reflex trend template of section 3.6.3, the process episode is determined based on the match of another trend template.

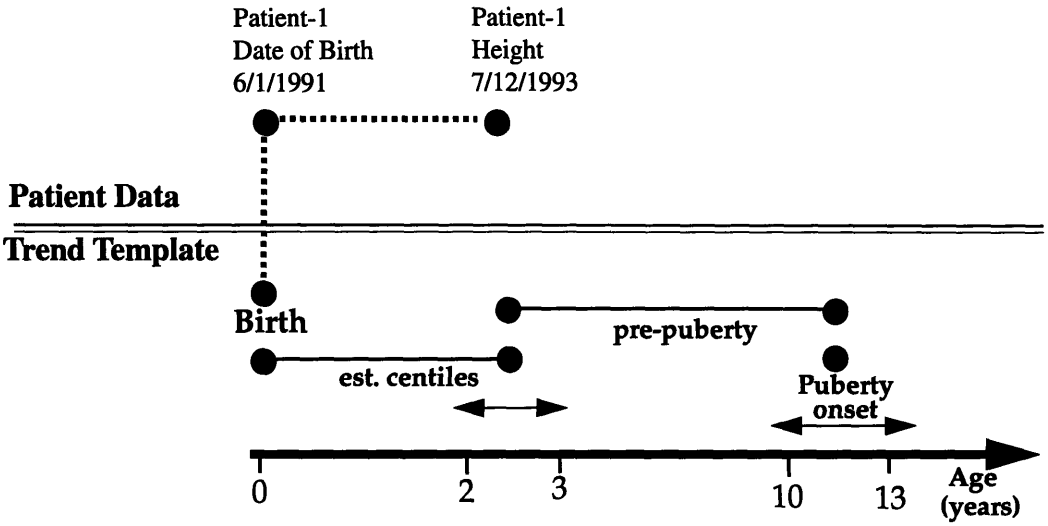


Figure 11 Anchoring a trend template landmark to the patient history.

The different trend templates within a monitor set must include the same anchor points. TrenDx simultaneously instantiates each member trend template to the same anchor point and matches the same set of process data to each template.

Once a trend template is instantiated, process data may be temporally placed with respect to the trend template intervals. In the figure we see that the height datum of `Patient-1` occurs roughly 2.1 years after the birth date of that patient. By traversing the series of temporal links, TrenDx can infer that the time of this height datum may occur *either* within the first or second intervals of this growth trend template. Thus the temporal uncertainty of trend templates leads to different temporal interpretations of how the data match. We call each such temporal interpretation a *chronology.* We call the corresponding placement of data into trend template intervals an *assignment.*

4.1.2 Temporal Contexts

Temporally anchoring a trend template to a monitored process is implemented using the *context* mechanism available in some temporal reasoners (e.g. TUP). A temporal context is a collection of consistent temporal assertions that offers a single interpretation of events in the world. For the purposes of TrenDx, a temporal context is used to describe either:

- the temporal relations in a process history,
- the temporal relations in a trend template,
- a *chronology* of how the temporal uncertainty of a trend template is refined when matched to process data.

Temporal contexts are necessary for TrenDx to reason about alternate assignments of data to trend template intervals. Each such assignment yields different temporal distances within a trend template, thereby requiring a distinct temporal context.

Contexts are arranged hierarchically in a tree, with temporal assertions from a parent context inherited by child contexts. Siblings in the context tree represent alternate possible chronologies.

The general architecture for temporal contexts in TrenDx appears below:

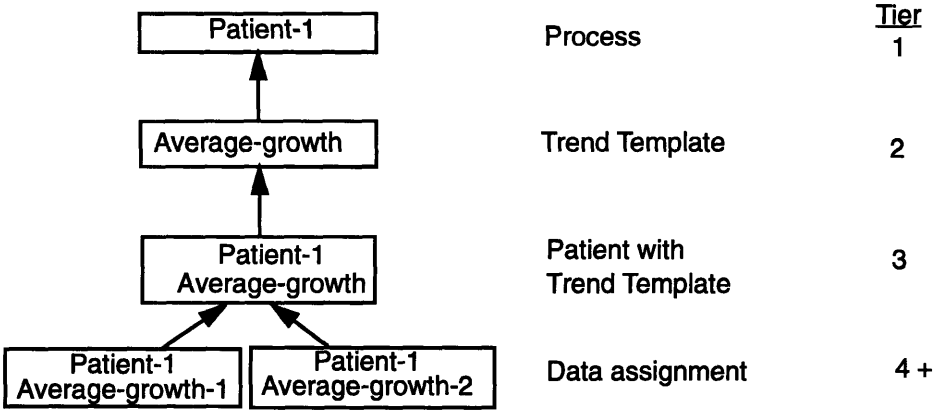


Figure 12 The temporal context hierarchy of TrenDx.

The root context for each process contains temporal information about that process. For a medical patient this information may be reported by the physician or hospital database, including the times of laboratory data and of significant life events, as well as the birth date. The times of events may be stated absolutely, by the Gregorian calendar, or relatively, within some bounds of another event in the patient context.

The average growth trend template presented earlier has its own context, which contains all of the temporal points and relations of that template. TrenDx assigns a hypothesis to a process by placing the context for the trend template of that hypothesis as a child of the process context[1]. TrenDx then places under the trend template context a third-tier context for temporal assertions about the process data

matching the trend template. In this third-tier context we create the linking relation that anchors the trend template to the process history.

The hypothesis of average growth for `Patient-1` initially only has the one chronology with a third tier context. When patient data are interpreted and assigned to intervals of the trend template, there may be alternate assignments and therefore alternate chronologies. In that case TrenDx branches to multiple "data assignment" hypotheses and associated (fourth tier) contexts. The context tree of TrenDx is similar to that used by the Clinician's Assistant, another TUP based monitoring system [Kohane 1992].

4.2 Structure of a TrenDx Hypothesis

A TrenDx hypothesis *hyp* for a process *Pr* consists of three main components:

1. a trend template TT(*hyp*)
2. a temporal context CONTEXT(*hyp*) containing relations between the times of data and the points in TT(*hyp*), and
3. an assignment ASSIGNMENT(*hyp*) of patient data to the intervals of TT(*hyp*). ASSIGNMENT(*hyp*) is a relation {(*int d*)}, where *int* is an interval in TT(*hyp*) and *d* is an interpreted datum of *Pr*.

Given TT(*hyp*), CONTEXT(*hyp*), and the data of *Pr*, the assignment ASSIGNMENT(*hyp*) is uniquely determined. ASSIGNMENT(*hyp*) is used by TrenDx in matching process data to value constraints

4.3 Processing a Data Cluster

The reasoning in TrenDx fits a data-driven process monitoring cycle as described in section 1.2. TrenDx processes data forward in time, reasoning simultaneously about a *cluster* of data that are measured at the same time slice for the same process *Pr*. For example, a pediatric growth visit may yield a single data cluster of a four data: height, weight, bone age, and pubertal stage. Each data cluster is matched to all hypotheses of *Pr* that constrain some of the parameters of the data in that cluster. For each such data cluster D_t, where t is the time of all data in the cluster, TrenDx evaluates PROCESS-DATA-CLUSTER(D_t), detailed below.

Let us define two predicates to be used in this algorithm. Let *d* be a datum of the process *Pr*, let *hyp* be a hypothesis of *Pr*, and let *int* be an interval in TT(*hyp*). We define

1. Actually, in theTrenDx implementation using TUP, a copy of the temporal context for the trend template is used. A trend template must be assigned to multiple patients. In the TUP context tree, every context has at most one parent.

CONSTRAINS(*int*, *d*)

to be true if and only if some value constraint bound to *int* constrains *d*. For a temporal context *tc* (see section 4.1.2), we define

WITHIN(*d*, *int*, *tc*)

to be true if and only if the time of *d* is definitely between or equal to the times of the start and end points of *int* in *tc*.

In the algorithm below *B* is a positive integer denoting the number of hypotheses per trend template retained in a beam search.

Algorithm: **PROCESS-DATA-CLUSTER** [D_t: data cluster]

Let *Pr* = the process of D_t.

Add D_t to the history of data for *Pr*.

For all monitor sets *M* in the record of *Pr* that constrain any parameters in D_t:

For all trend templates *TT* in *M*:

Let H = {*hyp* | TT(*hyp*) = *TT*} be the hypotheses with trend template *TT*.

For each *hyp* in H:

Let *tconts* = CONSISTENT-TEMPORAL-CONTEXTS(D_t, *hyp*) (see section 4.3.3).

If length of *tconts* = 1 [Let *tc* be the lone element in *tconts*].

Set ASSIGNMENT(*hyp*) = ASSIGNMENT(*hyp*)

$\cup$ {(*int d*) | d ε D_t and WITHIN(*d*, *int*, *tc*) and CONSTRAINS(*int*, *d*)}.

Update match to all value constraints in intervals with data of D_t.

Else Let *k* = length of *tconts*. [Call them tc_1, tc_2,... tc_k]

Link the contexts tc_1, tc_2,... tc_k as children of CONTEXT(*hyp*).

Create *k* new hypotheses hyp_1,..., hyp_k.

Set CONTEXT(hyp_i) to tc_i and set TT(hyp_i) to TT(*hyp*).

For *i* = 1 to *k*

Set ASSIGNMENT(hyp_i) = ASSIGNMENT(*hyp*)

$\cup$ {(*int d*) | d ε D_t and WITHIN(*d*, *int*, tc_i) and CONSTRAINS(*int*, *d*)}.

Update match to all value constraints in intervals with data of D_t.

Set CONTEXT(hyp_i) to C_i and set TT(hyp_i) to TT(*hyp*).

Add hyp_i as a hypothesis of *Pr*.

End For *i*

Remove *hyp* as a hypotheses of *Pr*.

End If

End For *hyp*

Update matching scores of all hypotheses of *pat* with trend template *TT*.

Prune all but the top *B* hypotheses of *Pr* with trend template *TT*.

End For *TT*

Send alarms or trigger other monitor sets as appropriate based on leading scores of hypotheses with trend templates in *M*.

End for *M*.

END PROCESS-DATA-CLUSTER.

To summarize, TrenDx computes for each relevant hypothesis *hyp* all the temporal contexts for assigning the data to the intervals of TT(*hyp*). For each such temporal context, a new hypothesis is created with an updated assignment of some new data to some intervals. Matching scores of the new hypotheses are updated. Then, in a use of beam search, the best *B* scoring hypotheses per trend template are retained, and the others pruned. Finally, the best scoring hypotheses per trend template may be used for sending an alarm or triggering other monitor sets.

4.3.1 Example of TrenDx Processing

Let's illustrate the above algorithm by summarizing how TrenDx processes two data clusters of the pediatric growth patient (from section 2.1.3), whom we shall denote *Pat2*. TrenDx matches the data of *Pat2* to the male average normal growth trend template partially described in section 3.6.1.[2] Presume for now that this trend template consists only of the two intervals shown in Figure 7 of section 3.6.1. Recall the temporal relationship between these two intervals:

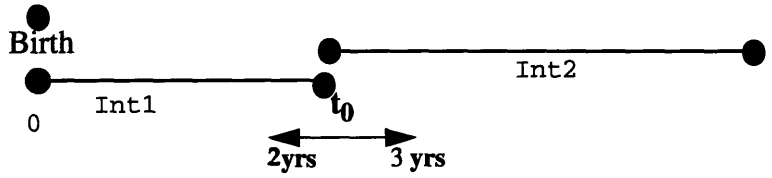


Figure 13 Portion of the average normal growth trend template.

The two intervals are consecutive phases and thus essentially meet at a time t_0 between 2 and 3 years after the landmark `birth`. From initial instantiation before any data are interpreted, *Pat2* has one hypothesis, call it *hyp*. TT(*hyp*) is male average normal growth. ASSIGNMENT(*hyp*) = ∅. The first data cluster $\{hd_1, wd_1\}$ to be processed are a height and weight at age 2.1. Because this age falls between 2 and 3 years, the data in that cluster may be placed in either `Int1` or `Int2`. TrenDx computes two new temporal contexts and two new hypotheses with these contexts and the appropriate data assignments. Let's call the new hypotheses hyp_1 *and* hyp_2; the assignments and contexts are below.

2. TrenDx also matches the data to the competing growth trend templates within the same monitor set. We focus here on the normal template.

ASSIGNMENT(hyp_1) = {(*Int1*, hd_1)(*Int1*, wd_1)},
ASSIGNMENT(hyp_2) = {(*Int2*,hd_1) (*Int2*, wd_1)}.

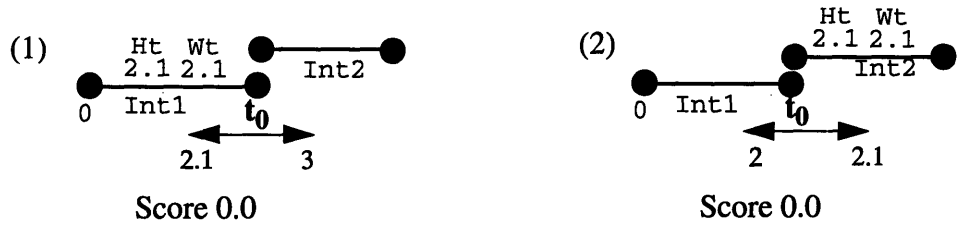


Figure 14 Alternate hypotheses of the average normal growth trend template with differing chronologies.

TrenDx has added additional information to the context of each new hypothesis. To CONTEXT(hyp_1) it adds that `(end Int1)` is after hd_1. To CONTEXT(hyp_2) TrenDx adds that `(end Int1)` is before hd_1. TrenD_x removes *hyp* from the active hypotheses of *Pat2* and adds hyp_1 *and* hyp_2. This is illustrated in Figure 14.

TrenDx's different temporal interpretations of the same data is related to the Time Map Manager's [Dean and Boddy 1987; Dean and McDermott 1987] different accounts of which boolean propositions are true over which temporal intervals. The main difference is that TrenDx operates on primary data. The *resulting* trend diagnoses of TrenDx may in fact serve as propositions managed by the Time Map Manager.

Returning to our example, the next data cluster of a height and weight, {hd_2, wd_2}, is at age 3.1. In processing this data cluster, TrenDx visits in turn each of the two active normal growth hypotheses (as well as the active hypotheses for competing trend templates). For each hypothesis, there is only one possible temporal context, because the current context constrains that hd_2 and wd_2 must belong to `Int2`. Recall that the `(end Int2)` is constrained to 10 to 15 years after birth. TrenDx adds to the assignment of each hypothesis; there is no need to update the context of either hypothesis.

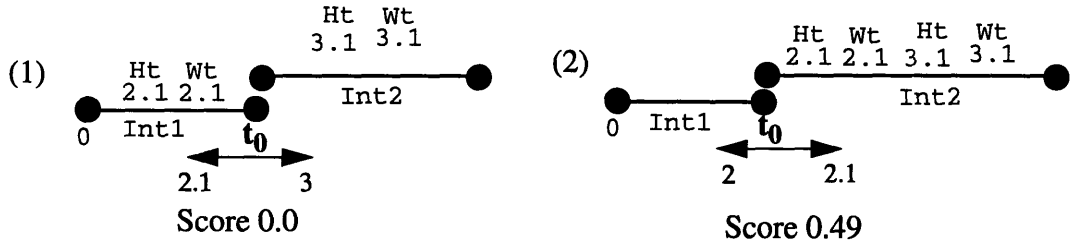


Figure 15 Continued matching to two normal growth hypotheses.

Each of the pictured hypotheses is labelled with a matching score. The scoring mechanism, which calculates goodness of fit to the low-order polynomial models of value constraints, is detailed in section 4.4.2.

4.3.2 Space of Alternate Chronologies

Left unspecified in the algorithm PROCESS-DATA-CLUSTER was how to compute all consistent temporal contexts for placing a time point within several trend template intervals. Before presenting the algorithm CONSISTENT-TEMPORAL-CONTEXTS(D_t, *hyp*) that computes these contexts, it is instructive to understand the complexity of this chronology space.

Say TrenDx is matching to a trend template with I intervals, and processes T data clusters, with the clusters at successive times t_0, t_1,...,t_T. The times for the begin and end points of some intervals may be highly unconstrained, as shown for `Int1` in the figure below.

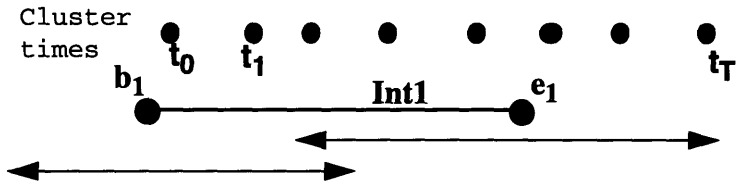


Figure 16 There are $O(T^2)$ placements of T data clusters in a maximally uncertain trend template interval.

We find the number of different chronologies for assigning the data clusters to `Int1`, by noting that for every chronology there is a smallest i, $1 \le i \le T$, for which b_1 is after time t_i. For each such i, similarly there is a smallest j, $1 \le j \le T$, for which e_1 is after time t_j. Also, j must be greater than or equal to i, because the end point of an interval must be after the begin point. Thus the number of chronologies is bounded above by the same order as the triangular number of size T: $O(T^2)$.

We extend this to the number of chronologies for matching the data cluster to a trend template with I intervals, we merely note that each interval may be similarly unconstrained, and that any combination of interval positions may make up a trend template chronology. The maximal number of chronologies is $O((T^2)^I) = O(T^{2 \times I})$.

Two aspects of trend templates drastically lower the number of chronologies in practice to well below this upper bound. First, the intervals of a trend template are *not* completely temporally unconstrained as suggested in Figure 16; interval begin and end points are temporally linked to each other and to landmark points. Indeed, the tighter these temporal constraints, the smaller the total chronology space. Second, usually a trend template has a maximal temporal extent over which data may possibly fit within any intervals. In pediatric growth templates this may be 20

years. In short-term intensive care unit trend templates (see section 3.4), this may be a half hour. Let T_m be this maximal temporal extent divided by the shortest sample period of any process parameter constrained in the trend template. Note then that both T_m and I are constant for a given trend template, and that the maximal number of chronologies, $O(T_m^{2 \times I})$, is in fact *constant* for a given trend template, independent of the process or the data stream. Consequently, the limits on considering all possible chronologies are limits inherent in our computational resources, but not in the computation itself.

4.3.3 Computing Consistent Temporal Contexts

Having considered the size of the chronology space, we return to the algorithm for computing the consistent temporal contexts for a given data cluster, CONSISTENT-TEMPORAL-CONTEXTS(D_t, *hyp*). TrenDx uses a straightforward recursive implementation, based on two insights:

1. A time point may be exactly one of {before, within, after} a time interval in a given temporal context.
2. Any temporal context placing a point with respect to intervals $\{I_1,..., I_{k+1}\}$ may be computed by some placement of the point with respect to intervals $\{I_1,... I_k\}$ and adding one more temporal relation.

Algorithm: **CONSISTENT-TEMPORAL-CONTEXTS**[D_t,data cluster; *hyp* hypothesis]

Let `contexts` = {CONTEXT(*hyp*)}
For each interval *int* in TT(*hyp*) that constrains some parameter in D_t.
  For each temporal context *tc* in `contexts`
  Determine which of {before, within, after} may describe how the time point of D_t relates to *int*.
  If there is more than one answer, say k (k = 2 or k =3):
    Create k contexts $tc_1,..., tc_k$ as children of *tc*.
    Set `contexts` = `contexts` $\cup \{tc_1,..., tc_k\}$ - *tc*
  End If
  End For *tc*
End For *int*
End CONSISTENT-TEMPORAL-CONTEXTS

Note that we only need to iterate over the intervals of TT(*hyp*) that actually constrain parameters of some data of D_t. The temporal placement of other intervals in the trend template is not germaine to the matching of D_t to *hyp*.

As stated in section 4.2, once a temporal context is computed for a given hypotheses and data set, the assignment of data to intervals is uniquely determined. TrenDx in fact computes the assignment as it computes the temporal contexts, by

supplementing the above algorithm with bookkeeping of what intervals each datum should be assigned to.

4.4 Scoring a Hypothesis

4.4.1 Matching Data to Value Constraints

TrenDx matches newly processed data to a hypothesis *hyp* first at the local, value constraint level. Recall the value constraint representation defined in section 3.5.3: in brief, each value constraint consists of:

1. a function that computes from data assigned to the interval a real-valued sequence $\{Y_t\}$, and
2. a low-order polynomial regression model with either qualitative or quantitative constraints.

Let us denote by D(*int*, *hyp*) the data assigned to interval *int* in the hypothesis *hyp*. TrenDx matches these data to each value constraint *vc* of *int* by computing the real-valued sequence $\{Y_t\}$ and estimating the parameters of the model of *vc* by minimizing sum of squared error between $\{Y_t\}$ and the values estimated by the model $\{\hat{Y}_t\}$. This widely used minimization is called *ordinary least squares (OLS)*. Clear explanations of the matrix solution to OLS may be found in [Strang 1988, chapter 3] and [Draper and Smith 1981, chapter 2].

Once the parameters of the linear regression model have been estimated, one must consider how to measure the quality of fit of the instantiated model to the $\{Y_t\}$. One measure of this quality of fit used in analysis of variance tables (see applied statistics texts, e.g. [McPherson 1990, chapter 18]) is the *residual mean square:*

$$\text{Residual Mean Square} = \sum_t \frac{(\hat{Y}_t - Y_t)^2}{DF}. \qquad \text{(EQ 9)}$$

The denominator DF is the number of degrees of freedom in the regression fit. In a sample of N data where p parameters are estimated, DF = (N - p).

TrenDx measures quality of fit using a variation of the residual mean square that allows comparison of fit to $\{Y_t\}$ of different magnitudes. The goodness of fit of *value constraint vc* for the hypothesis *hyp*, denoted by Fit(*vc, hyp*), is the mean absolute percentage error (MAPE) between function values $\{Y_t\}$ and regression model estimations $\{\hat{Y}_t\}$:

$$Fit(vc, hyp) = \frac{\sum_t \left| \frac{\hat{Y}_t - Y_t}{Y_t} \right|}{DF(vc, hyp)} \quad \text{(EQ 10)}$$

Note that the expression Fit*(vc, hyp)* is expressive enough to account for the interval *I* (a unique property of vc) and the data assigned to *I* (a unique property of *hyp*).

MAPE is particularly useful for comparing the goodness of fit between models of different variables [Kvanli, Guynes et al. 1989, section 16.7]. Variables at a smaller scale often have a smaller variance of residuals than those at a larger scale; a measure of *relative error* is impervious to such differences. Trend templates combine trends in variables that may be at different orders of magnitude. An acid-base medicine trend template may include trends in serum sodium, with normal range 135-145, and serum potassium, with normal range 3.5 to 5.0 [Sculley 1986]. A goodness of fit score based on relative error, such as MAPE, tends to normalize over the differing ranges of values.

The denominator of this fit, DF(*vc*, *hyp*) is the same as that used in the residual mean square: the size of the set $\{Y_t\}$ minus the number of parameters estimated in the model: DF = (N - p). For the low-order polynomial models of trend templates, where some parameters may have a priori quantitative estimates, p is between 0 and 3, inclusive.

As special cases, if the number of values matched to a value constraint results only in a trivial match (e.g. a linear model fitting but two data), the score for that value constraint is considered "trivial" and is not incorporated into the overall hypothesis score.

4.4.2 Combining Scores for a Hypothesis

The goodness of fit of a hypothesis *hyp* to the data assigned to the intervals of TT(*hyp*) is a weighted average of the fits to the individual value constraints, where the weight is the fraction of the total number of degrees of freedom used in matching to that value constraint:

$$Fit(hyp) = \frac{\sum_{VC \in TT(hyp)} DF(vc, hyp) \cdot Fit(vc, hyp)}{\sum_{VC \in TT(hyp)} DF(vc, hyp)} \quad \text{(EQ 11)}$$

Consequently, the import of each value constraint in determining the overall goodness of fit to a trend template is roughly proportional to the number of data matched to that value constraint. In essence TrenDx treats the entire trend template as a regression model, and computes the percentage error match to the larger

model. A hypothesis begins with a score of 0. That score remains at 0 until enough data are assigned to an interval to have a non-trivial OLS fit.

The above weighting scheme tends to equalize the effects of all data. Other weights on value constraint scores may be used. If a value constraint monitors sparse but diagnostically important data, the above weighting scheme may minimize the effect of that value constraint. One means of addressing this problem is for a knowledge engineer to weight each value constraint by diagnostic import.

In section 3.6.4 on page 47 we described hierarchical trend templates, where a value constraint may be a pointer to another trend template. For hierarchical trend templates the scoring algorithm could naturally extended. The matching score for a pointer value constraint would be the lowest matching score for the trend template the value constraint points to.

4.4.3 Distribution Assumptions on $\{Y_t\}$

TrenDx applies OLS to time-series linear regression models, which requires assumptions on the statistical distribution of $\{Y_t\}$. These are listed in [Ostrom 1990, pages14-16] and originally appeared in [Pindyck 1981, pages 47-51]. Recall that the time series regression model in a trend template value constraint has the form:

$$Y_t = \left(\sum_{i=0}^{N-1} a_i t^i \right) + \varepsilon_t, \text{ for N = 1,2, or 3.} \qquad \text{(EQ 12)}$$

Of the six assumptions given, the first four (listed as they apply to TrenDx) are explicit assumptions about the signal $\{Y_t\}$ and the error that are fairly straightforward, yet must be recognized by a TrenDx user:

1. $\{Y_t\}$ is in fact {constant, linear, or quadratic} in time t.
2. $E(t\varepsilon_t) = 0$ (nonstochastic).
3. $E(\varepsilon_t) = 0$ (zero mean).
4. $E(\varepsilon_t^2) = \sigma^2$ (constant variance).

The fifth distribution assumption may have more of an impact on the quality of a TrenDx match:

5. $E(\varepsilon_t \varepsilon_{t-m}) = 0$ for $m \neq 0$ (nonautoregression).

As well described in [Ostrom 1990, pages 21-26], should the error terms in a time series regression model be autocorrelated, then the estimated coefficients (e.g. slope of the OLS line) may be significantly different from the actual coefficients of the linear model. A consequence can be an *underestimate* of the error.

A widely used test to check for autocorrelation of errors is the Durbin-Watson d-statistic, described in most texts on regression, e.g. [Wittink 1988, page 189].

The statistic ranges from 0 to 4. A value close to 2 strongly suggests acceptance of the null hypothesis of no autocorrelation and a value close to 0 or 4 strongly suggests rejection of this null hypothesis in favor of autoregression. Small experiments using this statistic on intensive care unit hemodynamic and respiratory data suggest that the error in matching to linear and constant value constraint models is *not autocorrelated.* Similar experiments on height and weight data used on growth templates yield mixed results: some data show autocorrelation while others do not. Should one know a signal to have autocorrelated errors, one may need to replace TrenDx's use OLS with some form of generalized least squares for that signal.[3]

The final assumption of a time series regression model is incidental to TrenDx:

6. The error term ε_t is normally distributed.

This assumption is only needed for testing statistical hypotheses about the match of data to the regression model, or for predicting confidence intervals of future data.[4] TrenDx does neither of these. The program does not test value constraint matches as statistical hypotheses. TrenDx does not predict future values of $\{Y_t\}$; it re-estimates model parameters with OLS each time new data are added to the trend template interval.

4.4.4 Nearest Satisfier Value Constraint Model

Recall from section 3.5.3 that the low-order polynomial models for trend template value constraints may include either quantitative or qualitative constraints on the polynomial coefficients, or on derivatives. A quantitative constraint may be a numerical value or a range of values. If all coefficient constraints provided by the knowledge engineer are numerical values, TrenDx matches to a value constraint by estimating the *unconstrained* coefficients by ordinary least square (OLS) and then computing errors as in section 4.4.1. If the knowledge engineer provides either numerical range or qualitative coefficient constraints, TrenDx first fits the data and estimates coefficients with OLS. If the OLS-estimated coefficients meet all coefficient constraints, the matching score is the MAPE, as usual. If these estimates do *not* meet some numerical range or qualitative coefficient constraints, then TrenDx matches the same data to the nearest polynomial of the same degree that *does satisfy* the coefficient constraints

3. The use of GLS has not been implemented in TrenDx as applied to intensive care unit or pediatric growth signals.

4. Assumptions 1-5 are also required for these tasks.

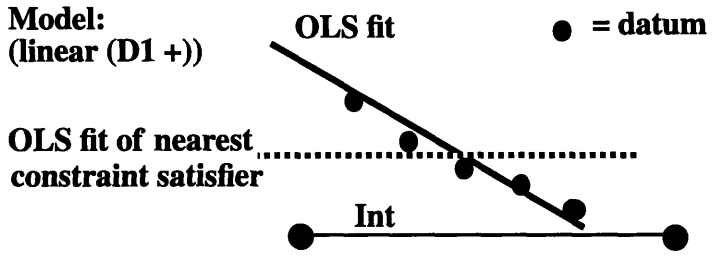


Figure 17 When a value constraint's qualitative parameter estimate fails, TrenDx matches to the nearest constraint satisfier.

The diagram above illustrates this "nearest satisfier." When the OLS match to an Nth order polynomial violates a qualitative constraint on the highest order coefficient, the nearest satisfier is the degenerate Nth order polynomial whose highest order coefficient is zero. In the figure, the value constraint calls for an increasing line and the data are decreasing. The closest increasing line to the data is a constant line. Generally, an (N - 1)st order polynomial is the limit of the "nearest" Nth order polynomial satisfying the highest order coefficient's qualitative constraints.[5] To illustrate a numerical range constraint, say a value constraint model calls for a linear increase with slope constrained between 3 and 5 units per minute. If the OLS estimated slope to actual data within the interval were 6 units per minute, then the nearest satisfier is the best fitting line of slope 5 units per minute.

One advantage of this technique is that the further away the data are from fitting the qualitative parameter constraints, the higher the error score to the nearest satisfier. In the figure, the more decreasing the data, the worse the error score is to the constant model. A *disadvantage* is that when the data only slightly violate the constraint, the matching score is roughly the same as the match to a value constraint modeling the data as fitting the nearest satisfier. In the figure, if the data were *slightly* decreasing, the match would be close to the match to a constant value constraint with no parameter constraint. Should the latter value constraint be on a competing trend template, TrenDx would not significantly discriminate between the two templates over the span of this data.

To compensate for this lack of discrimination for slight violations, TrenDx assigns a multiplicative penalty for violation of a numerical range or qualitative coefficient constraint. In all simulations and experiments within this thesis, the penalty is 1.25. The penalty may be set a TrenDx user.

5. Should a quadratic value constraint suffer a violation of a qualitative *first* derivative in matching the data, then TrenDx finds the nearest satisfier to be a constant model.

4.4.5 Importance of Removing Outliers

Polynomial regression matching by ordinary least squares may exhibit markedly different qualitative behavior in the presence of outlying data. The figure below illustrates that a single outlying point added to an increasing, concave data set can change the OLS best fitting quadratic model to a convex and at times decreasing one. In light of this, added care must be taken to insure that outliers are removed from data in the value constraint *function* (see section 3.5.3) before matching to the value constraint model.[6]

If a value constraint has already fit many data well, then a new outlying datum may be removed by comparison of how well the new datum fits to the model estimated using previous data. While in time series regression new data cannot generally be predicted by probabilistic means, empirical means can be used. TrenDx could, for example, compare the residual in fitting the new datum to the past polynomial estimate to the residuals from other data. If the new residual is unusually high (say more than 3 sample standard deviations), the point may be removed.

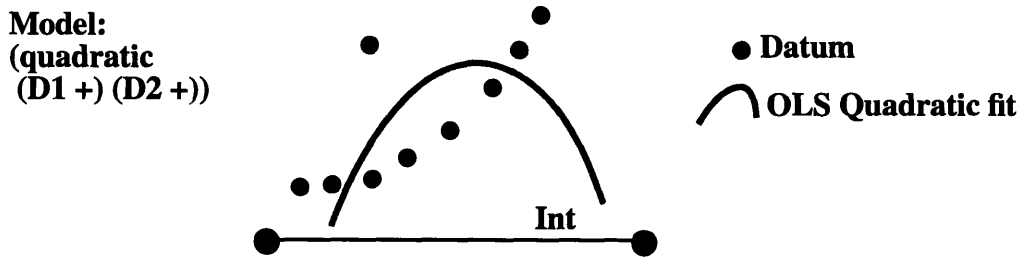


Figure 18 Outlying data may drastically alter the OLS quadratic fit.

Another statistical technique that is useful with many data is known as jackknifing. Each datum is removed from the set, and the goodness of fit calculated to the remaining data. If the goodness of fit without some datum d is substantially lower, then d is removed as an outlier.

4.4.6 Discriminatory Power of Low-Order Polynomials

I conducted several experiments using data generated from stochastic models to determine the discriminatory power of the low-order polynomial models used in value constraints. The methods and results of these experiments are presented in Appendix C.

6. Currently TrenDx does not remove outliers. The techniques described here are straightforward extensions.

The experimental results can help one choose a low-order polynomial within a value constraint modeling some signal $\{Y_t\}$. One needs to consider at least three attributes of the signal: the expected spread of values, the expected noise, and the likely distinct qualitative shapes. Generally one may model the signal using each of the expected qualitative shapes; each shape in a value constraints of a distinct trend template. However, if the noise in $\{Y_t\}$ is sufficient to blur the distinction between those shapes, then modeling with the highest order polynomials (i.e. quadratic) may yield little discrimination. Whether or not this blur occurs depends on the expected noise distribution and the spread of values of $\{Y_t\}$.

4.5 Pruning Hypotheses by Beam Search

Figure 19 illustrates that with each new data cluster D_T, TrenDx branches hypotheses of the same trend template to new hypotheses of different chronologies. After TrenDx matches these competing hypotheses, the program prunes the number of hypotheses through *beam search.* The top *B* scoring hypotheses per trend template are retained, where B is a positive integer. The other hypotheses are discarded.[7] In the figure, the trend template being matched is a normal model, T equals 4, and B equals 3.

TrenDx uses beam search exclusively for efficiency reasons. The worst-case number of chronologies for T data clusters in a trend template with I intervals is $O(T^{2I})$, and the resources needed to process all of these on a serial computer[8] are prohibitive. The time to interpret a data cluster grows linearly with the number of hypotheses. Furthermore, by applying beam search TrenDx keeps the width of its temporal context tree (section 4.1.2) constant, allowing for reclaiming of space.

4.5.1 Larger Beam Means Sounder Reasoning

In applying beam search as it processes data forward in time, TrenDx introduces this inference rule into its diagnostic reasoning:

- The best matching hypotheses in matching data to a trend template through time T are descendants (in the temporal context sense) from the best matching hypotheses at time T - t, for some $t > 0$.

An inference method is called *sound* [Enderton 1972, page 124] if the conclusions it deduces are true according to some semantics. For a diagnostic system, the

7. To avoid arbitrary removal of hypotheses, any chronologies with score equal to one of the top B are also retained.

8. In theory, a parallel computer with hundreds of processors may be used to explore a large fraction, but not necessarily all, of the chronologies.

semantic model is grounded in the "reality" of the process being diagnosed and so must be determined by an expert. Questioning the soundness of the above inference rule illustrates a trade-off of TrenDx.

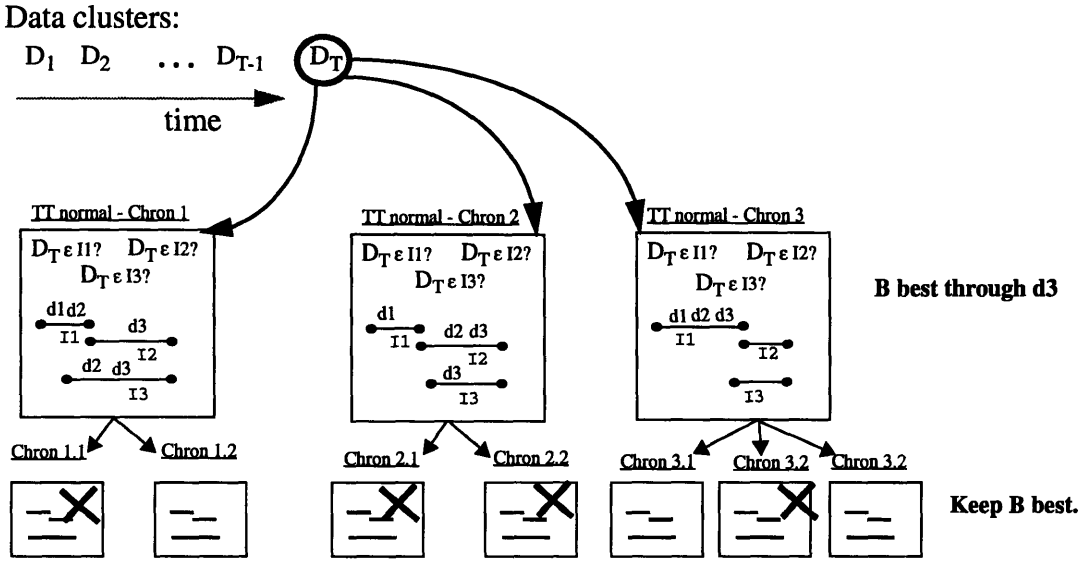


Figure 19 Branching and pruning of TrenDx chronologies for a single trend template.

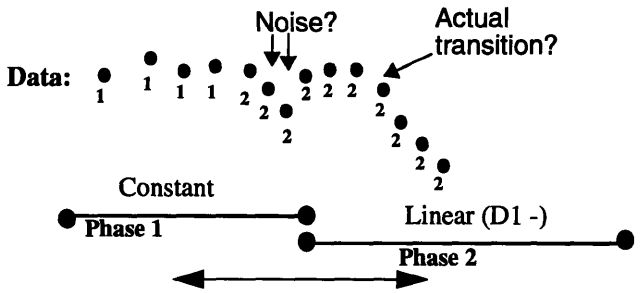


Figure 20 TrenDx with a small beam width may be susceptible to noisy signals.

As suggested by the above diagram, use of a small beam width may lead to unsound trend detection. Momentary noise in a signal may lead TrenDx to assign data to a new process phase prematurely. In the figure, a few consecutive points match well to a linear decreasing model, possibly removing another chronology that is consistent with the broader trend. In the broader trend, those few points are part of a long constant phase. Therefore, the soundness of the TrenDx beam search grows as the beam size gets larger, for then the program will be less susceptible to momentary, transient trends. This increased soundness comes at the expense of larger space and time costs.

4.5.2 Prediction Granularity: Temporal Soundness vs. Efficiency

The depth of the temporal context tree (section 4.1.2) grows linearly with each data cluster processed. As the time and space performance of TrenDx degrade linearly with this depth, TrenDx may gain by processing data clusters only as frequently as necessary.

Included in the representation of a monitor set is a *prediction granularity*: the maximum temporal distance between the actual time of an event in a trend template (landmark or interval point) and TrenDx's reporting of the time for that event. For example, a knowledge engineer may specify that TrenDx should report the times of pediatric growth events to within 3 months accuracy. Given a prediction granularity T_g for a monitor set, TrenDx may process data clusters up to $T_g/2$ apart, for this is frequently enough to report events with temporal accuracy T_g. In the growth example, data clusters need only be processed every 1.5 months.

There are two ways to process data clusters less frequently. One method is to filter data between the processed data clusters. Filtering is troublesome for sparse data sets because valuable data may be eliminated. The second approach, used in TrenDx, is to interpret all data, but to treat all the data clusters of the most recent $T_g/2$ time period as occurring at the *same time slice.* TrenDx does not lose data, but it may make temporally unsound decisions in assigning the data to intervals.

In the growth example, TrenDx may process a patient's data between ages 2 years 11 months and 3 years as if it all occurred at age 3 years. If a growth trend template interval constrains measurements between ages 3 to 5, some of the patient's data *before* age 3 will be assigned and matched in this interval. The hypothesis with this assignment will differ somewhat in matching score from a hypothesis based on only temporally sound assignments. The unsoundness and scoring difference may be reasonable costs for the benefit of vastly increased time and space efficiency.

4.6 TrenDx Output

The output of TrenDx matching data to a monitor set is a list of the top B hypotheses per trend template at any time. TrenDx matching data to trend template is an optimization problem over the space of different hypotheses for that trend template. One natural way to compare the match of competing trend templates is to compare the scores of the best matching hypotheses for each template.

The figure below is a corresponding two-dimensional plot showing TrenDx matching to a monitor set of four trend templates. Early in the matching, the normal model scores best. Over time, one of the fault models scores consistently better than the normal model and other fault models. In chapter 5 we use such plots to describe results of TrenDx matching to pediatric growth and ICU data. In chapter 6

we discuss how to apply the score differentials shown in Figure 21 to sending alarms or summarizing data.

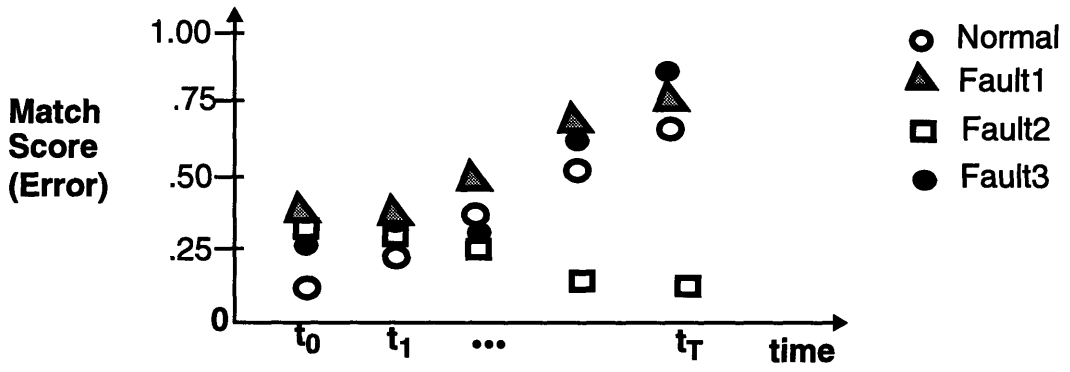


Figure 21 TrenDx output. One fault template continually matches better than normal template.

Each hypothesis also includes temporal restrictions on each point in its trend template. A TrenDx user may ask the temporal distance between any two trend template points within a hypothesis. The result is a range [min max] of distances in seconds, and can be readily converted to a distance in any time scale.

4.7 Computational Complexity of TrenDx

We can compute the worst-case time complexity TrenDx takes to process a data cluster and match to M monitor sets by examining the algorithm PROCESS-DATA-CLUSTER in section 4.3. Let the beam size be B, let the monitor set have T trend templates with an average of I intervals and V value constraints per trend template. Say we have processed D data clusters in the patient history.

Because beam search has been applied, B hypotheses remain for each trend template within each monitor set, for a total of BMT hypotheses. For each hypothesis, the procedure CONSISTENT-TEMPORAL-CONTEXTS queries each of the I intervals about the distance between a point and a a temporal interval. This is in fact two temporal queries, each of which is cubic in the number of total points [Vilain and Kautz 1986], in this case $O((D+I)^3)$. Thus the total temporal query time takes $O(I(D+I)^3)$. As can be seen from the definition of CONSISTENT-TEMPORAL-CONTEXTS, the worst-case number of new temporal contexts and thus proposed new hypotheses is $O(2^I)$. For each of new hypotheses, the value constraint matching by OLS is essentially a series of matrix multiplications and inversions, where the matrix size is O(D). This part of the algorithm therefore takes $O(D^3)$ for each of V value constraints or $O(VD^3)$. The final scoring of each hypothesis takes O(V).

Combining these partial orders of complexity, the combined time complexity is $O(BMT[I(D+I)^3 + 2^I VD^3])$. The only exponential part of this time complexity is the 2^I denoting the different number of new chronologies found. For a richly structured trend template, such as those illustrated in chapter 3, the number of new chronologies is far fewer. For example, the hemodynamic fault template, having eight intervals, includes three pairs of consecutive phases during the handbagging session. The worst-case number of new chronologies per data cluster for this trend template is in fact $8 = 2^{6/2}$ or $2^{(I-2)/2}$.

The space complexity of TrenDx arises from maintenance of the hypothesis structures and the temporal context tree. Due to beam search there are BMT hypotheses, each holding assignments that grow linearly with D. The temporal context tree stays at a constant breadth of BMT (roughly one branch of the tree per hypothesis) and worst case depth of DI, since every time CONSISTENT-TEMPORAL-CONTEXTS is called the tree may grow in depth by I. Hence the overall space complexity is BMTDI.

4.8 Optimizations of Temporal Reasoning for Monitoring

Initial artificial intelligence literature describing implemented temporal reasoning systems generally referred to the temporal reasoner as a "special sort of database management system" [Dean and McDermott 1987, page 5] to tell and ask temporal queries, and as "meant to be an accessory program for an expert system and not one of the primary computational activities." [Kohane 1987, page 38] This "black box utility" view of temporal reasoners is quite unrealistic for monitoring systems such as TrenDx, as well as for knowledge-based planners that also must manage temporal uncertainty. More recent writings [Williamson and Hanks 1993] have noted that temporal reasoners must provide a broad interface allowing users to manage their temporal relations. They also recommend that a user thinks about his or her problem structure to best use temporal reasoners.

This research heartily espouses both those latter views. Explicitly extending and pruning the temporal context tree is a critical portion of the TrenDx matcher. In terms of time complexity, when the temporal context tree for trend template chronologies is more than a few levels deep, as often happens, the time spent on answering temporal queries between points dominates all other TrenDx processing time.[9] TrenDx reduces the number of temporal queries and the time spent per query using means *outside* the temporal reasoner.

9. Time trials with TrenDx matching the oxygen handbagging data set to hemodynamic intensive care unit data revealed that over 95% of processing time was spent querying TUP for temporal distances between points. While use of a different implementation or temporal reasoner in TrenDx may have yielded a different percentage, the time complexity would still remain.

4.8.1 Reducing Queries to the Temporal Reasoner

Recall that the only temporal query requiring search in the temporal reasoner is: within a temporal context, what are the possible relations between a calendar time point *pt* (the time of a datum) and a trend template interval *I*? The result is a non-empty subset of {before, within, after}. The most time intensive way of obtaining the answer is to compute via exhaustive search the ranges of temporal distances between *pt* and begin(*I*) and between *pt* and end(*I*), and use these ranges to find the possible relations.

TrenDx employs two methods of determining relations between *pt* and *I* without explicit temporal search:

1. TrenDx pre-compiles temporal relations between the trend template intervals before any data or processes are defined. The possible interval relations have been enumerated by Allen [Allen 1983]. In matching a datum *d* to an interval *int2*, if TrenDx knows that *int2* is after *int*1, and *d* was before *int1*, then automatically TrenDx can deduce that *d* is before *int2*. TrenDx requires no additional queries to the temporal reasoner.
2. TrenDx memoizes temporal relations between previously processed data clusters and trend template intervals. If TrenDx has remembered that time t_0 was after *int*, and the current data cluster is at time t_1, later than t_0, TrenDx determines quickly that t_1 is after *int*.

4.8.2 Reducing Time Spent Per Query

In the previous section we noted that the temporal context tree has O(ID) worst case depth. Due to internal bookkeeping during search, a temporal reasoning program may spend more time computing distances for a deeper context tree.[10] This time per query is reduced by compressing the context tree after hypotheses are pruned.

Another saving in time per temporal query comes from rapid computation of temporal distances between two time-stamped data points. This difference is merely the difference in their machine-dependent internal times.

The next chapter demonstrates TrenDx matching clinical data to trend templates in two application areas: pediatric growth and intensive care unit monitoring. We illustrate the trend templates and the matching results, keeping in mind both clinical and computational performance measures.

10. For TUP we estimate search time to grow linearly with context tree depth.

5 TrenDx Results on Clinical Data

We have represented monitor sets of competing trend templates in two application areas: pediatric growth monitoring and intensive care unit monitoring. Small scale experiments matching clinical data to these monitor sets with TrenDx show the potential of the program for use in intelligent, context-sensitive monitors. The results also illustrate the trade-offs of efficiency versus soundness.

5.1 Pediatric Growth

We have represented two monitor sets for preliminary classification of children's growth based on pubertal development. The monitor sets, one monitor for girls and the other for boys, may be of use as screening tools for general pediatricians. Each monitor set consists of three trend templates:

1. *average normal growth*, where pubertal onset is within the times for the majority of children, and bone development is consistent with the child's age,
2. *early puberty*, where pubertal onset is earlier and bones develop at a quicker pace than average.
3. *constitutional delay*, where pubertal onset is later and bones develop at a slower pace than average.

These three growth trends are all variations of normal growth. Pattern (1) is the most typical and is assumed to describe a patient until another trend prevails. The primary treatment required for conditions (2) and (3) is to assure the child and the parents that in time the child will be similarly developed to his or her friends.

This monitor set is useful for several reasons. First and foremost, variations (2) and (3) of normal are not always considered by general pediatricians examining unusually tall or short patients. The pediatrician may consider such children to have endocrinological problems requiring a specialists' attention. Automatic labelling of the patient data as matching early puberty or constitutional delay may delay or obviate the need for a consult at a tertiary care center. Second, if TrenDx were to

suggest either pattern (2) or (3), the trend can be verified with inexpensive yet discriminatory tests:

- plotting of height and weight data on standards for populations of different pubertal onsets [Tanner and Davies 1985].
- measurements with a ruler of genitalia and breast development [Tanner 1990].
- bone age, found by an x-ray of the left wrist [Greulich and Pyle 1959].

Third, an automatic alert to constitutional delay or early puberty may be useful in merely alerting the general pediatrician that a patient needs to be more closely monitored. This closer monitoring may pay off, because in some cases patients showing a pattern of constitutional delay do in fact suffer from a more serious hormonal or skeletal disorder.

5.1.1 Trend Templates

The structure of the three trend templates in each monitor set were developed in conjunction with pediatric endocrinologists and through reading from specialized texts [Kaplan 1990; Tanner 1990]. The time ranges on landmark points and interval end points were estimated to include most of the variation for each population as published in [Tanner and Davies 1985]. The code that TrenDx interprets to build the growth trend templates is in Appendix A.

The trend template for male average normal growth appears below; it is an extension of the trend template in section 3.6.1.

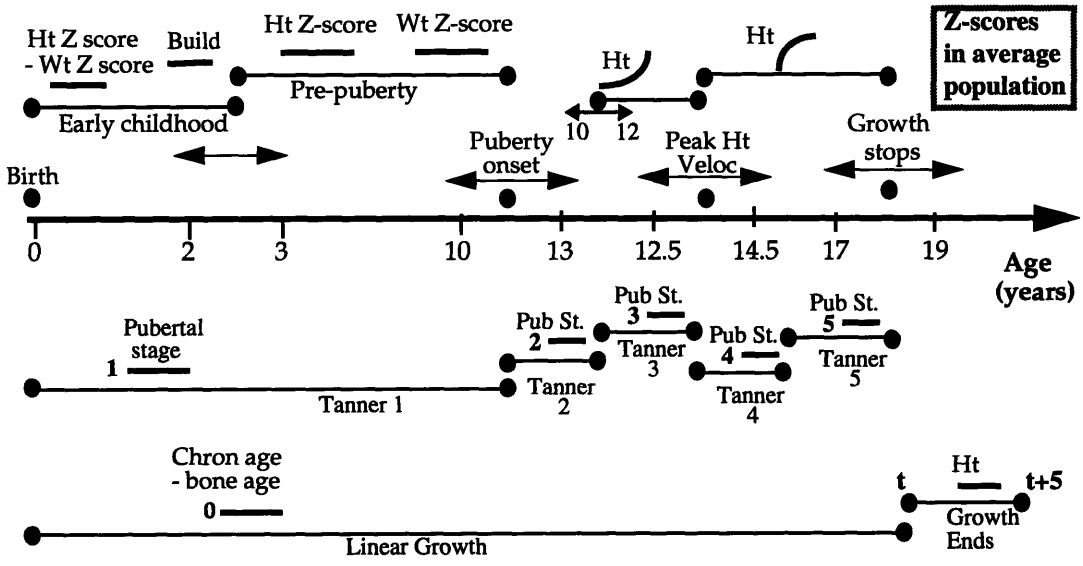


Figure 22 Trend template for male average normal growth.

This trend template includes four landmark points: `birth`, `puberty onset`, `peak height velocity`, and `growth stops`. `Puberty onset` occurs 10 to 13 years after `birth`, `peak height velocity` occurs 12.5 to 14.5 years after `birth`, and `growth stops` occurs 17 to 19 years after `birth`. The links between landmark points establish a conceptual time scale of the patient's age in years. This trend template includes eleven intervals, five of which constrain height or weight. The interval `early childhood` constrains that height and weight z-scores vary together (see section 3.6.1). It also constraints as constant a function of height and weight we call *build*, defined as

$$\text{build} = \text{weight} / (50\% \text{ weight at height age}). \quad \textbf{(EQ 13)}$$

This statistic is used by some pediatric endocrinologists as a measure of lightness or heaviness, and may be more consistent for children than the more commonly used body mass index [Garn, Leonard et al. 1986]. The interval `pre-puberty` constrains height and weight to stay at constant z-scores. The height and weight z-scores are with respect to the population of boys with average pubertal onset time, found in [Tanner and Davies 1985]. Two consecutive phases represent the pubertal growth spurt. During this growth spurt height rises sigmoidally; it is useful to divide this height variation into two quadratic phases. Height first rises quadratically with positive first and second derivatives, then rises quadratically with negative second derivative until growth stops. The first of these phases begins 10 to 12 years after `birth`. The transition point between these phases is by definition the peak height velocity, an important growth landmark to monitor. An interval after `growth ends` constrains height to be constant for five years after growth stops.[1]

Other intervals in this trend template constrain other patient parameters. Five consecutive intervals represent the five Tanner stages of pubertal development. Over each phase, the pubertal stage is constant at some value between 1 and 5. The transition between the first two of these phases is equal in time to `puberty onset`. The interval `linear growth` represents the entire period of increasing height. During this phase chronological age is restricted to be equal to bone age.

The trend templates for constitutional delay and early puberty are identical in landmark point and interval structure to that for normal growth, yet differ in their time ranges for the pubertal and post-pubertal landmarks and interval points. These time ranges were determined from the clinical standards for boys and girls with delayed and advanced puberty in [Tanner and Davies 1985]. Most of the detail of the constitutional delay template is illustrated in the figure below. Aside from the different time ranges, the distinctions between this trend template and that for average normal growth are:

1. In fact, height is constant for longer than five years. However, this trend template does not aim to model life beyond childhood.

- height and weight z-scores are with respect to the population of boys with delayed pubertal onset time.
- the constraints on bone age.

Bone age is constrained to be two years behind chronological age in the `pre-puberty` interval, one year delayed in the first growth spurt phase, and equal to bone age during the second growth spurt phase.

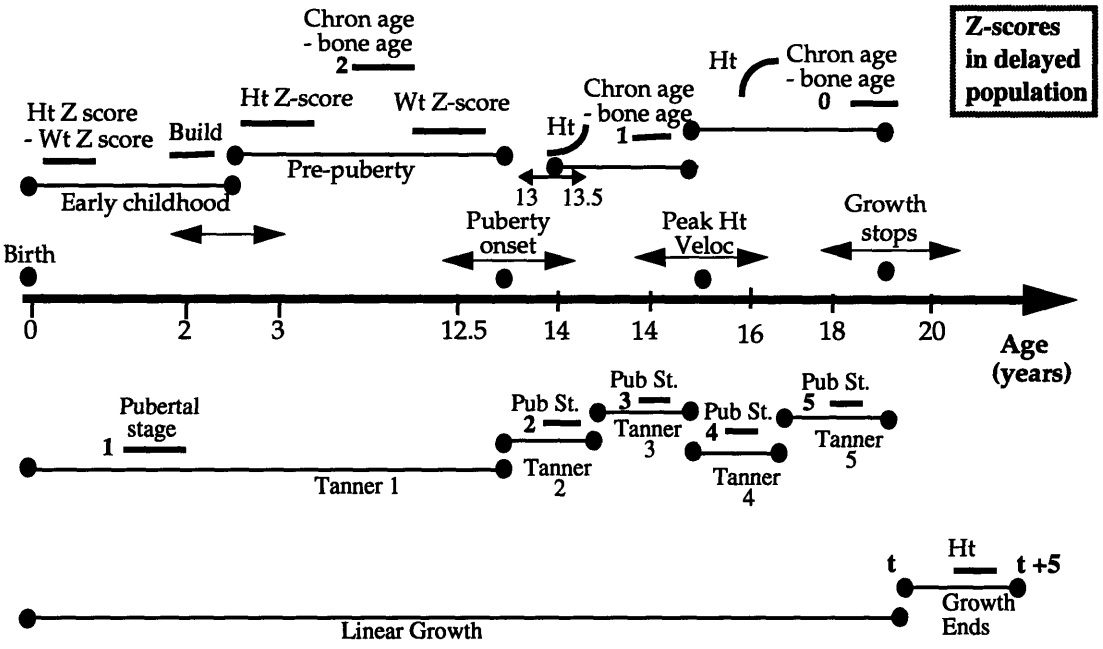


Figure 23 Trend template for male constitutional delay.

Similarly, the male early puberty trend template has time constraints that are generally earlier in the pubertal and post-pubertal regions. The male early puberty trend template uses height and weight z-scores for a population of boys with early puberty, and constrains bone age to be ahead of chronological age first by 2 years, then 1 year, then 0 years, in the same intervals as Figure 23.

The monitor set for girls' growth girls is very similar to that for boys, differing in times related to puberty onset and thereafter, and in using height and weight z-scores from female sub-populations. Appendix A fully specifies this monitor set.

5.1.2 Method

We tested TrenDx matching the growth monitor sets to the data of thirty consecutive growth records from the pediatric endocrinology clinic of Boston Children's Hospital. These cases were originally used in an experiment on the efficacy of a constraint-based version of TrenDx described in section 7.1.5. The cases were

biased toward abnormal patients due to the source of the data: 4 were normal, 10 had constitutional delay, 3 had early puberty, and 10 had growth hormone deficiency. These diagnoses were made by a pediatric endocrinologist examining the same data as TrenDx. The number of visits varied from two to over a dozen. Most visits consisted of only a height and a weight. Most data for abnormal patients included at least one visit with a bone age or a pubertal stage measurement.

TrenDx matched patient data to the monitor sets using a beam width of 4 hypotheses per trend template and a prediction granularity of 3 months. TrenDx produced for each patient at each visit a set of hypotheses for each of the three trend templates in the gender-specific monitor set. We considered the score of the best matching hypothesis for each trend template and plotted these best scores over time, as in section 4.6.

Each patient was presumed at first to match the trend for average normal growth until one of the other two patterns was triggered. We tested three mechanisms for triggering a pattern P (either constitutional delay or early puberty):

1. *Persistent gap:* For two consecutive visits, the best hypothesis for P scored 0.8 or less times the best hypothesis for average normal growth.
2. *Single gap:* For one visit, the best hypothesis for P scored 0.6 or less times the best hypothesis for average normal growth.
3. *Disjunction:* Either persistent gap or single gap.

After all 30 patients were run, we checked for the program's sensitivity and specificity [Sox, Blatt et al. 1988, pp. 72-74]in triggering each trend pattern P:

sensitivity for P = (Number with P and TrenDx triggers P) / (Number with P)

specificity for P = (Number without P and TrenDx does not trigger P) / (Number without P)

For the growth hormone deficiency patients, we considered a TrenDx alarm for constitutional delay to be correct.

A second trial tested TrenDx performance on 20 patients selected randomly from the paper records of a Boston-area general pediatrician. The general pediatrician noted no abnormal disorders in any of the patient records. Nearly all the records included primarily heights and weights; a few also included pubertal stages. The number of visits ranged from 2 to 20. For the purposes of this trial we presumed all 20 patients to have average normal growth, even though the data were not reviewed by any pediatric endocrinologists. TrenDx matched these patients' data to the growth monitor sets, and alarms were emulated as in the first trial. Sensitivities and specificities were then recalculated for all 50 patients.

5.1.3 Results

Below are growth data from a patient with constitutional delay from the first trial on 30 patients. This patient was described in section 2.1.3. The TrenDx matching results on this patient are in Figure 25.

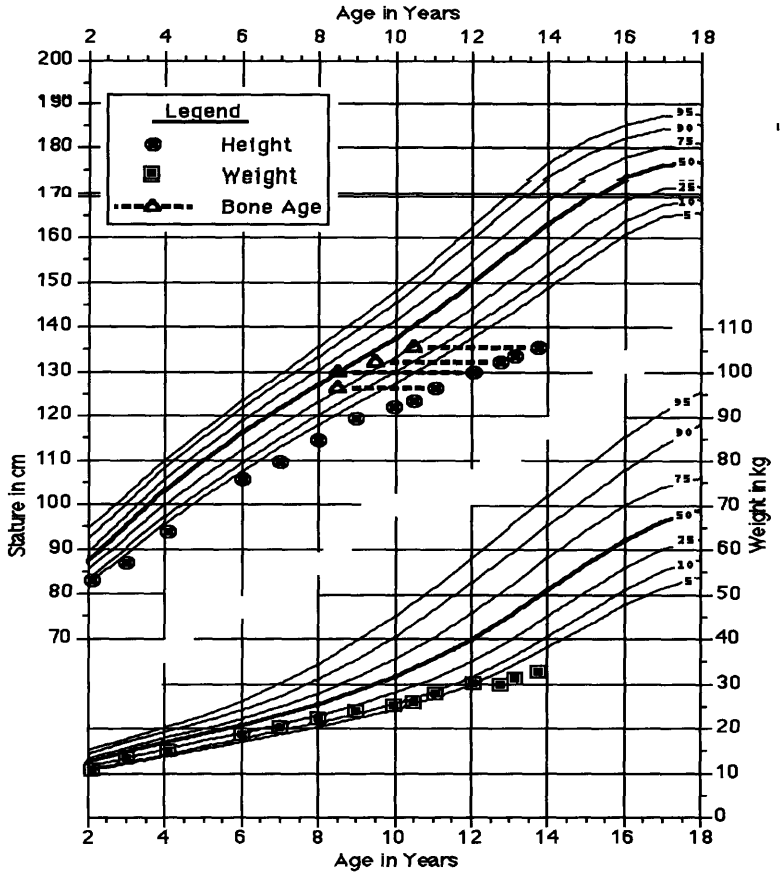


Figure 24 Growth data of constitutional delay patient.

TrenDx finds the constitutional delay trend template to match best to the patient from ages four through nine, where the height z-scores decrease steadily on the NCHS charts as well as on the standards for children with average pubertal onset. The scores for ages four and six are enough to trigger an alarm using the persistent gap rule. Indeed, this age range is characteristic for when constitutional delay [Kaplan 1990, page 49]. Another widening gap in TrenDx's favoring of constitutional delay begins at age eleven, when the patient had the first of four bone ages delayed by more than two years. Using the single gap rule, average normal growth cannot be overruled in this patient, perhaps suggesting that the multiplicative factor of 0.6 lower score than for average normal growth was too low.

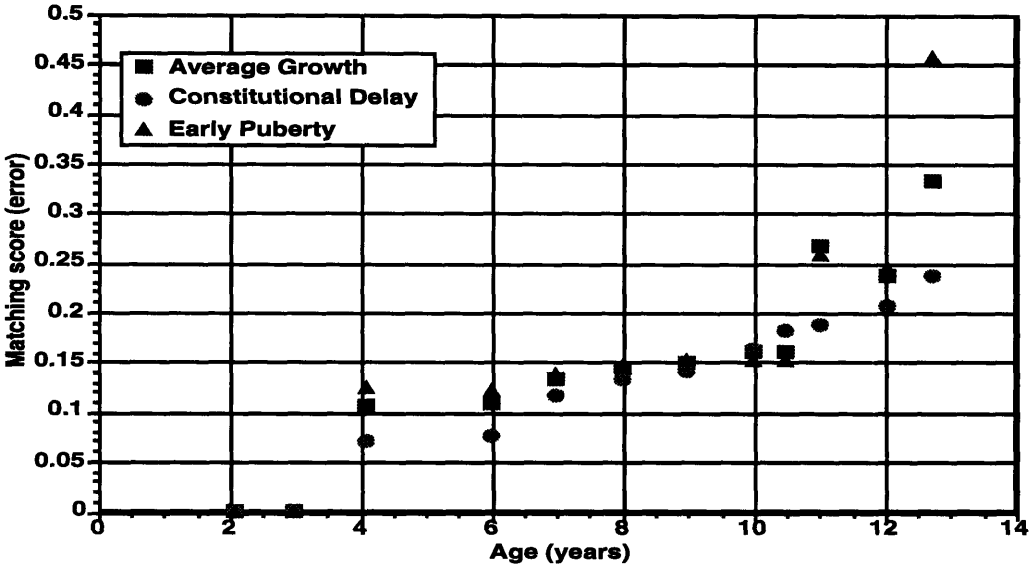


Figure 25 TrenDx results on constitutional delay patient.

The table below shows results for the thirty patients from the endocrinology clinic. As might be expected, sensitivity for either trend was a maximum under the union trigger rule, which allowed more triggering opportunities. Specificity for both trends was highest using the persistent gap rule that insisted on two consecutive visits with discriminating scores. However, specificity was but minimally diminished when using the union triggering rule.

		Persistent Gap			Single Gap			Union		
Disorder	No.	None	CD	Early	None	CD	Early	None	CD	Early
Normal	4	**4**	0	0	**3**	0	1	**3**	0	1
Const. Delay	10	7	**3**	0	5	**5**	1	4	**6**	1
Early Puberty	3	3	0	**0**	1	0	**2**	1	0	**2**
GH Deficiency	13	4	**6**	4	5	**6**	3	4	**6**	4
Cumulative Sensitivity	**30**		**.39**	**.00**		**.48**	**.66**		**.52**	**.66**
Cumulative Specificity	**30**		**1.00**	**.85**		**1.00**	**.81**		**1.00**	**.77**

TABLE 1 TrenDx matching results on tertiary care patients.

The sensitivity of TrenDx on these patients was diminished by two effects innate to regression techniques. First, in ten of the thirty patients there were but

two visits, most often producing trivial regression fits that could not distinguish the three trend templates. Experts were in fact able to distinguish based on two visits, particularly when one visit included a bone age or pubertal stage measurement. Second, the regression match smoothed the effects of isolated bone ages or pubertal stage with the less discriminatory effect of the frequent heights and weights.

In the figure below are growth data for one of the general pediatrics patients. TrenDx matching results are in Figure 27.

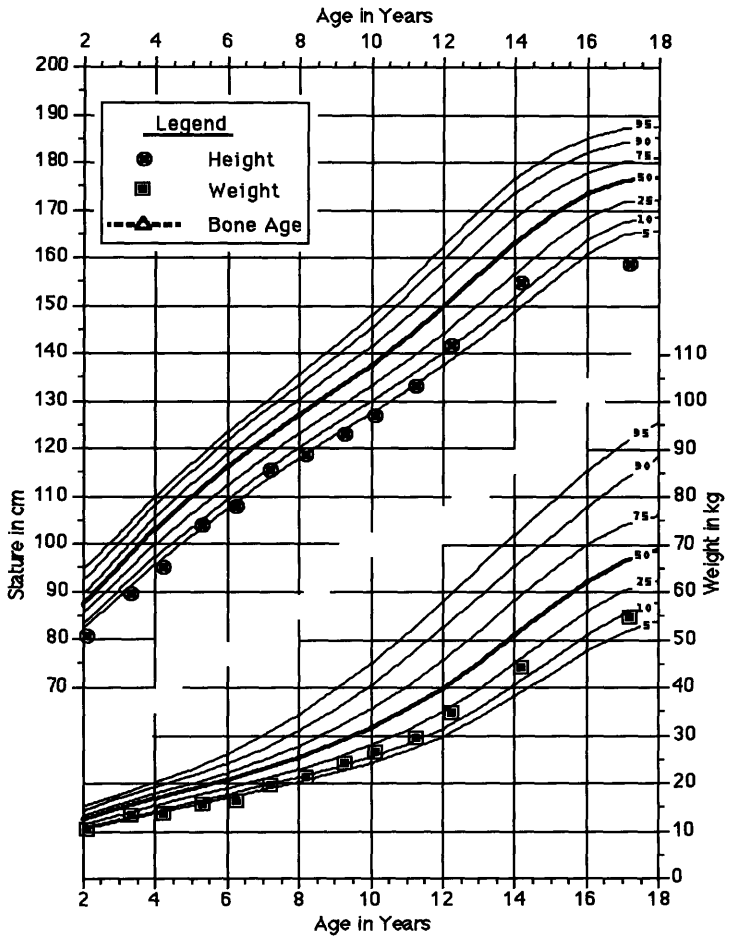


Figure 26 Growth patient from general pediatrician, diagnosed normal.

This boy's growth data consisted of thirteen postnatal visits consisting of only heights and weights. Throughout his life the patient stayed close to the same height and weight centiles on the NCHS charts. The TrenDx matching reveals that the average normal growth template was not overridden according to either trigger rule. Neither of the other two trends is ever substantially lower in score.

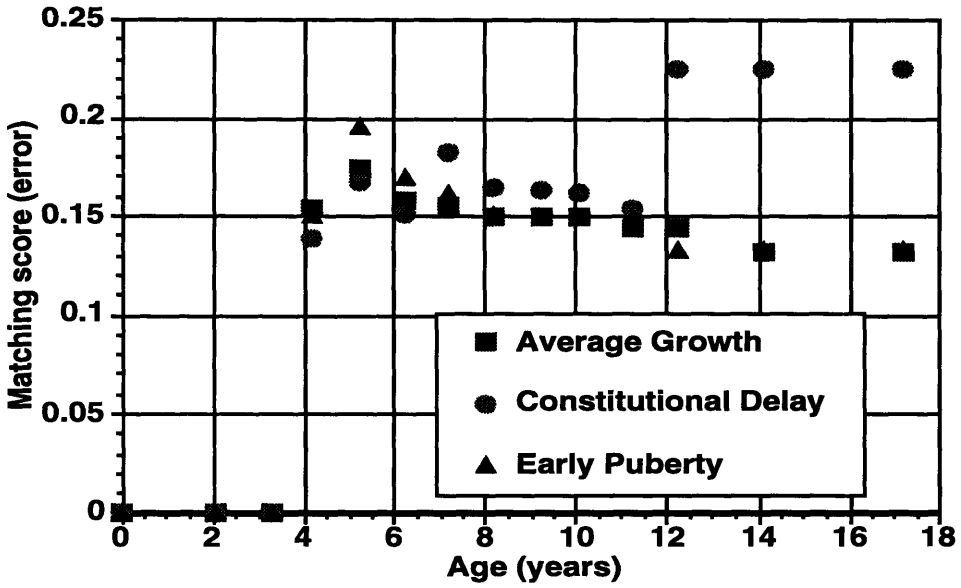


Figure 27 TrenDx results on general pediatrics patient.

The table below shows that, depending on the triggering scheme, TrenDx did not trigger an alarm in 70% to 80% of the general pediatric patients.

		Persistent Gap			Single Gap			Union		
Disorder	No.	None	CD	Early	None	CD	Early	None	CD	Early
Extra "Normals"	20	16	0	3	15	1	4	14	1	6
Cumulative Sensitivity	50		.39	.00		.48	.66		.52	.66
Cumulative Specificity	50		1.00	.85		.96	.81		.96	.75

TABLE 2 Updated TrenDx results incorporating general pediatric patients.

If we group these twenty patients with the four normals from the tertiary care clinic, the cumulative specificity results are near 100% for constitutional delay and near 80% for early puberty. However, in light of the distinct method of labeling these patients as normal, we caution too strong a conclusion.

In nearly half of these general pediatric patients the trend for early puberty was either triggered or scored slightly better than the normal average growth template for much of the lifetime of the child. One reason for this is that some of these children may in fact be better characterized as having early puberty, say by a pediatric endocrinologist. Another is that American children in recent decades, particularly

those from middle to upper-class neighborhoods as were these patients, may generally have earlier puberty than children in the NCHS study from the 1950's.

For some normal patients where alternate trends were not triggered, the average normal growth template was not the clear best match, but instead was closely matched by an alternate trend. Perhaps height z-scores in different puberty onset populations are not discriminatory enough to distinguish between these trends.

5.1.4 Discussion

The above experiments are inconclusive as to whether TrenDx is useful as a screening tool for pediatric growth monitoring. The sample sizes for normals and abnormals were too small for statistical significance. The different sources of patient diagnoses (some pediatric endocrinologists, some general pediatricians) provided a case set that was not uniform. To remedy these problems, we plan to conduct a larger scale growth experiment on hundreds of patients in which the results of different versions of TrenDx are compared to diagnostic monitoring of pediatric endocrinologists, general pediatricians, and medical students. This future experiment is outlined in section 9.1.

Even if the experimental methods were sound, the monitors need some improvement before being useful for clinical practice. The constitutional delay monitors had a high enough specificity (from 96% to 100%) so that its sensitivity (near 50%) may warrant consideration. However, the early puberty monitors still need an improvement in specificity (near 80%). We would like higher sensitivities, if possible, for both these monitors. Consequently we need to explore means of improving the monitor sets for pediatric growth.

5.1.5 How to Improve Monitoring Performance

Some improvements to monitoring performance may come from a different modeling strategy. Perhaps TrenDx would diagnose pediatric growth trends more accurately by matching to trend templates that monitor the past five years of data. Maybe TrenDx could distinguish between competing growth trends that incorporate quantitative estimates into the value constraint polynomials.

Other improvements can come from recognizing generally applicable properties of TrenDx matching to sparse data. First, in order to insure reliable value constraint matches, TrenDx should assign at least three or four data points per trend template interval. When modeling trends in sparse data sets, a knowledge engineer should minimize the number of disjoint intervals constraining a particular parameter. If trend templates contain too many such intervals, TrenDx may find a low-scoring or trivial match by assigning only one or two data points to each interval. Changing the early puberty trend template to remove disjoint intervals would likely raise the specificity of the early puberty monitors.

Second, TrenDx smooths out the diagnostic effect of measurements such as bone age or pubertal stage that are measured rarely yet which should have more discriminatory power than the more frequently sampled height and weight. Thus, either clinicians need to measure the sparser parameters more frequently, or a monitor builder may need to ensure that the more discriminatory measurements are given appropriate weight. Recall that the matching score of a hypothesis is a weighted average of the matching scores for each value constraint (section 4.4.2). Value constraints may be weighted by diagnostic import, and a knowledge engineer can assign higher weights to bone age and pubertal stage data. Higher sensitivities for the fault growth monitors should result.

Third, when the sequence of variables $\{Y_t\}$ for a particular value constraint includes values near zero, sizable errors in this value constraint can inflate the hypothesis match score. This is a result of using mean absolute percentage error as a matching score. This was the case for some growth patients with height or weight z-scores near zero but varying substantially. Careful statistical modeling or scaling of the $\{Y_t\}$ can reduce this effect.

5.2 Intensive Care Unit

We have built a monitor set representing two competing trends in response to 100% oxygen handbagging for an intensive care unit patient. We then evaluated TrenDx matching this monitor set to data during faulty handbagging of an eight month old girl in the Boston Children's Hospital multidisciplinary intensive care unit. We also examined responses of TrenDx performance on the same task to changes in the beam size and the prediction granularity. This evaluation was intended not as a clinical trial but as an illustration of TrenDx's potential to diagnose trends in high frequency, multiparametric data. We also varied beam size and prediction granularity in matching to the same data set to better understand performance trade-offs.

5.2.1 Trend Templates

The handbagging monitor set consists of two trend templates: adequate handbagging and hemodynamic fault. These templates describe patterns in four parameters: fraction of inspired oxygen (FIO_2), percentage oxygen saturation of hemoglobin (O_2 Sat), mean arterial blood pressure (BP) and heart rate measured by electrocardiogram (HR). The hemodynamic fault trend template was described in detail in section 3.4 on page 38. The trend template for adequate handbagging contains the same landmark points and the same response intervals for FIO_2 and O_2 Sat. The adequate handbagging trend template differs in its trends for BP and HR, both of which are constrained to be constant in a single interval whose length is the handbagging period.

The trend templates are instantiated by anchoring the landmark point `Handbag On` to the time of a special datum generated by a switch on the ventilator. The switch signals that the ventilator no longer supplies oxygen. In principle instantiation could happen via the results of a strong match to a preliminary trend template.

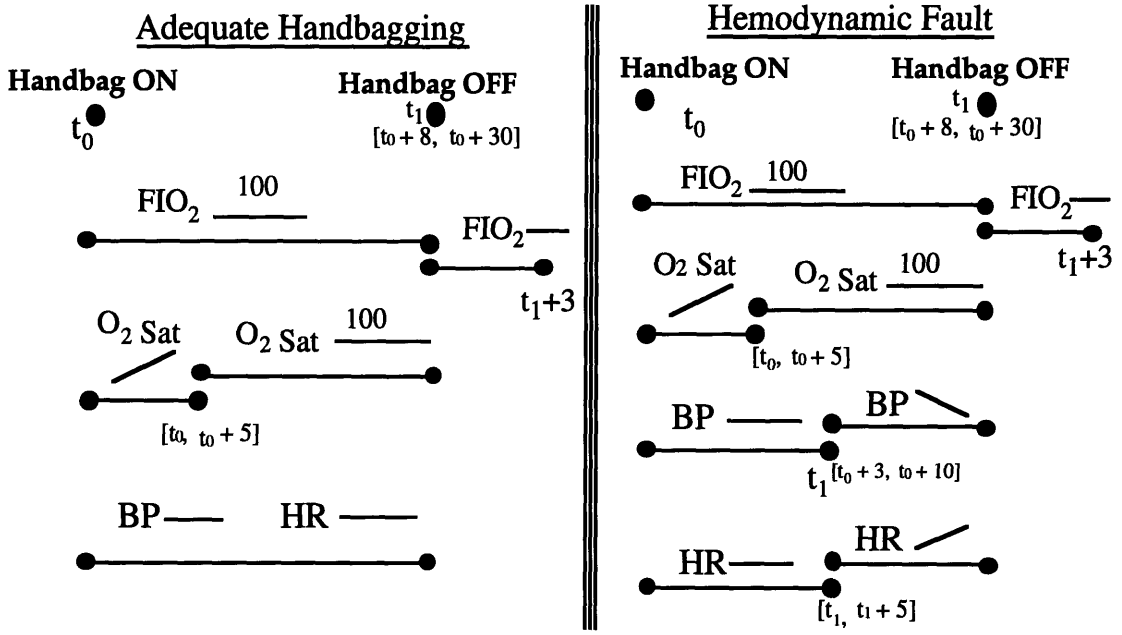


Figure 28 Monitor set for oxygen handbagging.

5.2.2 Method

TrenDx matched the monitor set in Figure 28 to four signals of data between 12:22 and 12:31 a.m. during which the patient received oxygen via a handbag. Several intensive care unit physicians agreed the patient was experiencing some hemodynamic fault during this period. TrenDx also matched the same monitor set to five other periods of handbagging in this patient during the same day. The results were similar enough that we only show results of matching to the first handbagging session.

TrenDx matched the nine minutes of data to the above monitor set using a prediction grain of 0 and a beam size varied from 1 to 4. We then matched TrenDx to the same data using a prediction grain of one minute and a beam size of 4. For each run of TrenDx we noted the scores of the best matching hypotheses over time, as well as the time estimates for uncertain interval endpoints.

5.2.3 Results

Figure 29 shows the four signals of ICU data and results of TrenDx matching to this data using a beam size of 4 and a prediction grain of 0. Time axes are the same.

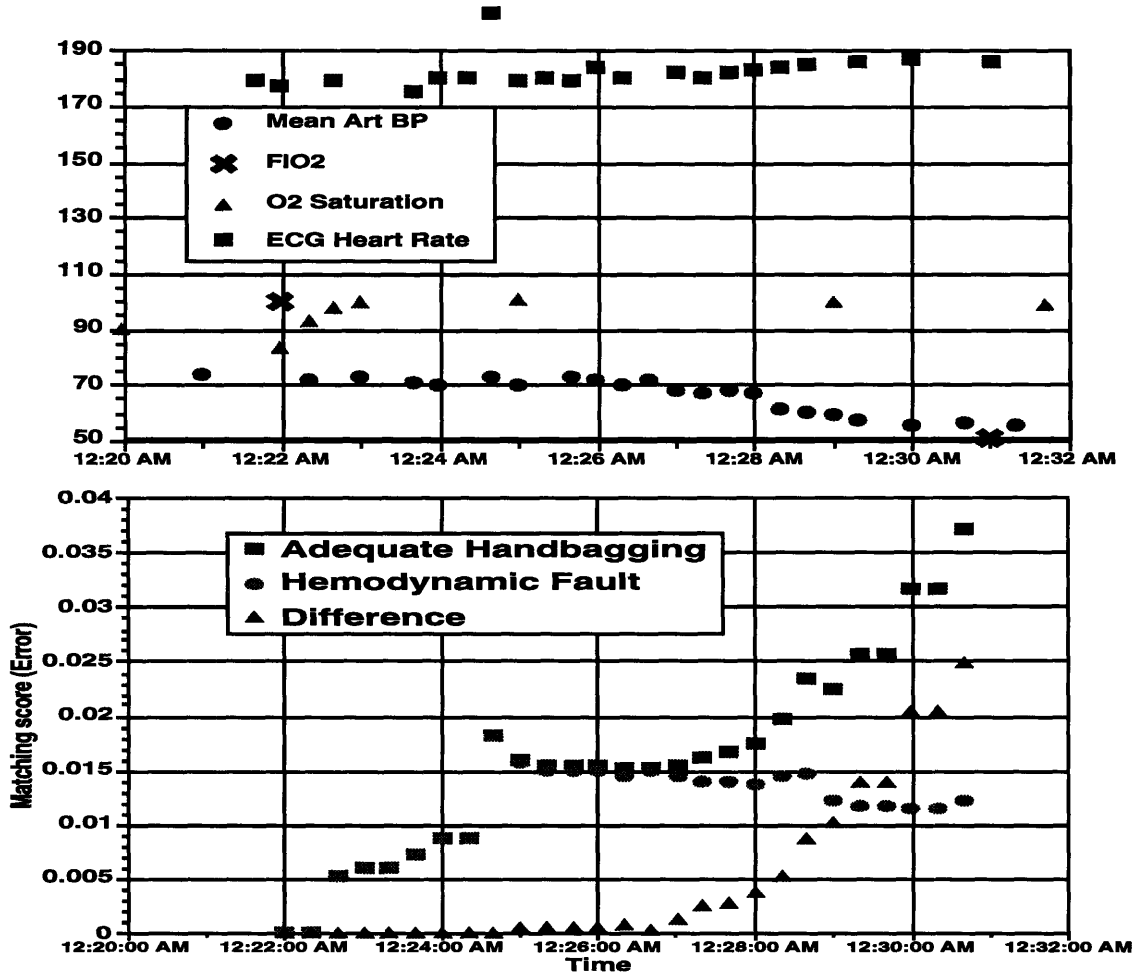


Figure 29 TrenDx results on ICU patient data. beam size = 4; grain = 0.

Note that the outlying heart rate between 12:24 a.m. and 12:25 a.m. caused a jump in the match scores to both trend templates. Had that outlier been removed or smoothed by filtering, the error scores for both trends would have remained lower.

The best matches of each trend template stay close in score until 12:27:21 a.m., when the percentage error for the adequate handbagging trend template rises while that for the compromised venous return stays level. The difference between these two, plotted at the bottom of the graph, rises steadily for the duration of the handbagging episode. This difference may be used as a means for sending an alarm. A monitor rule could check if the difference crosses a threshold, or if it steadily rises for a pre-specified duration.

Increasing the beam size from 1 to 4 widened the gap between the two hypotheses, as illustrated in Figure 30. For this monitor set, the best scores for adequate handbagging remained unchanged, and the best scores for compromised venous return decreased. The best hypothesis score for beam sizes 3 and 4 decreased only slightly over the best score for beam size 2.

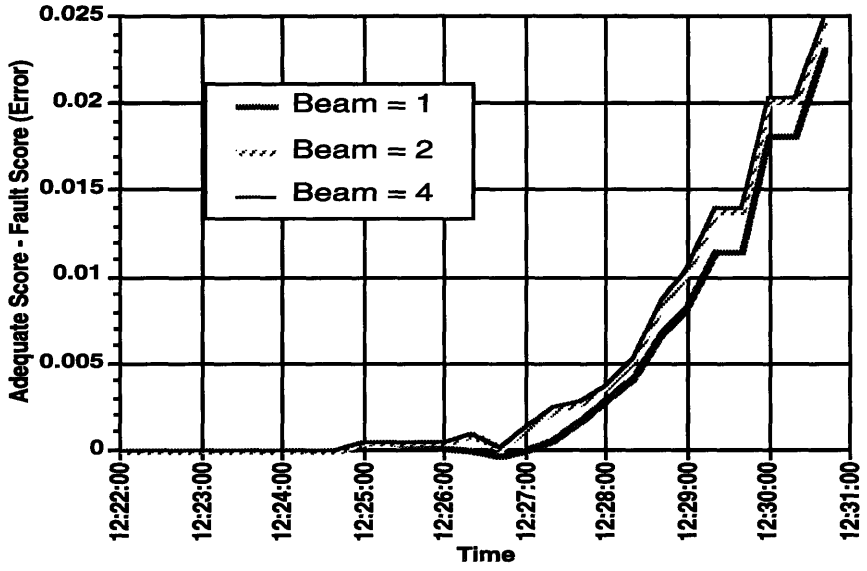


Figure 30 TrenDx comparative results on ICU data, beam sizes 1, 2, and 4.

Increasing beam size impacted how TrenDx estimated the phase transitions in the compromised venous return template. We noted how the best compromised venous return hypothesis estimated the three transition times in its trend template (see Figure 28) after TrenDx processed all the data. Figure 31 shows that for a beam size of four, this best hypothesis estimated that O_2 Sat began being steady between 12:22:41 and 12:23:01, that BP began decreasing between 12:25:01 and 12:25:41, and that HR began increasing between 12:27:41 and 12:28:01. A beam size of three produced the same estimates. However, a beam size of 2 pushed back the best estimate of the start of HR increasing two minutes, to between 12:25:41 and 12:26:01. As shown in Figure 31, this estimate seems premature compared to that for a beam size of 4

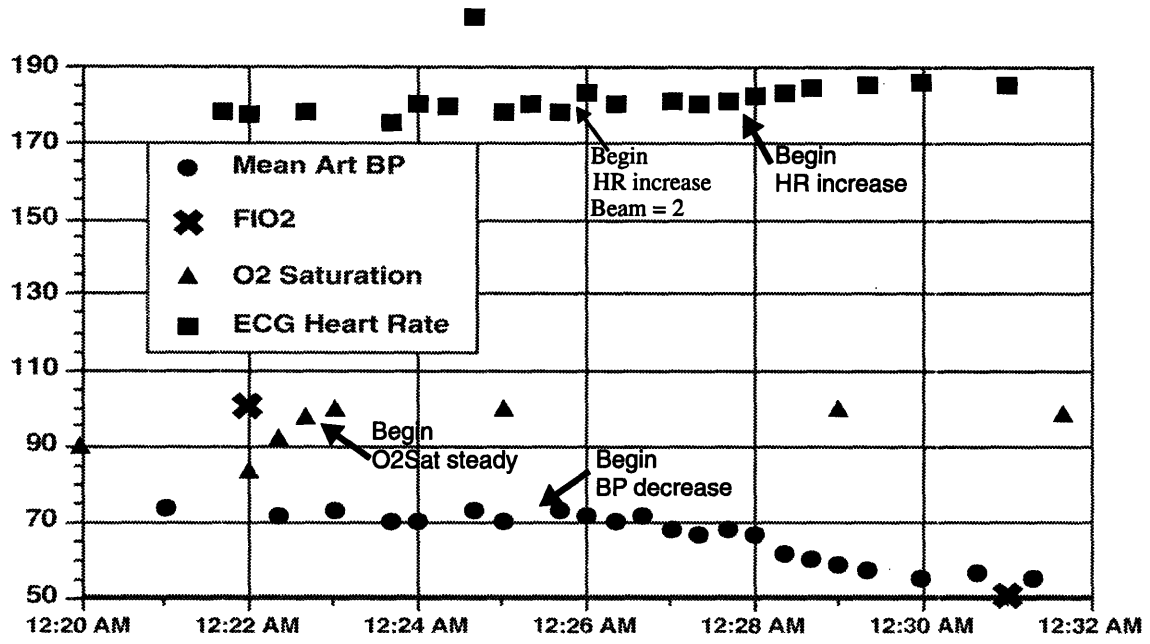


Figure 31 TrenDx reports of phase transitions in best scoring hemodynamic fault hypothesis with beam size = 4.

Increasing the TrenDx prediction granularity from zero to one minute meant that the program treated one minute's worth of data, or three time slices, as if it occurred simultaneously. This coarser processing resulted primarily in significant time savings: temporal queries and the depth of the temporal context tree were reduced roughly by a factor of three. Consequently, total processing time (see section 4.7) was nearly as low as the cube root of that with zero granularity.

As shown in Figure 32, with coarser granularity TrenDx discriminates similarly between the two trend templates as with zero granularity. The discrimination, as shown by the plot of "difference," is slightly less in magnitude and is delayed in time as compared with Figure 29. For example, with one minute granularity an increasing difference in hypotheses can be detected by 12:29:01, whereas with zero granularity this is detectable by 12:27:21.

The time estimates for transition points using one minute granularity are slightly different than with zero granularity. With the coarser grain, the best scoring compromised venous return hypothesis estimated that O_2 Sat began its steady phase between 12:23:01 and 12:25:01 (later as compared to zero grain), that BP began decreasing between 12:25:01 and 12:25:41 (same), and that HR began decreasing between 12:27:01 and 12:27:21 (earlier). Compare with Figure 31. These time estimates have an uncertainty of two minutes, double the granularity.

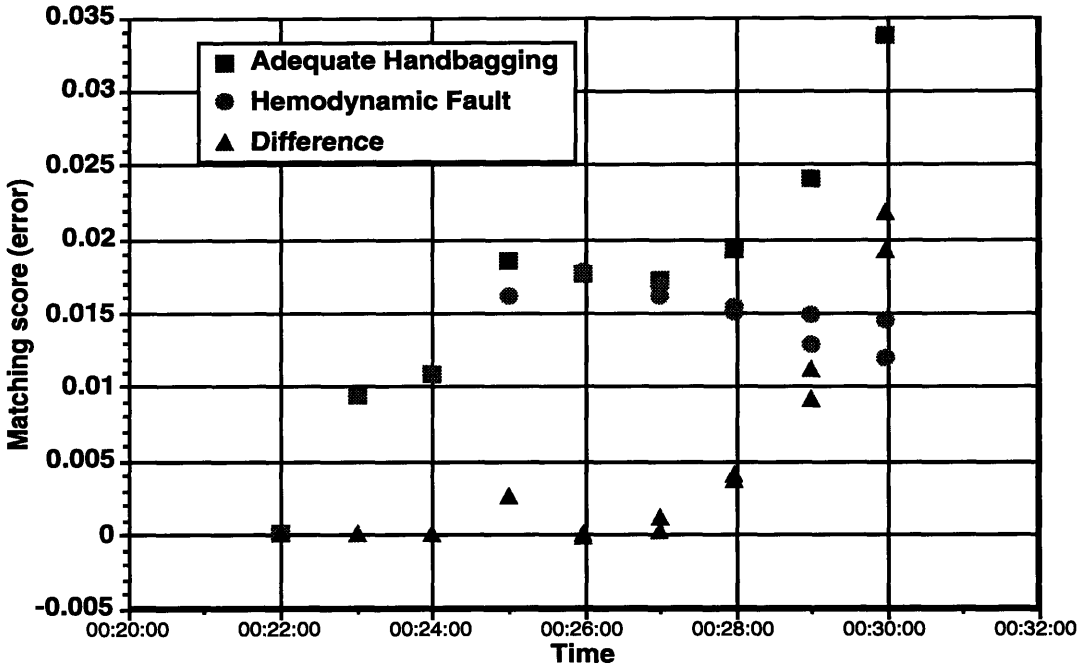


Figure 32 TrenDx results with prediction grain of 1 minute and beam size of 4.

5.3 Discussion

We cannot extrapolate the results of these small experiments on one patient to an argument for TrenDx's utility in detecting trends in ICU data. However, these limited results do suggest that low-degree polynomial regression models may be useful for characterizing trends in high-frequency data. Furthermore, by temporally linking trends of multiple variables, TrenDx can discriminate at least some clinically significant patterns. More experimentation is required, fitting larger patient populations to a variety of monitor sets describing different treatments and interventions.

The apparent discriminatory power of TrenDx shown in Figure 29 and Figure 32 must be considered in light of the value constraint experiments of section 4.4.6. In particular, we have observed on this data set that a linear value constraint matches linear decreasing data (e.g. the blood pressures) much better than a constant value constraint. However, the constant value constraint would match constant data *only slightly better* than a linear value constraint. Thus if TrenDx matched the same monitor set to data that seemed like adequate handbagging, the match to the compromised venous return trend template would probably be just a little poorer.[2] Had we supplied quantitative estimates of the polynomial coeffi-

cients (e.g. numerical slopes of the decreasing and increasing lines), TrenDx could discriminate better on such data.

The time and space savings from increasing the prediction granularity, or from decreasing beam size, may be important should the application require the monitors to track the data in real-time. As shown in this case, the cost of these savings may be a delayed monitor response to a clear clinical trend. This trade-off is better discussed in light of the how an overall monitoring system makes use of the TrenDx output. The next chapter presents a framework for a complete monitoring system.

2. Unfortunately, we were unable to acquire such an adequate handbagging case.

6 Completing the Monitoring System

The proceeding two chapters have described how TrenDx scores the match of process data to competing trend templates, and have illustrated matching to clinical data. In this chapter we discuss different methods for integrating TrenDx within an intelligent monitoring system.

6.1 Detecting When a Trend is Significant

Below we reproduce a figure from chapter 5, illustrating TrenDx results in matching intensive care unit data to two competing trend templates.

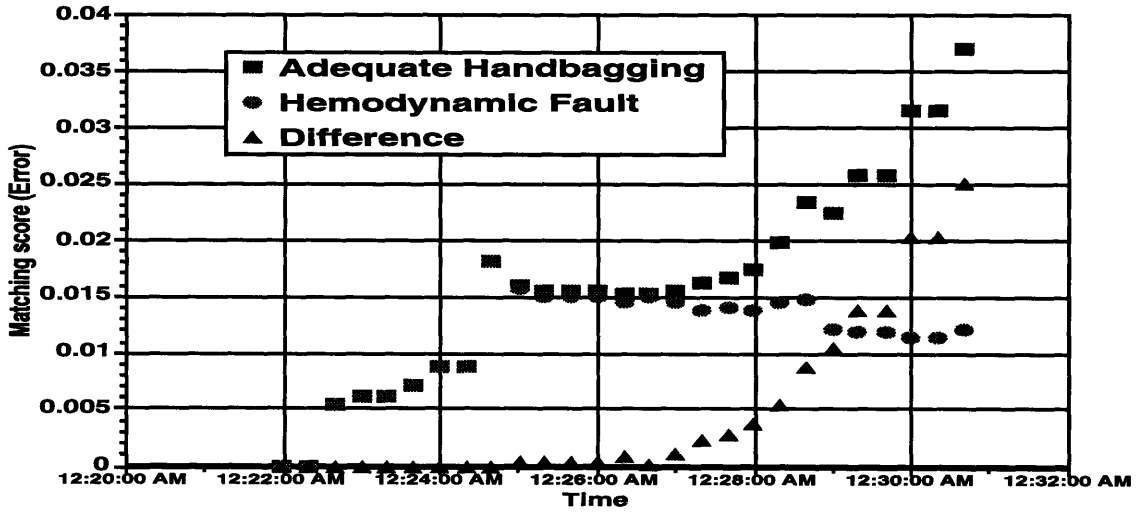


Figure 33 TrenDx results on ICU patient data.

Generally, TrenDx matches to a monitor set by computing for each time slice of data the best matching score for the normal trend template and for each competing fault. Throughout this chapter we denote the sequence of best scores for the normal template $\{TT_n\}$ and the sequence for the *ith* fault template $\{TT_{fi}\}$. High values in these sequences indicate a poor match to data. We denote by $\{TT_n - TT_{fi}\}$ the

sequence of score differences some fault model and the normal model. High values in this sequence indicate that the fault model matches better than the normal model.

We now must devise a scheme for answering the following questions:

- When has the normal model become a significantly poor match to the data to require attention?

  The answer is a property of the sequence $\{TT_n\}$.

- When is a fault model a significantly better match to the data than the normal model?

  The answer is a property of the sequence $\{TT_n - TT_{fi}\}$.

There are many possible means to answer these questions. In fact, trying to answer them uncovers another monitoring problem. Notice that each of the TrenDx result plots in Figure 33 is in fact a time-ordered sequence of data, and we wish to detect particular types of trends in these sequences. Does this mean we are back to square one of diagnostic monitoring? Not really, for we have significantly reduced the complexity of our monitoring task in two respects, First, we no longer must represent the *relevant diagnostic categories*, which requires extensive domain knowledge. Instead, we must represent the *relevant monitoring strategies*, which depend less on the progress of disorders but more on the monitoring environment. Second, we have also reduced the dimension of the relevant temporal patterns we seek. No longer are we finding multivariate patterns measurements, but instead we seek univariate patterns in either the time-sequence of best scores $\{TT_n\}$, or the time-sequence of difference in best scores $\{TT_n - TT_{fi}\}$, for some fault model.

In principle, we may use any univariate trend detection scheme available, including TrenDx. We illustrate a few straightforward methods, based on thresholds and accumulation of differences over time.

6.1.1 Thresholds

The simplest method of determining whether the score for of $\{TT_n\}$ or $\{TT_n - TT_{fi}\}$ is significant is establishing a threshold over which the matching score triggers an alarm. A threshold for a high score of $\{TT_n\}$, denoted by TT_n^*, is best determined by experience from training cases. Let $\overline{TTn}$ denote the average value of $\{TT_n\}$ over time for a single normal case. We presume $\overline{TTn}$ is normally distributed.[1] We recommend a two-stage supervised learning procedure. In the first stage, run a set of normal cases through TrenDx matching to the normal trend template

1. This presumption is based on intuition rather than a statistical result. Because $\{TT_n\}$ is non-negative, there may be some skew. With sufficient cases on can test an empirical distribution for normality [Sachs 1984, page 322].

and compute unbiased estimates of the mean and standard deviation of $\overline{TTn}$.[2] In the second stage, run both normal and abnormal cases through TrenDx using different levels of TT_n* for sounding an alarm. Select a value of TT_n* producing satisfactory sensitivity and specificity. If the normality assumption was a close approximation, then choosing TT_n* as two estimated standard deviations above the mean of $\overline{TTn}$ will yield a sensitivity of better than 0.95.

A significant threshold for $\{TT_n - TT_{fi}\}$ may be learned similarly: first estimate the distribution using faulty cases, then optimize sensitivity and specificity from normal and faulty cases. One may also find a threshold by relying on intuition, based on a TrenDx score giving a mean percentage error in explaining the data. $\{TT_n - TT_{fi}\}$ being above a threshold p means that the normal trend template is p percent more erroneous than the fault model. One may also consider judging a faulty trend as significant using a threshold for the *percentage better match* of the fault model: $\frac{TT_n - TT_{fi}}{TT_n} = 1 - \frac{TT_{fi}}{TT_n}$. In our growth experiment of section 5.1, we triggered alarms when this ratio exceeded 0.4 for a single time point.

6.1.2 Accumulators

More reliable sensitivity and specificity may be achieved by accumulating features of $\{TT_n\}$ or $\{TT_n - TT_{fi}\}$ than by mere comparison of a single value to a threshold. Various accumulators are used in *statistical process control* [Ryan 1989, chapter 5] for detecting if an industrial parameter is out of control.[3] We re-label the sequences of interest to include their time stamp t: $\{TT_{n,t}\}$ and $\{TT_{n,t} - TT_{fi,t}\}$. Any of the accumulators below may yield satisfactory performance, depending on the application environment:

- *Runs:* Alarm if R successive values are over a threshold K. In our growth experiment of section 5.1, we alarmed when the ratio $\frac{TT_n - TT_{fi}}{TT_n}$ exceeded 0.2 for a run of two values.
- *Duration:* Alarm if all values within some time range are over a threshold K.
- *Cumulative sum (CUSUM):* Alarm if the accumulated sum of values over a threshold K exceeds another threshold M.
- *Exponentially weighted moving average (EWMA):* Alarm if a geometrically weighted sum of W_t exceeds a threshold K:

2. These estimates are widely found in introductory statistics books, e.g. [Lindgren 1976, section 5.1].
3. We present motivations and charting techniques of statistical process control in section 8.2.

$$W_t = r(TT_{n,t} - TT_{fi,t}) + (1-r)W_{t-1}; W_0 = 0; 0 \leq r \leq 1. \quad \text{(EQ 14)}$$

The latter two methods are particularly useful for detecting small shifts in the mean of a time-indexed sequence [Ryan 1989, p. 122].

Just as when using threshold tests for significant trends, one should test any of these accumulation techniques on training data to choose parameter estimates yielding acceptable sensitivity and specificity.

6.1.3 TrenDx

The figure below shows a monitor set of two trend templates for interpreting whether the error scores of a fault trend are significant. Let TT_F denote the lowest error score of the fault trend, and TT_N the lowest error score of the normal trend. Both are time-indexed sequences. Each of the trend templates monitors changes in the difference (TT_N - TT_F) over the recent time period of length Δ. Δ is application dependent. In growth five years may be appropriate; in ICU monitoring five minutes may suit detection of short term trends. The trend template named "fault trend not present" checks that the difference in best scores stays near zero. The trend template named "fault trend present now" models that recently the difference in scores has become positive and increasing. The parameter β gives the rate of increase seen by experience in faulty cases.

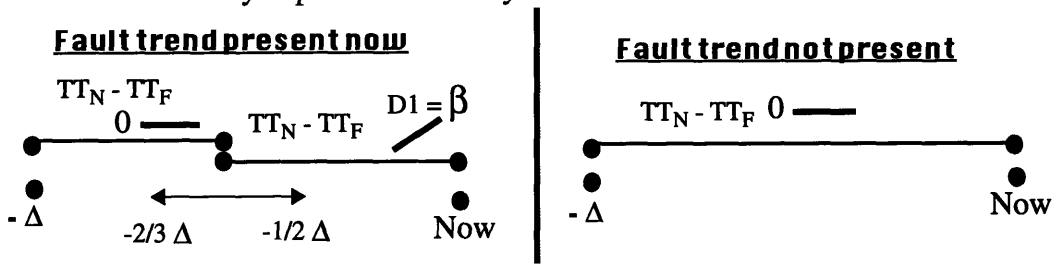


Figure 34 Proposed monitor set for onset of a fault trend.

Using TrenDx to monitor TrenDx error scores illustrates the potential recursiveness of this monitoring method. The results of matching to this monitor set will produce a time-indexed sequence of error scores much like that in Figure 33. However, monitoring this way really just delays the detection of trend significance one stage, with propagation of error in the conclusions.

The next few sections describe how, once the automated monitor has determined that at some time T_0 a trend is significant, the monitor can communicate this deduction or take further actions.

6.2 Generating Alarms

Having information that a fault trend is significant at time T_0, an automated monitor can send an alarm to operators or clinicians caring for the process. The alarm can state the name of the trend template and the time T_0. Additional text may include descriptions of important value constraints or intervals. The text for these-trend template components come fairly naturally from the knowledge representation and from the qualitative temporal interval relations. For example, the text for an alarm of a handbagging hemodynamic fault may read:

> Handbagging hemodynamic fault was detected at 12:27:21 a.m. A phase of decreasing blood pressure proceeded a phase of increasing heart rate.

An alarm may also display some or all of the data in the assignment of the best matching fault hypothesis at time T_0.

An automated monitor may wish to alarm based on a boolean combination of significant trends. Forward chaining of a set of production rules may follow any of the significance methods of the previous section.

A more thorough method for deciding whether to send an alarm at time T_0 is evaluation of a *decision model* weighing the costs and benefits of sending the alarm versus not sending the alarm. The utility of alarming is based on:

- Estimated probabilities of the normal and abnormal trends
- The probabilities, costs, and benefits of each action a clinician may take upon seeing the alarm.

The utility of not alarming is based on the trend probabilities as well as

- The probabilities, costs, and benefits of each action a clinician may take upon not seeing the alarm.

Considered as a single decision to made at time T_0, one can encode this decision model straightforwardly as a decision tree [Sox 1988, chapter 6]. This decision is more accurately made in the context of the time progression of the patient, partially as reflected by the best matching trend templates. A critical review of *dynamic decision modeling* techniques may be found in [Leong 1993].

6.3 Data Visualization

Operators in a monitoring environment with multiple channels of high frequency data may have extreme difficulty tracking all the data for significant changes. An intelligent trend detector such as TrenDx may be used to intelligently filter this data to show the operator only that data corresponding to an important component of a faulty process. This display can be driven by rules of the following form:

> If fault trend template TT_f is significant, then display data of parameter P_i during the time specification T_i (for i from 1 to some integer k).

The significance test may be determined via any of the means in section 6.1. These rules may be dependent on the process being monitored and the operator to observe the data.

Each time specification T_i is a Boolean combination of temporal intervals, and may be expressed in a temporal query language such as the time-line language of [Cousins and Kahn 1991]. Intervals of trend templates are natural choices of time intervals over which data should be displayed.

An example intelligent display appears below. The rule states that when the trend for a handbagging hemodynamic fault is significant, the display should show the blood pressures assigned to the decreasing phase of the trend template and a minute before.Thus this high frequency data stream has been filtered. This scheme for filtering data was developed as part of a broader framework for data visualization [Fackler and Kohane 1994].

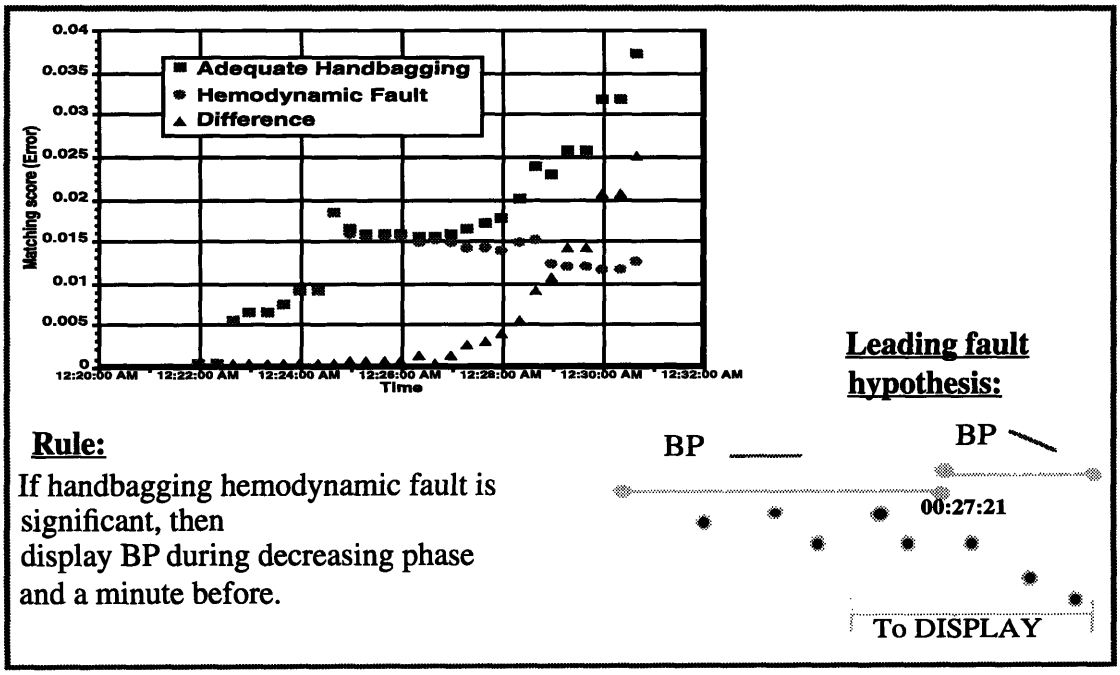


Figure 35 Intelligent data display based on significance of a fault trend.

The same rules may be used for automated summarization of process data over longer periods of time. A physician, for example, may not observe a particular intensive care unit patient for two hours, but could use a reliable report of all significant events in the data while he or she was away. This report can include all data generated by these display rules, with brief accompanying text.

6.4 Triggering Alternate Monitor Sets

In his survey on research into medical diagnostic reasoning, [Kassirer 1989] notes that "the diagnostic process focuses on one or more evolving diagnostic hypotheses" (page 893). A computer program receiving sparse time-ordered data on a handful of parameters may conceivably trigger plausible diagnoses, since human experts regularly accomplish this. Kassirer elaborates:

> Typically, the clinician triggers initial hypotheses merely from a patient's age, sex, race, and presenting complaints, but sometimes such hypotheses emerge exclusively from physical findings or laboratory data. Additional hypotheses are triggered as new findings emerge (page 894).

Monitor sets represent the set of competing trends in a diagnostic context. The significance of a fault trend may be a sign that the diagnostic context has changed. This in turn may warrant triggering of an alternate monitor set.

The monitor set representation may be supplemented to include rules of the following form:

> If fault trend template TT_f is significant, then trigger monitor set M_j. The temporal distance between point P of TT_f and the anchor point Q on each trend template of M_j is the range $[t_b, t_e]$.

The significance test may be determined via any of the means in section 6.1. TrenDx can apply these rules to monitor trends in a shifting diagnostic context.

6.5 Monitor Set Guidelines

6.5.1 Basic Principles

TrenDx diagnoses the significance of a fault trend based not on its own matching score but based on its *comparative matching score* relative to the normal trend. Consequently a knowledge engineer creating a trend template must keep in mind the other trend templates in a monitor set. Two broad principles should be followed:

1. A trend template should only include information that will distinguish it from competing trend templates.

This insures that the trend templates constrain a minimal number of parameters. For example, trend templates monitoring growth over a child's lifetime need not constrain serum electrolytes, for they vary in ways independent of growth disorders. It also insures that the time span of each trend template is long enough only to discriminate between the trends. Following this principle results in streamlined trend templates and a more efficient matching algorithm.

2. All trend templates within a monitor set should constrain the same parameters. Each parameter should be constrained over the same time span in all trend templates.

This insures that the scores of competing trend templates are directly comparable. If a sensor is noisy for some period of time, no trend template's score will gain or suffer more than any other's. Even given these restrictive guidelines, trend templates may differ in either the time or the value axis. The templates may differ in either the functions or the statistical models of their value constraints. The templates may also differ in the end points or length of corresponding intervals.

6.5.2 Rules

The fairly expressive trend template representation language in chapter 3 allows one to construct monitor sets that do not meet the above two principles. To limit such constructions we have developed rules for optimal monitor set construction. These rules apply to the competing trend templates within a monitor set. After each rule is an elaboration including comments on how it can be automatically verified.

1.The trend templates should represent the same landmark points.

The trend templates must have the same anchor points so that they can match the same data simultaneously. Extending this guideline to representing the same landmark points is for uniform explanations and temporal interpretations. By "the same" we mean having the same name and representing the same event. During matching the corresponding landmarks of different trend templates are often constrained to different times. This tule can be verified through a straightforward check of the names of landmark points of each trend template.

2.The trend templates should constrain the same process parameters.

The preceding section motivates this rule. Each value constraint in a trend template explicitly lists what parameters are modeled. The set of parameters constrained by a trend template is the union of the parameters modeled by each value constraint. Adherence to this rule can be verified by checking that this union is equal for all competing trend templates.

3.The total monitoring period for each measured parameter should be provably the same in each trend template.

The preceding section motivates this rule. Using the same anchor point in each trend template and a temporal reasoner such as TUP, one can determine the union of all time ranges over which a particular parameter is modeled in each trend template. There may be temporal uncertainty in the begin or end points of this combined range.

One must be able to *prove* that the total monitoring periods are identical because otherwise unappealing, temporally biased matches may result. For example, the left panel of Figure 36 shows two competing trend templates with a single

interval constraining temperature after a drug is given. At first glance the two templates seem to constrain temperature over the same period. The middle panel shows temperature data of a particular process. As TrenDx matches the data to the monitor set, the program optimizes the temporal uncertainty in each trend template to produce the lowest error. Consequently, the best hypothesis for TT_1 ignores data after time 2, and the best hypothesis for TT_2 ignores data before time 2. Both trends appear to fit the data well, but to different data. Sending or withholding an alarm based on matches to different sets of data is quite unsound.

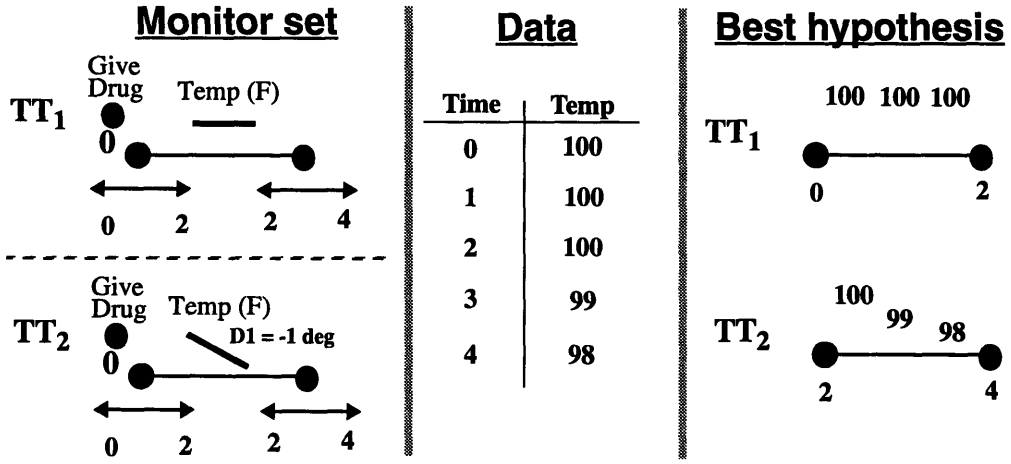


Figure 36 A monitor set resulting in best hypotheses that throw away data.

To automatically prove that the time spans of two intervals are equivalent, a program can show that the corresponding begin and end points are either (a) temporally equal and certain, or (b) linked by a fixed temporal distance to another temporally certain point. For example, the two trend templates in the handbagging monitor set of Figure 28 on page 84 model blood pressure over the same time span: the time between the two landmark points. Hence these two trend templates will match the same blood pressure data.

4. Remove sets of intervals with the same temporal and value constraints in all trend templates, if those intervals do not determine the times of other intervals or landmark points.

For example, the two intervals modeling oxygen saturation in Figure 28 on page 84 have identical time and value constraints in both trend templates. The match of patient data to these intervals with be the same in each trend template. Furthermore, these intervals do not determine the times of either landmark point or of other intervals. A program can automatically deduce this by inspecting the network of temporal links in each trend template.

One may make an exception to this rule if these intervals are needed to test the quality of match of an individual trend template. This necessity can come from experience on training cases.

6.5.3 Critiquing Programs

The above rules[4] may be directly encoded in a computer program that automatically inspects the competing trend templates within a monitor set and suggests improvements to their structure. We name this hypothetical program the *monitor set critiquer.* Wellman has noted [Wellman 1988] that critiquing a representation is computationally easier than automatically generating new representations.

BUNYAN [Wellman, Eckman et al. 1989] is a critiquing program that searches a decision tree for representational flaws that may lead to misleading optimal decisions. The program matches the given tree to a set of rules describing faulty trees. If a rule is satisfied, BUNYAN generates text describing the modeling error and offering improvements.

BUNYAN, and the monitor set critiquer suggest changes to types of representations based on implicit knowledge of the corresponding inference algorithms. These critiquing programs are completely independent of the application domain. A critiquer may generate more convincing and informative recommendations by drawing on a domain-specific knowledge base. The potential for this improvement has been demonstrated by the series of programs based on ATTENDING that critique therapy plans in anesthesia, pharmacy, ventilator management, and other medical applications [Miller 1986]. Each such program explicitly represents the clinical goals of its specialty. Consequently, the ventilator management advisor VQ-ATTENDING can advise how well a proposed ventilation plan supports the goal of stabilizing certain patient respiratory parameters.

6.6 Learning Trend Template Time Parameters

In some application domains one may not have complete enough knowledge to construct trend templates from domain experts or published literature. In this case one must automatically learn some component of the trend templates from training data.

We have designed the trend template knowledge representation aiming to minimize what must be learned. The trend template language of landmark points and intervals facilitates defining the *structure* of important phases and events in a trend. We presume that a knowledge engineer can readily describe the qualitative relations between pairs of phases: precede, overlap, meet, etc. Furthermore, we have chosen low-order polynomial value constraint functions so that a knowledge engineer can readily choose what characteristic shape fits the data over which phase. We assume the value constraints are specified except for the parameters to

4. Other rules may be added to this hypothetical program.

be estimated in matching to an individual. We also assume that the trend templates satisfy the guidelines of the previous section.

We present a supervised learning algorithm for estimating the time parameters of a trend template given the overall landmark point and interval structure and completely specified value constraints. The time parameters in each trend template are fully specified by a set of temporal distances between pairs of points: d_1, d_2,..., d_k. We assume that these distances are jointly distributed according to a multivariate normal density:

$$f(d_1, \ldots, d_k) = \frac{1}{\sqrt{(2\pi)^k \cdot |\Sigma|}} e^{\frac{-1}{2}\left(\sum_{i=1}^{k}\sum_{j=1}^{k} \sigma_{ij} (d_i - \mu_i)(d_j - \mu_j)\right)} \quad \text{(EQ 15)}$$

where μ_i is the mean for temporal distance d_i, Σ is the covariance matrix with determinant $|\Sigma|$, and σ_{ij} is an element of the inverse matrix Σ^{-1}.

We learn the set of μ_i and the coefficients of Σ for each trend template that give TrenDx desirable monitoring performance. We do this in two steps. In the first step, we gather for each trend template dense training data of cases presumed to fit the diagnostic class of that trend template. Each case is matched using TrenDx to the trend template is supposedly matches. We run TrenDx with a prediction granularity of zero and as large a beam size as is practicable, After each case runs, we select the best scoring hypothesis and obtain from the temporal reasoner the temporal distances we aim to estimate. Because the data are dense and the prediction grain is zero, each temporal distance will have minimal uncertainty; we take as the temporal distance the midpoint of the uncertain range of distances. We then use the temporal distance estimates to construct maximum likelihood estimates of μ_i and Σ. These estimates appear in most pattern recognition texts; see for example [Schalkoff 1992].

In the second step, we use our estimate of the distance distributions within each trend template to construct uncertain distances in the trend template definition. Each uncertain distance d_i is represented as the uncertain distance $[\mu_i - \alpha_i\sigma_{ij}, \mu_i + \alpha_i\sigma_{ij}]$; that is, the mean distance plus or minus some number α_i of standard deviations. We iteratively increment the α_i from 1 to 3 in small steps for each distance, and select the settings yielding the best combination of sensitivity and specificity in matching all training cases through TrenDx on all competing trend templates. We send alarms based on some method from earlier in this chapter.[5] Note that for each iteration, the theoretical sensitivity may be determined as the integral of the

5. The alarming method may also be a parameter to be learned, and can therefore be part of this iteration.

multivariate normal distribution (EQ 15) within the boundaries $[\mu_i - \alpha_i\sigma_{ij}, \mu_i + \alpha_i\sigma_{ij}]$. This calculation is demonstrated in [Stroud 1971].

6.7 Monitoring in a Large Clinical Setting

Until now we have described how TrenDx matches data from a single process to a single monitor set. Consider a clinical environment such as a hospital or an industrial plant in which many processes or patients are monitored. Each process has scores of parameters measured and at least as many distinct clinical contexts. Given such an environment and limited computational resources, one must design an architecture for managing the storage of process records and monitors.

Only the most relevant records and monitors should be in active memory. Queuing theory [Kleinrock 1976] investigates the related theoretical issues. The *multi-monitor simulator* [Haimowitz and Kohane 1991] is a workbench for exploring the practical trade-offs in implementing caching schemes for pertinent monitors. This program tests how different caching mechanisms scale up as the number of monitors and the number of patients increase.

In this chapter we demonstrate that TrenDx can be viewed not only as a trend diagnosis program but also as part of an intelligent monitoring environment. This environment can assist users in building representations. It can communicate with justifications the results of monitoring. It can also display concise summaries of high bandwidth data.

7 Alternative Representations and Algorithms

The representation and algorithms in the proceeding chapters give one method for detecting multivariate, temporally uncertain trends. In this chapter we present and critique other variations of trend templates and TrenDx.

7.1 Constraint-Based TrenDx

We call *constraint-based TrenDx* a program that matches process data to representations called *constraint-based trend templates*, which force data assigned to an interval to meet boolean constraints. The complete representation language and matching algorithms of constraint-based TrenDx may be found in [Haimowitz and Kohane 1993a; Haimowitz and Kohane 1993b]. In this section we focus on how constraint-based TrenDx differs from the polynomial regression-based TrenDx of chapters 3 and 4.

7.1.1 Value Constraints

In constraint-based trend templates the value constraints restrict a real-valued function of the data assigned to an interval to fall within a certain range. Each value constraint is an expression of the form

$$m \leq f(D) \leq M \qquad \text{(EQ 16)}$$

where f is some real valued function defined on a set of process data, m is a minimum (possibly $-\infty$), and M is a maximum (possibly $+\infty$). In constraint-based TrenDx the function f is evaluated on the set D of data currently assigned to that interval and the result is checked for being between m and M.

The figure below shows a portion of the male average normal growth trend represented as a constraint-based trend template. This trend template has the same interval structure and temporal constraints as the growth pattern of Figure 22 on page 74. What differs is the value constraints. This trend template encodes that height and weight centiles vary in the same way during `early childhood` by constraining the difference between the average velocity of height Z-scores and the

average velocity of weight Z-scores to be within a small number α of zero. During `pre-puberty` this trend template has two value constraints: the average velocities of height Z-scores and that of weight Z-scores are within small values of zero. The average velocities are computed for the two most recent heights or weights assigned to that interval. The knowledge engineers selected values so that TrenDx had 100% sensitivity in diagnosing this growth trend on a training set.

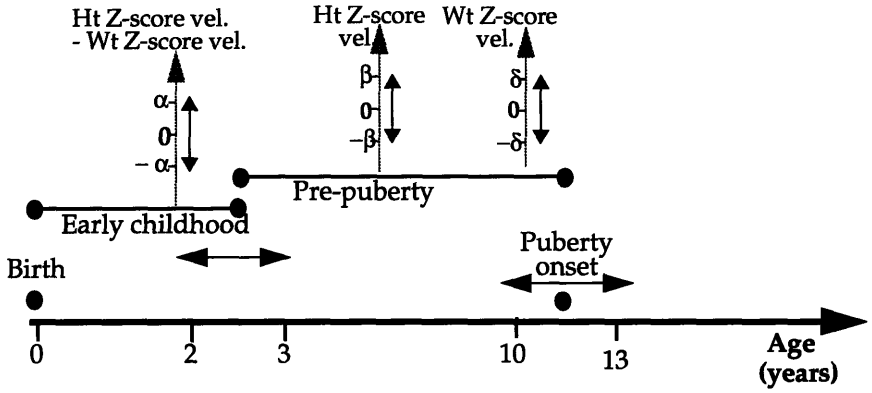


Figure 37 Portion of a constraint-based trend template for male average normal growth.

7.1.2 Trigger Sets

To incorporate triggering of diagnoses, constraint-based trend templates include a function *TRIGGER* that computes for each value constraint and the direction of failure a set of alternative trend templates.

$$TRIGGER: vc \times \{low,high\} \rightarrow \{TT_1,TT_2, \ldots, TT_k\} \quad \text{(EQ 17)}$$

Thus if the value constraint function f(D) falls below the minimum m or above the maximum M, constraint-based TrenDx matches the data of the monitored process against the new set of triggered disorder templates. Those templates for which the data match may become active hypotheses. For example, consider the height value constraint of interval `pre-puberty` of the average normal growth trend template, reproduced in Figure 38. If for the two most recent patient heights the computed average velocity of height Z-scores is below -δ then the suggested trend templates should include constitutional delay but not early puberty. If the first difference is above +δ, just the opposite. Therefore trigger sets help decompose the space of trend patterns to consider. This triggering mechanism is similar to the use of daemons by the Present Illness Program [Pauker, Gorry et al. 1976] to trigger disease frames into the list of active hypotheses.

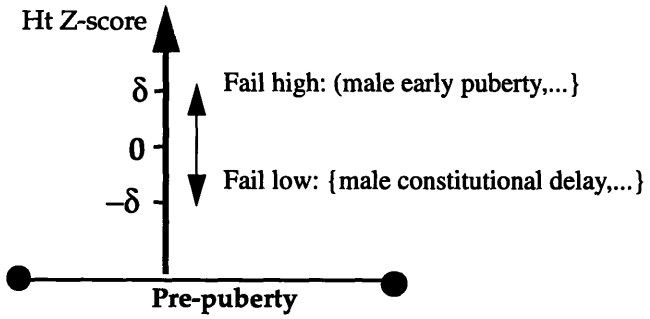


Figure 38 Trigger sets on a value constraint in a constraint-based trend template.

7.1.3 Expressiveness of Representation

The form of value constraints in constraint-based trend templates corresponds to restrictions on laboratory values found in medical domains where no pathophysiological model is clearly understood. For example, the section on "Diagnostic Procedures in Liver Disease" in Harrison's Principles of Internal Medicine [Podolsky and Isselbacher 1991, page 1309] describes laboratory results of patients with liver disorders. Of primary importance are the hepatic enzymes that aid decomposing and rebuilding amino acids, one of which is aspartate transferase (AST). The authors note that a patient with massive hepatic necrosis may have marked elevations (perhaps over 500 IU) of AST within the first 24 to 48 hours, but 3 to 5 days later the levels may be in the ranges of 200 to 350 IU. The levels of AST are ill-specified in both time and value primarily because the text aims to summarize a pattern for all hepatic necrosis patients; inter-patient AST trajectories may vary significantly. Below is a trend template for this pattern.

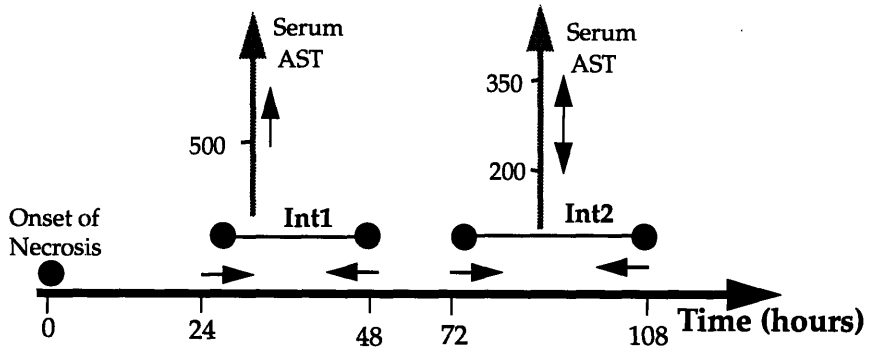


Figure 39 Trend template for AST pattern in hepatic necrosis.

The landmark point for this template is the onset time of hepatic necrosis. Interval `Int1` represents the period where AST levels are above 500 IU. This

period begins 24 hours or later after onset of necrosis, and ends 48 hours or sooner after the onset of necrosis. Interval `Int2` represents the period where AST levels are between 200 and 400 IU. This period begins 72 hours or later after onset of necrosis, and ends 108 hours or sooner after the onset of necrosis.

One can also represent as a value constraint that a certain parameter P_t must be close to a real-valued function g(t) over some time interval by using the value constraint:

$$-\delta \leq f(D) = (P_t - g(t)) \leq \delta \qquad \text{(EQ 18)}$$

The positive number δ is a noise threshold. It may equal 2σ, where σ is the standard deviation of a white noise variable. Examples of potentially useful g(t) are polynomials, exponentials, and trigonometric functions.

7.1.4 Constraint-Based Matching

Constraint-based TrenDx diagnoses trends by matching process data to the constraints of trend templates. The program, like regression-based TrenDx, may maintain multiple hypotheses for the same trend template. Those hypotheses where all process data have matched the value constraints are retained. Those hypotheses where some data do not match are discarded. If no hypothesis for a trend template remain, then as constraint-based TrenDx removes the last hypothesis for that trend template it also triggers other trend templates.

To illustrate the difference between matching by constraint-based TrenDx and regression-based TrenDx, we describe how constraint-based TrenDx performs on the constitutional delay patient case we illustrated in Figure 15 on page 57. Through age 3.1 of the patient, two hypotheses are possible for the average normal growth trend template, as shown in Figure 40. In this patient, the z-score (with respect to the average puberty population) for the height at age 2.1 is -1.25 and that for the height at age 3.1 is -2.52. The average velocity of -1.27 per year is well below the minimum on the height value constraint on the `pre-puberty` interval. Therefore in the second hypothesis, the new height does not match the trend template. As shown in Figure 40, constraint-based TrenDx prunes the hypothesis that assigns all data to the `pre-puberty` interval. Regression-based TrenDx retains both hypotheses and gives a better score to number (1).

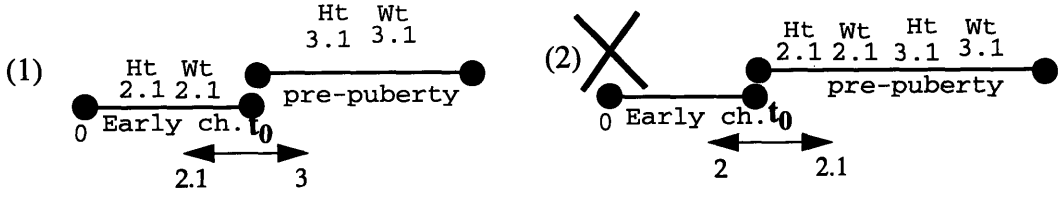


Figure 40 Constraint-based TrenDx prunes hypotheses where data do not all satisfy the value constraints.

Constraint-based TrenDx continues to compare the patient's growth data to the remaining hypothesis. This hypothesis persists through several years of data as constraint-based TrenDx assigns more heights and weights to `pre-puberty`. Eventually, the height z-score for the average pubertal population drops from -2.08 (age 6) to -2.61 (age 7). The average velocity from age 6 to age 7 falls below the lower bound on the height value constraint of `pre-puberty`. The last average growth hypothesis is pruned. Since there are no more hypotheses for the male average normal growth template, constraint-based TrenDx triggers male constitutional delay according to the trigger set of `pre-puberty`. The program matches the patient's data to the new trend template. A single hypothesis of constitutional delay persists longer than did any hypothesis of average normal growth.

7.1.5 Exploratory Clinical Trial [1]

7.1.5.1 Methods

We conducted a pilot clinical trial of constraint-based TrenDx to evaluate its performance and to determine the weakness of the representation and knowledge engineering. Data from 30 patients seen at the Division of Endocrinology at Boston Children's Hospital were retrieved from the Clinician's Workstation (CWS), an on-line charting system [McCallie, Margulies et al. 1990]. The data included height, weight, sexual staging and bone age measurements. The patients were selected by filtering the problem list associated with each patient record in the CWS. Since this was an exploratory experiment rather than a rigorous test of efficacy, we specifically selected those problems and patient types for which we had engineered trend templates, as well as growth hormone deficiency, to explore how best to implement templates for this trend.

The first ten patient cases were used as a training set. As errors in the performance of constraint-based TrenDx on the training set were identified, we modified the trend templates by changing time ranges of the interval end points, by changing bounds of the value constraints, and by adding new intervals with new value constraints.

The remaining twenty patient cases were used as a test set. These included patients with growth hormone deficiency, constitutional delay, average tempo of development, and early puberty. Each case was read by constraint-based TrenDx in chronological order. The program recorded all the diagnoses and the age of the patient when they were considered or rejected. At the time of this trial, we had not developed trend templates for intervening trends like that for acquired growth hormone deficiency. All constitutional delay trend templates which constraint-based TrenDx eventually rejected for any of these reasons:

1. This section was originally written by Dr. Isaac Kohane for [Haimowitz and Kohane 1993b].

1. the velocity of the height Z-score was too low,
2. the velocity of the weight Z-score was too low, or
3. the bone age was too far behind chronological age

were scored as diagnosing growth hormone deficiency when the constitutional delay trend template was ruled out.

Concurrently, a panel of three expert endocrinologists was given the twenty test cases and the task of diagnosing each patient. The endocrinologists were given the benefit of seeing all data for each case at once. By contrast, constraint-based TrenDx saw data one visit at a time. The experts were required to judge the earliest age at which they could make their diagnosis. Note that this level of growth chart scrutiny is unusual in a busy general pediatric office. Several of the important contextual clues to the diagnosis that are usually gleaned from the patient history or laboratory results (e.g. results of a serum growth hormone test) were not available to either the clinicians or constraint-based TrenDx. Given the limitations of the presented data, even in those cases where the panel consensus was different from the diagnosis stored in the CWS we took the panel consensus as the expert standard for a "correct" diagnosis.

7.1.5.2 Results

10 of the 20 patients were diagnosed by the panel as having one of these six disorders: normal growth, short stature, constitutional delay, early puberty, precocious puberty, and obesity. Of these constraint-based TrenDx diagnosed 9 of 10 correctly. In 8 of the 9 correct diagnoses the clinicians reached the diagnoses at the same time as constraint-based TrenDx; in one case the clinicians were earlier. In the one case misdiagnosed by the program the panel diagnosed constitutional delay, and the program diagnosed average prepubertal growth. This occurred because the patient's velocity of height standard deviation never crossed the lower bound of the value constraint on the average prepubertal growth trend template ($-\delta$ in Figure 37).

The other 10 patients were diagnosed by the panel as having growth hormone deficiency. Of these, constraint-based TrenDx diagnosed 5 of 10 correctly. In 3 of the 5 misdiagnosed cases a constraint on proportionality (such as body mass index or build, see section 5.1.1) could have been used to correctly trigger growth hormone deficiency. For instance, constraint-based TrenDx misdiagnosed one case of growth hormone deficiency as having constitutional delay. In this case, the clinicians noted that the weight and height did stay within a broad channel of their standard deviation. However they also noted that the weight standard deviation of the patient was creeping upwards at the same time that her height standard deviation was creeping downwards. As we had not encoded in the constitutional delay template any constraints on the proportionality of height and weight after infancy, constraint-based TrenDx did not notice these subtle but significant opposing trends in

height and weight. From this we have learned to add a constraint on proportionality to most trend templates.

The results of this trial were encouraging in that they demonstrated that constraint-based TrenDx could diagnose a few trends. However, the number of test cases, and the nature of the cases do not permit us to make any conclusions regarding the performance of the program in general pediatric practice.

7.1.6 Comparing Regression-Based and Constraint-Based TrenDx

Due to different experimental methods and small sample sizes, the results for constraint-based TrenDx cannot be reliably compared to those of regression-based TrenDx (section 5.1.3 on page 78). We report some results only as a very rough comparison. In the experiment with constraint-based TrenDx we checked only for sensitivities. Results on twenty patients were .50 for growth hormone deficiency and .90 for all other disorders. The results for regression-based TrenDx were from thirty patients, including the aforementioned twenty. Sensitivites were .46 for growth hormone deficiency and .65 for all other disorders. The reduction in sensitivities in regression-based TrenDx is most directly due to averaging out of infrequent yet diagnostically important measurements: bone ages and pubertal stages. In section 5.1.5 on page 82 we have discussed how to modify value constraint weighting to raise sensitivities for regression-based TrenDx.

Direct comparisons of sensitivities aside, regression-based TrenDx offers clear advantages in ease of modeling, and in robustness and information content of results. Matching in constraint-based TrenDx may sometimes be brittle, in that a single datum just beyond a value constraint threshold rules out a hypothesis. Because the only status of hypotheses are satisfaction or failure, constraint-based TrenDx cannot rank the quality of match among different hypotheses. A user of constraint-based TrenDx only knows which hypotheses are *possible*, not which ones match the data well. Building a constraint-based trend template requires either learning or knowledge engineering of the value constraint boundaries.

By contrast, regression-based TrenDx matches noisy or unexpected data more robustly. The matching algorithm ranks the competing hypotheses both within and across trend templates. Value constraints in polynomial regression-based trend templates require less knowledge engineering of internal parameters: one may choose from seven qualitative models according to the expected shape of the data.

7.2 Other Chronology Search Strategies

As mentioned in section 4.3.2, the space of chronologies for a given trend template grows polynomially with the number of data clusters processed. TrenDx employs beam search to limit the number of chronologies. However, as discussed in section 4.5.1, with a small beam size TrenDx may prematurely remove chronologies due

to noisy data. The three techniques in this section take different approaches to managing the chronology space.

7.2.1 Parallel Hypothesis Evaluation

The TrenDx matching algorithm (section 4.3 on page 54) is readily made parallel at two levels. First, TrenDx applies the same matching procedure to each active hypotheses. Second, TrenDx computes all consistent temporal contexts for a hypothesis in part by asking the same temporal query to each of a set of candidate temporal contexts. A parallel implementation of TrenDx could drastically reduce processing time at both these levels, and thus in the entire algorithm. It would also allow a much larger beam size, should beam search be necessary.

In principle, TrenDx could assign the structure of each active hypothesis to a processor in a single-instruction, multiple data (SIMD) computer. A review of several SIMD computers that have been applied to artificial intelligence is in [Shrobe 1988]. TrenDx could issue the same commands to each processor for matching a particular data cluster to its hypothesis. Furthermore, the commands for finding all consistent temporal contexts (see section 4.3.3 on page 59) can lead each processor to recruit additional processors for each possible temporal context. This parallel version of TrenDx has not been implemented.

7.2.2 Persistence Heuristic

When matching to a monitor set that tracks a single variable over many consecutive phases, TrenDx could use a persistence heuristic to reduce branching to alternate chronologies. The heuristic states that if the current datum matches the first of two consecutive phases well (within some threshold), then TrenDx need not consider assigning that datum to the second phase. We illustrate in Figure 41. Although some data appear within the population uncertainty of the trend, TrenDx assigns these data to phase 1 without considering phase 2. The main time and space savings comes from TrenDx not having to create the additional hypotheses or temporal chronologies. This heuristic has been implemented. Time savings were dramatic when TrenDx used this persistence heuristic in matching data to the blood pressure classification monitor set of section 3.7.

Once the data no longer match sufficiently well to the first phase, TrenDx can branch as usual to consider assignment to either phase one or phase two. As subsequent data arrive, the poorest matching chronologies will be removed by the beam search. This technique of deciding how long a phase persists is similar to the statistical techniques used for finding the times of level shifts and variance changes in [Tsay 1988].

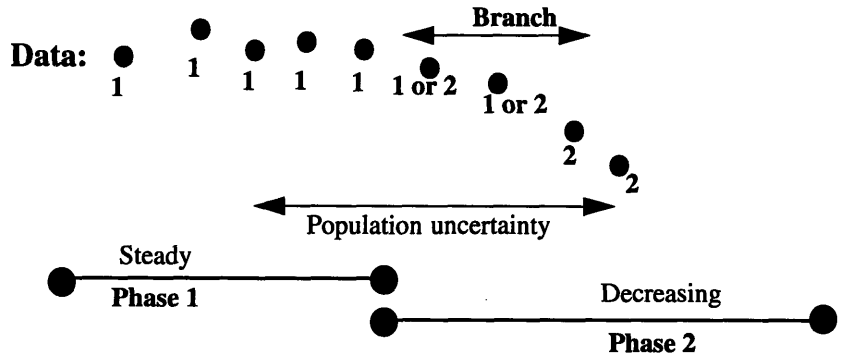


Figure 41 Use of a persistence heuristic to limit branching of chronologies.

Recall that the goal of TrenDx matching is to find the chronology for a trend template that *optimizes* the matching score. For a univariate, simple trend as in Figure 41, applying the persistence heuristic will result in a chronology fairly close to optimal. However, this heuristic can lead to unsound and damaging reasoning for a trend template tracking multiple variables in distinct, temporally related phases. Consider the trend template for compromised venous return (section 3.4). The phase transition for blood pressure is temporally linked as zero to five minutes before the transition for heart rate. Should TrenDx extend the first phase of the blood pressure trend, the result may be a poor overall match if there is a rapid rise in heart rate.

A TrenDx user may choose whether or not to use the persistence heuristic within a given monitor set. The persistence heuristic is best used on univariate trends over consecutive phases, and is recommended in such cases due to dramatic improvements in efficiency. The few examples tested have shown virtually no loss of diagnostic discrimination when using the persistence heuristic.

7.2.3 Repeated Iteration of Chronologies

An alternative to beam search of competing chronologies per trend template is re-evaluating all chronologies with each new data cluster. Call the anchor point on a trend template p, and let $p_1, \ldots, p_n$ be the points in the trend template that are temporally uncertain with respect to p. Let the temporal distance range between p_i and p be $[s_i, t_i]$. Given a prediction granularity of 2δ, we can subdivide each temporal uncertainty range into segments of size δ. If the number of segments for point p_i is m_i (roughly $(t_i - s_i)/\delta$), then the total number of chronologies to consider for matching a data stream is $\prod_{i=1}^{n} m_i$. For each chronology[2], each of the points $p_1, \ldots, p_n$ are fixed in time with respect to the anchor point p.

A new version of TrenDx, not yet implemented, could iterate through each of the temporal chronologies and match them with respect to all previously matched data. This would be performed with each new data cluster. TrenDx would store the top several hypotheses per trend template in the record of the monitored process.

The benefits to this approach are twofold. First, when all trend template points are fixed in time with respect to the anchor point, the time for matching a hypothesis to data is *drastically* reduced from when some points are temporally uncertain. Second, this algorithm guarantees a better soundness in choosing the optimal chronology per trend template, since it actually considers all chronologies within the specified granularity. A drawback is that much of the matching would be continually reproduced. This problem could be somewhat remedied through memoization of matching scores between data sets and intervals. If soundness of temporal chronologies is a priority, and a constant factor decrease in efficiency is tolerable, this matching technique may be quite helpful.

When considering any variant to regression-based TrenDx with beam search, one must consider one's goals in the trade-offs between computational efficiency and soundness of temporal reasoning. One must also decide what ranges of the sensitivity versus specificity trade-off are acceptable. Finally, one should run experiments on training data to compare diagnostic performance between alternate techniques and regression-based TrenDx with beam search.

2. Some of these chronologies may be temporally inconsistent. A temporal reasoning program must notify TrenDx of this so that TrenDx will skip matching data to this chronology.

8 Related Work

TrenDx consists of two main components. The first is a set of *patterns* describing how data vary over time in some population. The second is a *matching algorithm* that scores how well data from a particular process match the patterns. In this review of related work we focus on the patterns and algorithms of other diagnostic monitoring schemes. We also compare the applicability of each competing method with that of TrenDx.

We first cover two general statistical approaches to diagnostic monitoring: system identification and statistical process control. These sections broadly cover most techniques applied to biomedical trend detection [Avent and Charlton 1990]. We then turn to the artificial intelligence (AI) literature, first describing representations of temporal patterns and then monitoring algorithms.

8.1 System Identification

System identification is modeling a stochastic system with parameterized equations, and estimating the parameters from observed data that supposedly fit the model. In a model to be used for trend detection, time is one of the parameters.

In diagnostic monitoring, each model may represent a diagnostic category, say a symptom or a disorder. Some of the available data are separated as *training data* used for estimating parameters of the model. The rest of the data are then matched to the fully specified models. In diagnostic monitoring, matching time-ordered process data to parameterized models has two primary benefits:

1. One can measure how well the new data fit each model, thereby diagnosing the process.
2. Having selected the best model, one can (under certain distribution assumptions) predict how the process will perform in the future.

A review of all techniques in system identification is beyond the scope of this thesis; for excellent introductions see [Kumar and Varaiya 1986; Ljung 1987]. Here we describe methods often used for trend detection.

8.1.1 Regression Analysis

In regression analysis the system to be identified models a single dependent (or response) variable Y to be some function of a vector of independent variables **X**. We focus specifically where Y and **X** are indexed by time t, in which case the model has the form:

$$Y_t = M(Y_t, \mathbf{X}_t) + \varepsilon_t \qquad \text{(EQ 19)}$$

where the function $M(Y_t, \mathbf{X}_t)$ defines the regression model, and ε_t is the unexplained component of the response at time t.[1] This equation naturally accounts for multivariate patterns, and more than one regression equation may be used. Note that the model function only relates variables within the same sample time.

The parameters in $M(Y_t, \mathbf{X}_t)$ are typically estimated in one of three ways:

1. distribution-free minimization of least squares
2. distribution-based maximum likelihood given the data
3. Bayesian updating, in which a *probability density function* of each parameter is learned from the data

Each technique is found in most applied statistics or pattern recognition texts, e.g. [Schalkoff 1992, chapter 3]. We have discussed the assumptions required for using this model for accurate estimation and prediction in section 4.4.3 on page 62. In particular, this model estimates parameters accurately when the sequence $\{\varepsilon_t\}$ is not autocorrelated.

One area to which regression modeling has been frequently applied is *growth*, whether it be of organism populations or of children's heights and weights. For over sixty years researchers have modeled human height as a multi-parametric regression equation. One growth regression model for height is the nine parameter *triple-logistic curve* [Thissen and Bock 1990, page 294]:

$$\text{Height}(t) = \frac{a_1}{1+e^{-b_1 \cdot (t-c_1)}} + \frac{a_2}{1+e^{-b_2 \cdot (t-c_2)}} + \frac{a_3}{1+e^{-b_3 \cdot (t-c_3)}} \qquad \text{(EQ 20)}$$

The three addends in the model represent the three components of childhood growth: infancy, middle-childhood, and adolescence. The authors give rough medical interpretations of each parameter (for components k = 1, 2, 3):

1. This development is similar to that in [McPherson 1990, page 518]

a_k = component k's contribution to mature stature

b_k = a parameter proportional to the maximum growth velocity of the component (maximum rate of growth is (a_k * b_k) /4 centimeters per year)

c_k = the age in years at which the maximum growth velocity occurs [page 295].

Probability distributions for the nine parameters are found by Bayesian updating, under the assumption of a multivariate normal distribution. *A priori* parameter distributions are found from national growth standards [Hamil, Drizd et al. 1979].

In a review article on this work [Thissen and Bock 1990], the authors find these models useful for comparing patterns of growth between populations (e.g. different racial groups), and for comparing the effects of hereditary versus environmental factors on growth. They have not suggested that the regression models be used for tracking or diagnosing growth disorders in individuals. Reliable use of regression models alone for diagnosis would require large growth studies of patients in each diagnostic category. Furthermore, multivariate trends improve discrimination between competing growth diagnoses better than trends in height alone.

TrenDx uses low-order polynomial regression models within its value constraints. TrenDx aims to distinguish competing diagnoses by the temporal extents of their regression relationships.

8.1.2 Time-Series Analysis

Time series analysis models generally specify how past and future values interrelate. [McPherson 1990, chapter 21] notes that a time series Y_t consists of four main components:

- one or more seasonal components S_t, present in some natural and econometric time series,
- a cyclical component C_t, present in many physical systems,
- a trend line T_t, representing the predominant, non-periodic change in the mean over time, and
- a stochastic or random component R_t, the combined effect of factors not otherwise accounted for in the model.

Typical means of combining these four components are either additive:

$$Y_t = S_t + C_t + T_t + R_t \qquad \text{(EQ 21)}$$

or multiplicative:

$$Y_t = S_t \cdot C_t \cdot T_t \cdot R_t \qquad \text{(EQ 22)}$$

We merely overview time series analysis here; standard introductory texts are [Box and Jenkins 1970; Priestly 1981]. The focus of time series analysis is accurate modeling of the stochastic component R_t, which is usually presumed *station-*

ary, i.e. having constant mean and variance over the period of interest. There may be autocorrelation in the sequence R_t. One often used framework for modeling R_t is the *autoregressive moving average* model, ARMA(p, q):

$$R_t = \sum_{i=1}^{p} a_i R_{t-i} + \sum_{j=0}^{q} b_j \varepsilon_{t-j} \qquad \text{(EQ 23)}$$

where the ε_t are independent, identically distributed white noise variables. An alternative means of modeling R_t, and perhaps periodic components as well, is through *spectral analysis* [Percival and Walden 1993].

Time series and spectral analysis have been applied extensively to econometrics [Stock and Watson 1988], financial markets, physical phenomena, and adaptive control of biomedical signals such as serum glucose [Fischer, Schenk et al. 1987]. These applications have shared a need for close fits between model and data, and accurate prediction of future values. Neither of these is necessary in diagnostic monitoring.

Our focus has been on diagnostic monitoring by modeling trend components of multiple variables over different phases. Generally, the goodness of fit and predictive power of trend template models may not be as accurate as time series models that account for cyclical and autoregressive components. The comparative advantage in using TrenDx is ease of model development, especially in domains with scant training data, while maintaining discriminatory power between diagnoses. Although time-series models may in principle be used within value constraints of trend templates, they should be incorporated only if competing diagnoses can be much better distinguished.

8.1.3 Hidden Markov Models

Another type of stochastic model for time-varying data is the state-based hidden Markov model (HMM); a good introduction is [Rabiner and Juang 1986]. HMMs have been applied to pattern recognition from high-frequency data, such as recognizing words in human speech [Parsons 1986, chapter 11]. Each pattern is represented by a network of states, each with a set of output probabilities and transition probabilities. The Markov assumption is that the probability of a process being in a particular state is dependent only on the previous state. Given a sequence of data, one can compute the probability of competing HMM models generating that data. The large number of parameters per HMM dictates that each pattern be trained with sizable data sets.

Hidden Markov models and trend templates may both represent time-varying trends of multiple parameters, yet each can represent patterns the other cannot. States in an HMM are analogous to intervals in a trend template, in that each represents process phases of variable time length. State transition probabilities in an

HMM correspond to temporally uncertain phase transition points in a trend template. HMMs are designed for representing patterns consisting of a sequence of consecutive phases. Trend templates can represent not only sequences of consecutive phases but also patterns with overlapping or disjoint phases. Trend templates are also much more compact representations for multivariate trends. Furthermore, trend templates can directly encode strict time boundaries between interval endpoints. The primary advantages of HMMs are their clear probabilistic semantics and straightforward learning algorithms.

8.1.4 Neural Networks

[Whitely and Davis 1992] have applied back-propagation neural networks [Rummelhart, Hinton et al. 1986] to diagnosis of bivariate, piecewise linear trends. They tested their techniques in identification of normal versus faulty tank loading in chemical plants. The diagram below illustrates the qualitative difference between a normal and a faulty trend in this domain. Numerical scales are different for the two signals, and have been omitted.

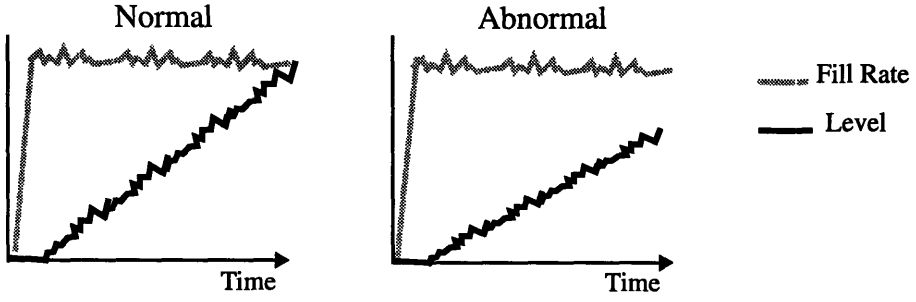


Figure 42 Normal and abnormal tank filling trends for neural network learning.

The network consisted of three layers: one hidden layer, with one input and one output node. Each training case consisted a sequence of ten readings of fill rate and ten corresponding readings of tank level. Experimental results showed that after training, the network could distinguish perfectly between normal and abnormal cases. However, randomly generated test cases were always classified as normal, showing that the true definitions of the two patterns had not quite been learned. This overall approach may be promising for learning similar trends to those diagnosed by TrenDx. The value of this technique must come from much further testing and expansion to trends with larger numbers of variables.

Neural networks may have been applied too broadly in this project. The above examples illustrate clear patterns expressible with a mixture of qualitative and quantitative constraints. What needs to be learned are the critical time transitions and stability values. Were the problem re-framed this way, the network classifier might have achieved higher sensitivity. Knowledge about what patterns are expected should not be discarded.

8.1.5 Shortcomings

Most system identification techniques rely on large amounts of training data for reliable estimation of internal parameters. This may be a shortcoming in some application areas. In pediatric growth monitoring it is difficult to acquire sufficient training data from populations of children with growth disorders.

The internal parameters tend to hinder explanations of detected trends. Our research goals included not only detecting clinically significant trends but also explaining in text and graphics the results of our diagnoses. We have outlined means of achieving this explanation in chapter 6. Most system identification techniques are not well suited to explanation, chiefly because of their many internal parameters that have no intuitive meaning. The widely published auxologist J.M. Tanner corroborates this view in this summary of statistical growth curve models:

> Many attempts have been made to find mathematical curves which fit, and thus summarize, human and animal growth data. Most have ended in disillusion or fantasy; disillusion because fresh data failed to conform to them, fantasy because the system eventually contained so many parameters (or 'constants') that it became impossible to interpret them biologically [Tanner 1990, pages 8-9].

While trend templates do incorporate regression models with internal parameters, these models are within an overall structure of landmark points and phases that lend themselves to natural language and graphic descriptions.

8.2 Statistical Process Control

Statistical process control (SPC) is a half-century old field rooted in the monitoring of industrial processes, and has been applied to quality control of manufacturing, to building of automated controllers, and to sensor validation. A primary goal of SPC is to conclude whether a *statistically significant* change has occurred in measurements of an industrial process. [Ryan 1989] is a good overview of the motivations and statistical techniques in SPC. We shall merely discuss how SPC may be used for detecting multivariate, temporally uncertain trends.

In SPC a set of samples of an intermediate industrial parameter is measured repeatedly over time. For example, every five minutes the diameters of ten randomly selected bolts may be measured, or the temperatures in four locations of a tank recorded. Some statistic of this sample, typically mean, range, or standard deviation, is calculated and plotted over time. Call this statistic P. In using *control charts*, or *Shewhart charts*, one assumes this statistic is approximately normally distributed, and therefore can establish limits within which this statistic will almost always fall. These limits, called the upper control limit (UCL) and lower control limit (LCL) are typically chosen to equal the mean of P plus or minus three sample standard deviations of P for all values displayed in the chart. If the monitored system is in statistical control, then samples of P will fall outside of the two control

limits approximately 0.27 percent of the time. An extensive survey of control charts with examples is in [Kume 1987, chapter 7]. The figure below illustrates an "X-bar" chart plotting the mean of the samples over time. The dark line in the middle is the mean of the sample means.

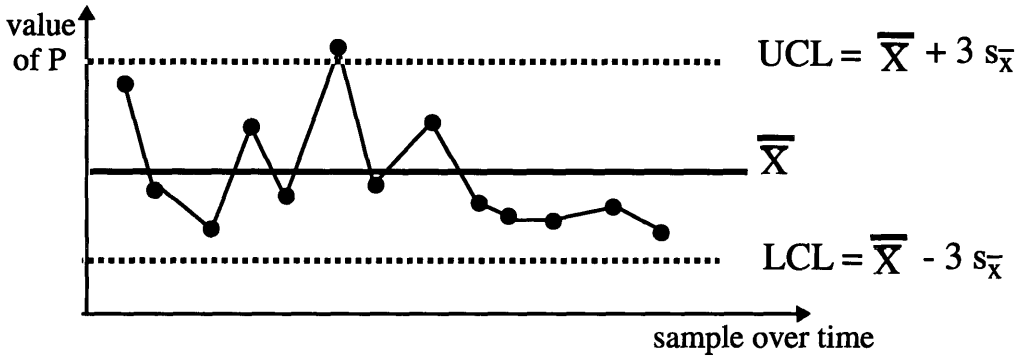


Figure 43 An $\overline{X}$ control chart.

If a sample measurement of P falls outside of the control limits, it may be detected either as a chance event, an outlying point due to measurement error, or a sign that the monitored process is out of control. A major goal of SPC is to detect and alarm for an out of control process with a minimal *average run length* (ARL). The ARL is the expected number of samples before an alarm is sent. A concurrent yet competing goal is a low false positive rate. In principle control charts are similar to the threshold monitors in widespread use in hospital intensive care units.

One may construct control charts to detect multivariate trends either with a chart for each variable or a single combined chart that plots a T-distributed statistic that makes use of the sample covariance matrix [Ryan 1989, page 215]. However, control charts do not lend themselves to expressing temporal correlations between different trends.

As with any trend representation, the expressiveness of control charts limits the types of automatic inference that can be performed. The last six sample points in the above chart may signify a decreasing trend that cannot be automatically detected. Similarly, although trend templates have a more expressive language, TrenDx can only recognize explicitly modeled trends. The primary advantage of control charts is that their grounding in statistical sampling lets one assign probabilities to the meaning of outliers.

Several accumulation methods are used in SPC to more quickly and reliably determine when the process is out of control, or has undergone a shift of mean values of P. Three elementary methods are counting of runs, cumulative sum (CUSUM), and exponentially weighted moving average. These methods are described in detail in [Ryan 1989, chapter 5]. In section 6.1 on page 91 we

described applying these accumulation techniques to sending alarms based on TrenDx matching results.

8.3 Representations of Temporal Patterns

Two areas of AI research address the representation of time-varying constraints on measurable parameters. One body of work encodes the *history* of a process as a sequence of time intervals over which parameters are constrained. Another set of literature represents *predicates* of parameter values at specified times with the intent of matching the predicates to process data.

8.3.1 Process Histories

In his landmark paper on naive physics [Hayes 1985, page 479] describes the importance of representing *histories*: "pieces of space-time with natural boundaries, both temporal and spatial." Hayes expresses histories by extending the traditional situations and actions language of AI search and planning.

8.3.1.1 Qualitative Histories

This notion was related to constraints on measurable parameters in the *process histories* of Qualitative Process Theory [Forbus 1985, page 115]. For example, the history of a ball dropping through a flame includes the five states of the ball's position (above the flame, at the top, within, at the bottom, or below) and the changes in the ball's temperature (zero, positive, or negative) during intervals bridging those states.

Qualitative behaviors generated from the QSIM qualitative simulator [Kuipers 1983] are used as *disease histories*. These histories can be matched to patient data for diagnostic monitoring. [Coiera 1990] From a qualitative model for acid-base and electrolyte balance Coiera uses QSIM to produce qualitative behaviors of the sequence of significant changes of pH, CO_2 and HCO_3^- in distinct acid-base disorders. These behaviors can be soundly combined by *qualitative superposition* to produce disease histories for combined acid-base disorders such as concurrent respiratory and metabolic acidosis. The qualitative models are also applied to diagnosis of acid-base disorders from monitored data. [Coiera 1989]

The high frequency, multiple data stream environment of chemical plant monitoring is similar to that of an intensive care unit. Consequently, automated intelligent monitoring is an active research area in chemical engineering. Several researchers have investigated monitoring normal and faulty trends using a fixed syntax of qualitative trend components. [Cheung and Stephanopolous 1990a] developed a library of seven qualitative trends based on *triangular components*: qualitative first and second derivatives. This trend library is roughly identical to the seven low-order polynomial models in trend template value constraints. Their

algorithm can divide a univariate data stream into consecutive phases over which one of the seven qualitative trends holds. In later work [Cheung and Stephanopolous 1990b] extended this framework to detect univariate trends at multiple time scales by first applying various Gaussian smoothing functions to the data. [Konstantinov and Yoshida 1992] developed a more extensive library of two dozen trend components, based on concatenations of qualitative first and second derivatives. These representations are limited in only handling univariate trends over fixed time slices. Therefore they cannot capture the temporal correlations between variables that trend templates can.

8.3.1.2 Steady State Representation

AIS [Yeh 1991], models and analyzes dynamic systems that at steady state iterate a fixed sequence of actions, or *phases.* Each phase is represented as parameter constraints over a temporal interval; several such meeting intervals form the complete sequence of actions. For example, AIS models a steady state beat sequence of a healthy left ventricle with four intervals. The relevant parameters are the ventricular volume (V) and pressure (P), the pressures of incoming blood from the lungs (P_i) and of output blood to the body (P_o), the heart rate (HR), and the volume functions of pressure at systole (V_s(P, HR)) and diastole (V_d(P)) found in physiology texts. The phases are:

1. *Isovolumetric contraction:* The ventricle contracts, but no blood moves. V [volume] is constant while P increases to P_o
2. *Systolic ejection:* The ventricle continues contracting, but now blood is ejected out to the body. P is constant while V decreases to $V_s(P_o,HR)$
3. *Isovolumetric relaxation:* The ventricle relaxes and blood movement stops. V is constant and P decreases to P_i.
4. *Diastolic filling:* The ventricle continues relaxation and blood enters from the lungs. P is constant and V increases to $V_d(P_i)$, *Return to phase 1* [Yeh 1991, page 319].

These phase descriptions can in fact be expressed as trend templates (which are somewhat more expressive) and can even be used for monitoring the actions of an individual patient (though this is unrealistic in the ventricle model). AIS does not monitor but instead generates global parametric constraints holding over the entire process, a task for which its more specialized representation is needed.

8.3.2 Predicates on Parameter Values Over Time

The Skeletal Dysplasia Diagnostician (SDD) [Keravnou and Washbrook 1990] uses a temporal representation similar to trend templates. Bone disorders are represented in "contexts" that are patterns of temporal constraints on findings. Features

are represented as qualitative states of physical findings. For example, below is part of the context for metatropic dwarfism

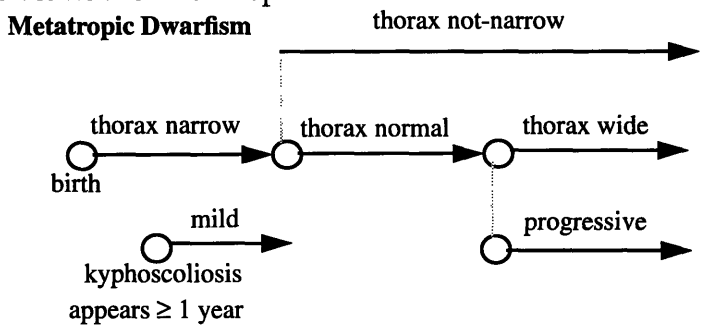


Figure 44 Part of a disorder context in the Skeletal Dysplasia Diagnostician.

The SDD disorder representation is more limited than trend templates in that time points cannot be fully flexible, and symptoms are limited to tokens for qualitative states. The authors claim that this representation is customized to the way expert radiologists reason about bone disorders. Like TrenDx SDD diagnoses a patient by matching the findings (entered as symbols) to the disease contexts. In SDD hypotheses are "activated" based on specific findings as in the Present Illness Program [Pauker, Gorry et al. 1976].

Several AI programs have encoded temporal predicates in associative rules to test conditions in a process for a diagnostically significant pattern. Perhaps the first use of such rules is by [Allen and Koomen 1983] in STRIPS-like planning with temporal extents in the "blocks world" domain. A sample rule for stacking one block on another is:

> IF STACK(x,y) occurs over time S_{xy}
> THEN CLEAR(y) holds over time C_y,
> such that S_{xy} *finishes* C_y;
> and ON(x,y) holds over time O_{xy},
> such that S_{xy} *meets* O_{xy};
> and CLEAR(x) holds over time C_x,
> such that S_{xy} *during* C_x [page 742].

The italicized temporal interval relations are in the epistemology of [Allen 1984].

THRIPHT [Kohane 1987], which uses TUP to represent time, applies similar associative rules for inferring a medical patient's disorders and their temporal extent. This rule from THRIPHT can deduce a possible future onset of hepatitis:

> IF begin of jaundice is before now
> AND begin of IV drug abuse is before now
> AND begin of IV drug abuse is before begin of jaundice
> THEN assert that the begin of hepatitis is before now [page 68].

The monitoring program M-HTP [Larizza, Moglia et al. 1992], which monitors heart transplant patient for possible infection, includes abstractions of patient parameters in its rules. For example:

> IF a decrease in platelet count
> *overlaps* a decrease in white blood cell count
> for at least three days
> during the last week,
> THEN suspect a cytomegalovirus (CMV) infection
> DO request a CMV antigenemia test
> request a CMV viremia test [page 121].

A separate portion of M-HTP performs temporal abstraction as described in section 8.4.3 on page 127. The TUP based monitoring program Clinician's Assistant (CA) [Kohane 1992] uses similar rules, and has been applied to monitoring diabetic pediatric patients.

[Winkel 1990] has developed a program for automated therapy monitoring that uses "decision rules" similar to those above. The program can suggest changing a treatment protocol based on current patient data. For example, some rules suggest modifying the frequencies of particular laboratory tests during future temporal intervals based on current lab results.

RX [Blum 1984] is a program that inspects data of medical patients who had received a particular drug and infers potential causal relationships between the drug dosage and any effects on laboratory parameters. One goal of RX is to suggest experimental designs for studies to verify potential causations. Toward this purpose the program represents temporal restrictions on when data should be sampled from the records of patients in the study. For example, a model for a study on the effects of the drug prednisone on serum cholesterol in patients with nephrotic syndrome includes that no parameters should be sampled while the patient has an interacting disorder such as hepatitis or ketoacidosis. The model also mandates that data are spaced wider in time than the estimated delay from the start of nephrotic syndrome to the patient's reaching a steady state of cholesterol.

8.3.3 Temporal Uncertainty of Boolean Propositions

Much research in AI has addressed the more modest representation of the time intervals over which a Boolean (true or false) statement P is true. Both TUP [Kohane 1987] and the Time Map Manger (TMM) [Dean and Boddy 1987; Dean and McDermott 1987] can represent and answer queries about the duration of truth of Boolean propositions. This duration is called the *persistence* of the proposition in TMM papers. Both TUP and TMM can maintain different accounts of the persistence of propositions if there is ambiguity in the knowledge base.

Other AI researchers have addressed a related representation question: what is the certainty at time t of a proposition P being true? [Dean and Kanazawa 1988] have presented a probabilistic estimate of this certainty. They distinguish between two types of temporal entities. *Fluents* are propositions (denoted by P) that, once they become true tend to stay true without additional effort. Thus fluents have some non-zero probabilistic truth over a time interval. *Events* are propositions (denoted by E) that are true at a single time point, and their truth affects the truth of fluents. E_P denotes an event that triggers the fluent P to become true. (P, t) denotes that the fluent P is true at time t and (E,t) denotes that an event of type E occurs at time t. Time is assumed to be discrete and linear, and the time separating any two consecutive time points is $\Delta > 0$. Without enabling or disabling events, one can represent the belief in the truth of P at time t based on the belief in P at time t-Δ:

$$p((P,t)) = p\langle(P,t)|(P,t-\Delta)\rangle \cdot p((P,t-\Delta)) + p\langle(P,t)|\neg(P,t-\Delta)\rangle \cdot p(\neg(P,t-\Delta)) \quad [2]\text{(EQ 24)}$$

The conditional probability $p\langle(P,t)|(P,t-\Delta)\rangle$ is called a *survivor function* in queuing theory. This captures the tendency of a proposition to become false without any explicit contravening effects. An example survivor function is a decreasing exponential function of the time that has elapsed since the last observation.

$$p\langle(P,t)|(P,t-\Delta)\rangle = e^{-\lambda\Delta} \quad \text{(EQ 25)}$$

If we have evidence of specific events that may make P true or false in the future, then (EQ 25) becomes an integral over the times of the enabling or disabling events. These uncertainty calculations have been expanded to a temporal logic and probabilistic network formalism for reasoning about time-dependent propositional truth [Kanazawa 1991].

8.3.4 Trends as Fuzzy Membership Functions

A *fuzzy membership function* describes the belief (from 0 to 1) of a proposition as a function of independent parameters. Definitions and methods for matching data to such membership functions may be found in [Terano, Asai et al. 1992; Zadeh 1965]. We note that fuzzy membership functions may be used for monitoring trends in data if the independent parameters include time and data values.

A fuzzy logic based representation of propositional uncertainty is presented in [Console, Rivolin et al. 1991] as part of an epistemology for causal and temporal reasoning in diagnostic knowledge-based systems. A finding is represented as possibly true over a temporal interval using a fuzzy membership function. The diagnosis program insists that a hypothesis is valid only if there exists a consistent temporal extent for each finding.

---

2. Note that $\neg(P,t) \equiv (\neg P,t)$.

Other researchers have defined trends based on time-varying fuzzy membership functions of measured parameters. [Steimann 1994; Steimann and Adlassnig 1994] have represented recent trends in respiratory parameters commonly found in an intensive care unit. Recall that $Sa0_2$ denotes percentage oxygen saturation of hemoglobin. The trend in the figure below defines a trend of "adequate oxygenation" as $Sa0_2$ above 97% (with minimum of 93%) for the last 5 minutes. The belief in this trend is 1 if a datum is between 97% and 100%. Belief decreases linearly to 0 for data valued from 97% to 93% The belief in this trend holding for a set of data over the past 5 minutes is the minimum of the belief for all data.

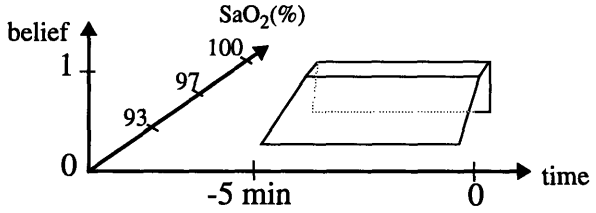


Figure 45 Fuzzy membership function for "adequate oxygenation." Reproduced from [Steimann 1994] with permission of author.

The monitoring system DIAMON-1 embeds these trend definitions within a fuzzy state transition diagram to define when the patient is in a clinical state worthy of sending an alert. State transitions are based on the belief in certain trend predicates or boolean combinations thereof. The program can generate beliefs in various patient states over time.

[Sittig, Cheung et al. 1992] estimate the duration of long term hemodynamic trends using fuzzy membership functions of regression slopes. If the slope of a linear regression on recent data is within the same fuzzy category as the slope found from more distant data, the duration of the long term trend is extended to cover the recent data.

Fuzzy membership functions are a possible alternative for encoding uncertainty in trend template value constraints. The DIAMON-1 trend for adequate oxygenation captures temporal and value uncertainty, as does a trend template embedded within an interval of uncertain duration. Value constraints differ in employing least-squares regression and measuring match strength by mean absolute percentage error rather than belief. Use of fuzzy membership functions could allow TrenDx hypotheses to be assigned a "belief" in fitting the data, rather than a percentage error score.

One consideration in modeling with fuzzy membership functions is determining the belief surfaces and the key boundary (or "fiduciary") points. These are usually knowledge engineered with experts; research into learning the boundaries and beliefs from data using neural networks is in [Jang 1993]. Another concern is whether beliefs in competing trends can be meaningfully compared. Statistical

(e.g. MAPE) or probabilistic approaches offer a uniform, consistent meaning of a trend score that is not guaranteed by the more subjective belief surfaces.

8.4 Knowledge-Based Monitoring Algorithms

8.4.1 Model-Based Process Monitoring

Much of the work in intelligent monitoring has applied model-based reasoning. One class of work includes combined qualitative and quantitative simulation. SIMON [Uckun and Dawant 1992] is a monitoring system for intensive care units that extends Qualitative Process Theory [Forbus 1985] by incorporating numerical values. SIMON can predict the direction of change and the next landmark value for independent quantitative parameters of the qualitative model. MIMIC [Dvorak and Kuipers 1989] monitored disorders in simple mechanical devices (e.g. ideal water heater) using the qualitative simulation program QSIM supplemented with quantitative time and value ranges [Kuipers and Berleant 1988]. MIMIC uses these ranges to generate "dynamic envelopes" for what values parameters may take at a given time. PREMON [Doyle, Sellers et al. 1989] applies causal models to diagnose faults in aerospace systems. Each causal links describes how a change in value of one parameter results in a change over time in value of another parameter. Because in aerospace systems not all sensors can be continually checked, PREMON gives priority to sensors most richly connected in the causal model.

Other model-based monitoring systems represent the underlying process as a Bayesian probabilistic network. Although a probabilistic network usually presents a static model of a process, it can be used for monitoring by updating evidence nodes denoting measurable parameters as new data are received. Then probabilities are propagated to better estimate the likelihood of competing diagnoses; if an estimate for a dangerous diagnosis exceeds a threshold, an alarm is sent. In this spirit the ALARM system [Beinlich, Suermondt et al. 1989] monitors anaesthesia effects using a cardiac and pulmonary model. VentPlan [Rutledge, Andersen et al. 1990], a similar belief network for monitoring patients on a mechanical ventilator, has the added feature of a temporal model for updating probabilities of qualitative parameter values. If the value of parameter P is "low" at time t, then at time $t + \delta$, where δ is the half-life of P, VentPlan can deduce that P will likely be low but has a small probability of being normal or high. Guardian, a prototype ventilator management monitor for the surgical intensive care unit, alternates between reasoning by belief net and by an anatomical model depending on the urgency of the patient's condition. Guardian is noteworthy in its method of filtering input data [Washington and Hayes-Roth 1989], changing the sampling rate and filter width dynamically depending on time resources. A belief network for monitoring blood cell counts in uremic patients treated with erythropoietin [Berzuini, Bellazzi et al. 1992] encodes pharmacokinetic models, the parameters of which have time-varying probability

distributions. The parameter distributions learned for other patients are also represented and used for monitoring the current patient.

Model-based monitoring approaches are applicable only in domains where one can construct a causal model of the monitored process. TrenDx does not impose such a requirement. Trend templates may also be a useful addition to monitors based on qualitative or semi-quantitative simulation. The templates can indicate which sets of future qualitative states correspond to the same trend hypothesis. This may help to reduce branching of behaviors and thus improve monitoring efficiency.

8.4.2 State-Based Process Monitoring

Other intelligent monitoring programs model the process as a finite state machine. Each state corresponds to a diagnosis, and the arcs represent transitions made between diagnoses in the event of certain patterns of data. VM [Fagan, Kunz et al. 1984] is an early monitoring program for ventilator management in which states denote patient conditions like "stable hemodynamics" or "oxygen deficient." In VM associative rules determine the transition of the patient between diagnostic states. Rule antecedents include the current patient state and laboratory measurement judgements such as "heart rate is *acceptable,*" where "acceptable" is determined by the current patient state. Rule consequents are changes to another patient state.

More recently [Nelson and Hadden 1992] have applied finite state machines to trend detection in monitoring of sensors on the space station Freedom. Their technique of state-based feature recognition (SBFR) models trends as finite-state machines. For example, an "increasing" trend representation includes a begin state named "reset" an end state named "increase trend," and two intermediate states named "possible increase." Arcs guide transitions between states based on the average velocity between consecutive measurements. State machines may themselves be used as arcs in "higher level" state machines for process faults. Thus the machine for "control moment gyroscope bearing failure" includes an arc testing for "increasing trend in bearing temperature." As we have mentioned in section 3.6.4 on page 47, trend templates can be extended to have a similar hierarchical structure.

8.4.3 Temporal Abstraction of Data to Qualitative States

A related problem to diagnostic process monitoring is temporal abstraction of time-ordered data into intervals characterized by qualitative categories. For example, a program can abstract from the data in Figure 46 that from 8:00 to 10:00 serum Na^+ was decreasing and from12:00 to 2:00 Na^+ was increasing. Probably the program can make no abstraction from 10:00 to 12:00. The difficulty of sound

abstraction increases as the data grow farther apart in time and more uncertain in value.

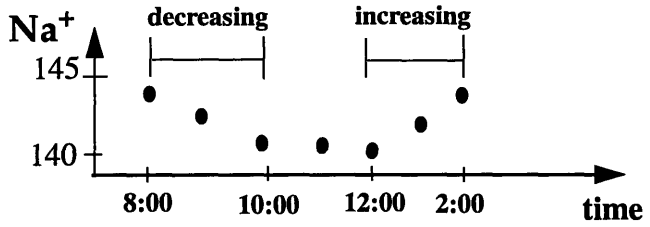


Figure 46 Temporal abstraction.

A simple, domain independent temporal abstraction mechanism is for the program to detect *runs* of similar, or similarly changing values, and abstract over the interval spanning the length of the run. [Kahn, Abrams et al. 1990] compared various criteria for run detection in a program that segmented a diabetic patient's log book into regions in which he or she is taking equivalent insulin doses.

Domain dependent temporal abstraction allows the knowledge engineer to encode properties of the data that dictate the rules for abstraction. [Russ 1993] outlined an epistemology for temporal abstraction for his Temporal Control Structure (TCS). Russ implemented a two point temporal abstractor based on the persistence of data values. RESUME [Shahar, Tu et al. 1992] abstracts data points into intervals and subsequently combines abstracted intervals into clinical patterns of data. The rules of combination are based in part on the ontology of temporal properties in [Shoham 1988]. Also needed for combining intervals are knowledge engineered distances Δ giving the maximum gap between consecutive data times or values for which abstraction is allowed. The Δ's implicitly model parameter uncertainty or persistence. The temporal abstractions generated by RESUME are sound in terms of measurement semantics. To minimize irrelevant temporal abstractions, a temporal abstractor must be supplemented with a descriptive language of what patterns of intervals are clinically significant, including trends. RESUME interacts with a temporal query system [Das, Tu et al. 1994] that asks for significant abstractions. [Kohane and Haimowitz 1993] argue that trend templates are also a language of significant temporal abstractions.

TOPAZ [Kahn, Fagan et al. 1991] abstracts data of chemotherapy patients into intervals where the measured values did or did not meet the predicted values of a pharmacokinetic model-based reasoner. TOPAZ generates text descriptions of remarkable clinical events such as the following:

> a series of 13 visits after starting chemotherapy, starting on 30 November 1983 and ending on 24 May 1984, with predicted WBC counts that were systematically lower than expected for this period [Kahn, Fagan et al. 1991, page 169].

Summaries like this are useful in filtering and highlighting the subset of the data that is worthy of attention.

8.4.4 Temporal Truth Maintenance

In some clinical settings certain data are reported significantly later than they are measured. For example, in an intensive care unit, blood tests may arrive up to an hour after they are measured. When these data arrive, they report values of measurements in the past. These past values may refute the conclusions that a monitoring system made in the past, as well as the consequences of these conclusions.

These issues are quite relevant to TrenDx, which processes data in order of arrival. TrenDx uses the best scoring hypotheses of competing trend templates to send alarms or trigger other monitor sets. Should data from the past arrive and markedly change a hypothesis score, the alarms or triggers generated by TrenDx based on past patient status may need to be reconsidered.

An automated monitor could benefit from a *truth maintenance system* that can modify the temporal extent of conclusions and change the temporal persistence of variables. The Temporal Control Structure (TCS) [Russ 1985; Russ 1993] manages and updates such information, and was used in a prototype expert system for monitoring patients with diabetic ketoacidosis. TCS models a process as a network of *modules*, each of which update the truth of propositions or the values of variables over time. The inputs and outputs of various modules are interconnected, and have temporal extent. Therefore, newly arrived data in some modules may affect the duration of conclusions in other modules.

Figure 47 outlines a TCS process that could add temporal truth maintenance to TrenDx. We consider a simple case of a single active monitor set with two trend templates: a normal trend TT_N and a fault trend TT_F. Processes and data flow are gray for TT_N and thick black for TT_F. For each trend template the B (beam size) different hypotheses are shown. Each hypothesis includes as inputs the temporal distances Δ_i between points in its trend template. Note that TCS does not handle temporal uncertainty and so the temporal distances must be determined using another temporal reasoner such as TUP. These inputs are used by the hypothesis modules to compute assignments of data to intervals and matching scores.

At left, a new data cluster D_t arrives; it may include current or past measurements. D_t is routed to each active hypothesis for both trend templates. The hypothesis modules update their matching scores. The two "best score" modules pass the best match score for each hypothesis, as well as the time of this data cluster, to each of the decision modules at right. These decision modules have internally a history of previous best scores, and an algorithm for deciding when a trend is significant, when to send an alarm, and when to trigger other monitor sets. Based on these new best scores and the cluster time, these modules may change the duration of a trend's significance, or rescind an alarm or trigger.

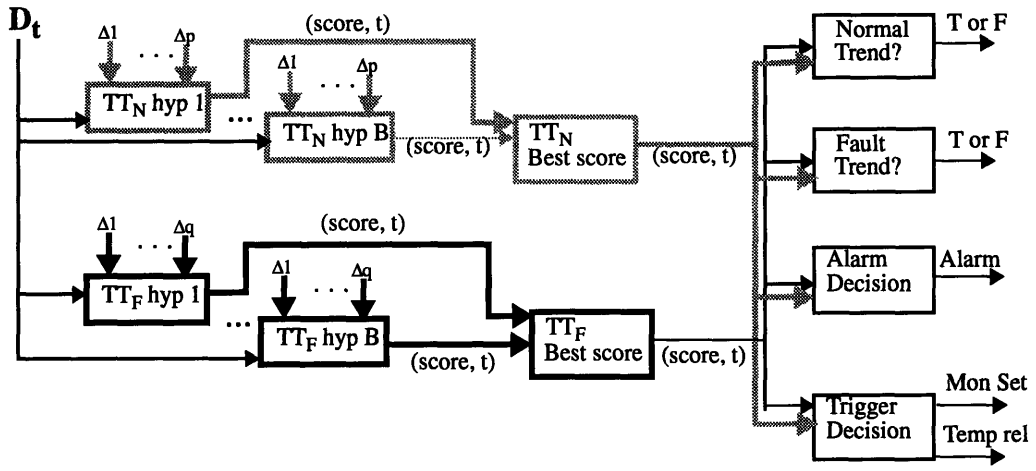


Figure 47 TCS process for TrenDx chain of reasoning about a monitor set.

The process network in section Figure 47 on page 130 does not account for the beam search in TrenDx. A datum measured in the past may markedly increase the score of an older hypotheses that TrenDx pruned. If so, the decision to prune that hypothesis in the past may have been mistaken, with possible consequences for what alarms were sent or what monitor sets were triggered. If some hypotheses pruned from matching were retained, then in principle TCS may be used to reinstate these older hypotheses. This would add another layer of detail to the process network in the figure. Retaining *too many* stale hypotheses may use up space resources, the scarcity of which was an original motivation for beam search.

This chapter has illustrated that TrenDx extends and combines several bodies of research. Temporal reasoning work in artificial intelligence has rarely addressed monitoring hundreds of data points, or combinations of temporal and value uncertainty. AI diagnostic systems have typically not reasoned with symptoms from time-ordered data. System identification work has rarely been subjected to constraints of lack of training data and the need for explanation. Out of these necessities came inspiration for this research.

9 *Conclusions*

9.1 Summary

The preceding chapters have demonstrated particular trend templates and TrenDx matching algorithms, as well as results in matching these trend templates to clinical data. Those chapters stand on their own merits. In this concluding chapter, we step back and consider general lessons learned from this research on knowledge-based diagnostic monitoring.

9.1.1 Combining Models in Two Axes Can Discriminate Trends

TrenDx has demonstrated that by combining weak methods of uncertainty in both the temporal and the value axes, a matching algorithm can discriminate between competing trends. Neither the polynomial regression-based TrenDx or the constraint-based TrenDx relies on precise matches to clinical data; therefore these methods can produce only rough predictions of future data values. However, the representation captures essential temporal and value differences between trends so that the *relative accuracy* of matches can be used for diagnosis.

9.1.2 Framework for Diagnostic Monitoring

In chapter 6 we emphasized that TrenDx naturally fits into a larger framework for automated diagnostic monitoring. The structure of trend templates, including landmark points and time intervals, models the time-evolving patterns of process variables. This correspondence facilitates knowledge engineering. Furthermore, grouping of trend templates into monitor sets leads to straightforward, automated rules for checking consistency of competing trend templates (section 6.5) and learning temporal distances (section 6.6). The matching algorithm scores the match to each trend template as a weighted average of matching scores to each phase of the trend. (chapter 4). Competing hypotheses are compared based on their average percentage error in matching the data. The comparison may be over recent time intervals or over an entire process lifetime.

The scores of competing trends can drive other actions that make use of the trend template structure. An automated monitor sends *alarms* if a faulty diagnosis is highly ranked (section 6.2). The monitor may also display portions of data when faults are present, or summarize faulty data over a long time period (section 6.3). The monitor can make its alarms and displays include only *relevant* times and data by exploiting the structure of the best-matching trend template. A faulty diagnosis in one monitor set can be used to switch clinical context to other monitor sets (section 6.4).

9.1.3 Flexible Representation and Pattern Matcher

The trend template representation and associated TrenDx matching algorithms are flexible and can accommodate different levels of modeling detail. We have demonstrated this by showing both a polynomial regression-based TrenDx and a constraint-based TrenDx. Still other versions of trend templates and TrenDx are possible. For example, one could incorporate more complex stochastic models within value constraints of trend templates and still stay within the spirit of the representation. In some econometric trend templates value constraint functions could remove cyclical and seasonal components, and the remaining signal could be modeled as both a trend and autoregressive unknown variation. As another example, if an expert can reliably characterize beliefs in a trend for different regions of values and times, then value constraints may consist of fuzzy membership functions.

TrenDx could match data to either new variety of trend templates with only slight modifications. However, one should only expend the extra statistical training or extra knowledge engineering as required for discriminating alternate diagnoses.

9.1.4 Broad Applicability

We have demonstrated the potential use of TrenDx in two very different diagnostic monitoring applications. Pediatric growth monitoring is characterized by a few key measurements with a sample period ranging from a few months to over a year. Intensive care unit (ICU) monitoring involves tracking dozens of parameters with sample period under a minute. The common concern in both of these domains is *data overload*: neither the general pediatrician nor the ICU nurse has enough time to consider the entire sequence of time ordered data. Consequently, in each domain the experts need reliable interpretations and concise, understandable summaries of when these data have changed with clinical significance. In the next section we identify computational aspects of a domain that make TrenDx applicable.

9.2 When TrenDx is Applicable

TrenDx is particularly applicable in domains meeting most of these criteria:

- Experts can identify a manageable number of clinical contexts in which the monitored process can be. In each such context, an expert can identify a set of competing trends in monitored data. One of these trends is the expected pattern of data in that situation. The clinical contexts are inter-related according to a state transition framework. Transitions may be either clearly understood in the domain or based on abnormal trends in the data.
- The underlying model generating the process data is not well understood quantitatively. Thus one cannot readily build a stochastic or differential equation model that accurately predicts subsequent data values from past ones.
- There are not enough training cases for all competing diagnoses to adequately estimate parameters of quantitative models.
- Trends are naturally thought of as consisting of important process events and meaningful phases, Experts wish to know at what times these events and phases took place, or to inspect the data nearby these events and phases.
- There is enough empirical evidence to distinguish competing trends within a clinical context according to the temporal extents of, and data variation within, the trends' phases.
- The temporal and data variation within each phase is somewhat uncertain.

Each criterion lends itself to the TrenDx approach to diagnostic monitoring. TrenDx approach lets one quantify temporal and value uncertainty either through training data or through empirical knowledge of experts.

9.3 Future Work

9.3.1 Large-Scale Clinical Trial

The experimental results on pediatric growth monitoring in section 5.1 demonstrated the potential of TrenDx diagnosing some growth trends. We have begun to design a controlled, large-scale clinical trial to better evaluate TrenDx as a screening tool in a general pediatrician's office. We will test the sensitivity and specificity of different versions of TrenDx on hundreds of patients and compare the results to those of diagnosticians with varying expertise in pediatric growth.

Currently we are collecting a large case mix of patients, including healthy children of average, delayed, or advanced pubertal onset. Also included will be children with hormonal, nutritional, and genetic disorders. Our records will include the heights, weights, bone ages, and pubertal stages of the children.

9.3.1.1 Issues in Evaluating Monitoring Systems[1]

The traditional method for evaluating the efficacy of a medical device, including a computer program, is a *field trial,* i.e. to place it within clinical practice. Then patient care may be evaluated both before and after the device was incorporated [Rind et. al. 1991]. Such experiments are difficult to achieve for exploratory trials of artificial intelligence-based consultation systems. [Wyatt and Spiegelhalter 1992] have noted that before field trials one must independently assess the technical accuracy (e.g. sensitivity and specificity) of the computer program. [Forsythe and Buchanan 1992] have noted that even in such a preliminary trial, one must put careful thought into a variety of issues, including:

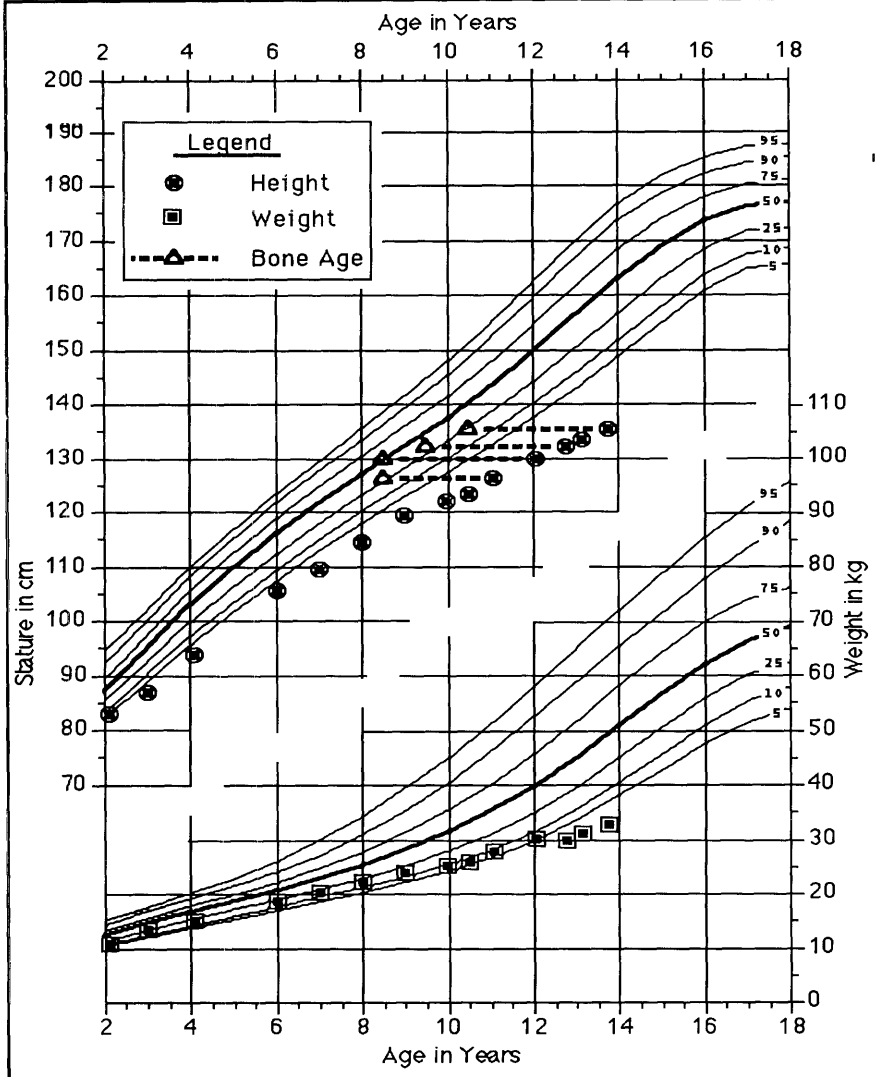
- *What is the test task?* Inherently the computer program cannot interpret all visual, oral, and textual data that a clinician can. One approach is to restrict both computer and human subjects to a limited task of numerical data to "level the playing field." However, this makes difficult a direct comparison between program and actual practice.
- *What is the gold standard?* A program may be directly compared to the diagnosis or recommendation in the patient record. Alternatively experts may merely comment as to the plausibility of the program's output [Bankowitz, Lave et al. 1992]. Another option is for expert evaluators to perform the same task as the program [Feldman and Barnett 1990]. This latter method was used in the trial of constraint-based TrenDx (section 7.1.5).

9.3.1.2 Proposed Method

We are still debating the nature of our gold standard for this experiment. We do plan to directly compare the results of at least two versions of TrenDx to human subjects of varying expertise in pediatric growth monitoring. These TrenDx programs will have expanded knowledge bases of normal and abnormal growth trends. The use of diagnosticians of varying expertise will allow more accurate characterization of TrenDx performance. Our hypothesis is that at least one version of TrenDx will be as sensitive and specific as a general pediatrician, and also better than medical students with some pediatric rotation experience.

Figure 48 illustrates the experimental test form to be given to human subjects. Heights, weights, bone ages, and pubertal data will be noted. The data of each patient will be displayed on multiple sheets of paper, one sheet per pediatrician visit. The human subjects may receive the additional benefit of a text summary of the patient history through each visit.

1. This section written with assistance from Phillip Phuc Le.



Patient 2

Age	Height	Weight	Bone Age
2.1	83	11	
3	87	13.8	
4.1	94	15.5	
6	105.8	19	
7	109.7	20.5	
8	114.5	22.5	
9	119.5	24.3	
10	122	25.5	
10.5	123.5	26.3	
11.05	126.2	28.2	8.5
12.03	129.9	30.4	8.5
12.75	132.2	30.2	9.5
13.16	133.5	31.4	
13.75	135.6	33	10.5

Please rank the diagnoses you believe are appropriate:

__ Normal
__ Normal, short stature
__ Obese
__ Malnourished
__ Constitutional Delay
__ Growth Hormone Deficiency
__ Early Puberty
__ Precocious Puberty

Figure 48 Subject test form for large-scale growth experiment. Data courtesy of Boston Children's Hospital. Graphics courtesy of Phillip Phuc Le.

9.3.2 Other Application Domains

We have suggested qualities that may make a monitoring application conducive to using TrenDx (section 9.2). However, only actual tests in other settings can verify broad applicability. As we suggested in section 2.3, other potential applications include industrial process control, power plants, econometrics, and finance.

9.3.3 Toolkit for Trend Detection

Currently the interface for building trend templates and running TrenDx is entirely based on a programming language built in Common Lisp. The knowledge engineer encodes s-expressions defining landmark points, intervals, and trends. Data are entered via s-expressions that include the patient record number, time, parameter, and value. The entire modeling and monitoring environment could more easily be adopted with a graphical interface for creating trend templates, and a means of matching data stored in standard database and spreadsheet formats.

9.3.4 Data Uncertainty

TrenDx assumes that all data are time-stamped and completely precise. However in many monitoring domains the data may be uncertain. This uncertainty can be of at least two types. One is that the *value* is uncertain, perhaps due to measurement error. The other is that the *time* is uncertain because the data were not gathered at the time of occurrence (as in patient history).

Value Uncertainty

One may take either of two approaches in modeling the value uncertainty of data given to TrenDx. One is to assign *standard errors* of measurement to each laboratory parameter, and then compute *value intervals* for each datum. For example, if height is a parameter with standard error of 3 millimeters (see section 1.3.2) and a height datum reads 160.5 centimeters, then TrenDx will compute a value interval of [160.2 160.8] centimeters for that datum. Consequently, the regression fit in value constraint evaluation will be an error minimization through a *space of variable values*.

Another representation of data value uncertainty is to provide a probability distribution, most likely normal, for each datum. The standard deviation may be identical to the parameter's standard error. For example, the above height may be assigned a normal distribution with mean 160.5 and standard deviation 1.5 millimeters. With this representation the regression fit in value constraint evaluation will be a weighted average of errors, where the weights come from the probability distributions of the variables within an interval.

Temporal Uncertainty

Most automated monitoring programs assume that all data are time stamped. In pediatric growth monitoring, where family history may provide important findings, this assumption may need to be relaxed. The Temporal Utility Package (TUP) [Kohane 1987] was partially motivated by uncertain timing of findings as reported by patients. For example, the menarche of a female patient's mother is important in assessing the expected onset of puberty of the girl. Usually mothers can approximate this finding to within one year.

TrenDx could represent temporal uncertainty of findings just as TUP did, with a time point offset by a (min max) range from an anchor point. For example, a mother's menarche may be represented as (11 years, 12 years) after her birth. Matching temporally uncertain findings to intervals in a trend template will be only slightly more complex than for time-stamped data. The branching hypotheses issues in section 4.3.3 are identical.

9.3.5 Recommending Tests

When the differential diagnosis of a patient includes two or more closely ranked disorders, an expert diagnostician will order a laboratory test whose results could distinguish between them. TrenDx may be extended to automatically recommend laboratory tests in these circumstances.

The means for this should lie in the structure of trend templates. The test to be recommended in distinguishing between two competing hypotheses is the parameter that is constrained differently in their trend templates at the current time. For example, consider the trend templates for average normal growth and constitutional delay. During middle childhood (ages 3 until puberty) the templates differ in two parameters. In constitutional delay a different set of growth standards is used, and the bone age is delayed. In this case TrenDx could first suggest the new standard (the cost is only time and paper) and then possibly suggest a bone age x-ray also (a more expensive yet still noninvasive test).

Two important research issues arise. One is how to determine if a given parameter is constrained the same or differently in two trend templates. This can be automatically inferred from the trend template structure. Another issue is representing the expected utility of each test. which includes the discriminatory power. Similar issues were addressed by the Acute Renal Failure program [Betaque and Gorry 1971].

9.3.6 Multiple Disorders

Each trend template described in this thesis represents a diagnostic class of patients with a single disorder. TrenDx could diagnose a disease combination only with an explicit representation in a trend template of data variation in patients with

that combination. An alternative is to diagnose multiple disorders by somehow combining the effects predicted by the trend templates for the individual disorders. ABEL [Patil 1981] diagnosed combined acid-base and electrolyte disorders by using the additive effect of single disorders on serum electrolyte levels. A similar technique is qualitative superposition [Coiera 1990] of single disease histories to produce a multiple disease history. However, computing the combined effects are much more difficult if one cannot produce a causal or qualitative model of the underlying domain.

9.3.7 Integrating Text and Graphics

Clearer, more persuasive alarms require presentation with a combination of text and graphics. TOPAZ [Kahn, Fagan et al. 1991] generated natural language text in its summaries of time-ordered data (see section 8.4.3). The program used an augmented transition net formalism with schemas similar to those of the TEXT program [McKeown 1985]. ANA [Kukich 1983] generates three paragraph summaries of stock exchange activities from data in the Dow Jones News Service. The program used a context-dependent grammar that incorporated domain knowledge. The example below of an ANA summary illustrates that domain knowledge can help in determining what should be described:[2]

> After drifting downward through most of the trading session, the stock market finished on the minus side. Trading was active yesterday.
>
> The Dow Jones average of 30 industrials dropped 3.2 points to 1188.27, as the transportation and utility indexes both edged downwards.
>
> Volume on the big board was 106170000 shares compared with 110240000 shares on Wednesday. Big board losers held a slim lead, with 841 stocks down and 811 stocks up. [page 111].

Other interesting issues related to text generation of trends include describing durations of uncertain temporal intervals [Nakhimovsky 1988], and generating the appropriate tense based on the relationship between the time of the event and the time of the utterance [Passonneau 1988]. Both of these challenges are illustrated in the following statement from a possible TrenDx alert:

> For approximately the last two years the patient has lost height centiles.

A graphical plot showing the data of the alarming trend could improve a generated alarm. A picture of a temporal interval surrounding the data may serve to highlight this region (see figure below)

2. I have added proper capitalization to the actual thesis output

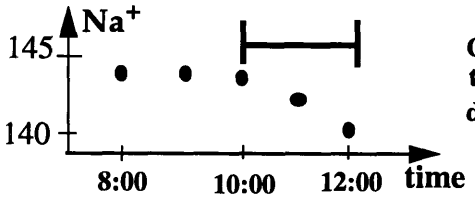


Figure 49 Portion of possible future TrenDx trend alarm.

Pediatric growth alarms may display the growth chart standards within the alarm. As discussed in section 6.3, the interval structure of trend templates can be used to determine which data to display with the text.

If TrenDx alarms may be read by different audiences, the program could incorporate user models [Kass and Finin 1988] to aid textual output. For example, parents may react more anxiously than pediatricians to diagnoses of growth disorders. Responses may need to be empathetic to the individual [Haimowitz 1991].

9.3.8 Subjective Trend Evaluation

As a single agent, ultimately TrenDx represents one view of what trends in data are clinically significant. This view may differ across individuals, and even across computer programs if the subjectivity is explicitly represented. Psychological literature [Friedman 1990] has determined that people estimate the length of time intervals differently depending on factors including:

- the number of stimuli or "key mental processes" in the interval,
- whether something pleasing is expected at the end of the interval.

These psychological results may be applied to encoding the judgement of a series of data as a clinically significant trend. In pediatric growth charts, the judgment of whether the trend is significant and what tests to perform may depend in part on the frequency of measurements, how soon one expects puberty, etc.

9.4 On Modeling Natural Phenomena

This thesis is valuable for computer science in applying innovative artificial intelligence techniques to diagnoses of clinically significant trends. The trend template parameters have been estimated based on published clinical literature, estimates of expert clinicians, and studies of actual patient cases. The TrenDx matching algorithm is motivated in part by how I have observed clinicians reasoning about time-ordered data.

Many artificial intelligence projects have posted impressive theoretical results, though motivated by simple, even contrived problems. These results generally

have difficulty scaling up to large applied problems. This thesis research has taken the approach in *first observing* a natural, scientific disciplines and *then formulating* representations and algorithms. In this respect I follow one of Allen Newell's maxims for a dedicated scientific life [Laird 1992, page 36]:

> *A scientist is a transducer from nature to theory.*
> *Seek out nature and listen to her.*

Appendix A Pediatric Growth Monitor Sets

Below is the computer code that TrenDx interprets to create monitor sets for pediatric growth. There are two monitor sets, one for boys and one for girls. Each monitor set consists of three competing trend templates: average growth, constitutional delay, and early puberty. The monitor set definitions follow the definitions of their member trend templates. These monitor sets are also described in section 5.1.

The code is written in a trend template definition language, implemented on top of the Common Lisp Object System (CLOS) {Steele, 1990 #242}. The higher level trend template language was developed for compact, readable encoding of trend templates. Commands beginning with "`def`" define a trend template object. For example, `deftt` defines a trend template, and `definterval` defines an interval within that trend template.

Value constraint definitions include some subset of the following: name, function to apply to data within an interval, parameters (i.e. types of measurements) that the function is applied to, and a model: a low order polynomial with parameters estimated either qualitatively or quantitatively. For example, a constraint `'(constant *)` means constant at any level, while `'(constant 2)` denotes constant at value 2. A constraint that includes `D1` or `D2` constrains the first or second derivative of a polynomial model.

A.1 Boy Growth Monitor Set

```
(deftt "boy-average-growth"
  :landmarks '(birth puberty-onset growth-stops peak-ht-velocity)
  :intervals  (list (definterval :name "early-childhood"
                    :constraints
                    (list  (defconstraint :name "constant build"
                           :func #'bd-infant-normal
                           :parameters '(weight height)
                           :model '(constant *))
                          (defconstraint :name "height weight co-vary"
                           :func #'wt-minus-ht-infant-zscores
                           :parameters '(weight height)
```

```
        :model '(constant *))))
(definterval  :name "pre-puberty"
 :constraints
 (list  (defconstraint :name "maintain height centiles"
        :func #'avg-sd
        :parameters '(height)
        :model '(constant *))
       (defconstraint  :name "maintain weight centiles"
        :func #'child-nchs-sd
        :parameters '(weight)
        :model '(constant *))
       ))
(definterval :name "linear-growth"
 :constraints
 (list (defconstraint :name "normal bone age"
        :func  #'chron-age-minus-bone-age
        :parameters '(bone-age)
        :model '(constant 0))))
(definterval :name "tanner-1"
 :constraints
 (list (defconstraint  :name "tanner 1"
        :parameters '(pubertal-stage)
        :model '(constant 1))))
(definterval :name "tanner-2"
 :constraints
 (list (defconstraint  :name "tanner 2"
        :parameters '(pubertal-stage)
        :model '(constant 2))))
(definterval :name "tanner-3"
 :constraints
 (list (defconstraint  :name "tanner 3"
        :parameters '(pubertal-stage)
        :model '(constant 3))))
(definterval :name "tanner-4"
 :constraints
 (list (defconstraint  :name "tanner 4"
        :parameters '(pubertal-stage)
        :model '(constant 4))))
(definterval :name "tanner-5"
 :constraints
 (list (defconstraint  :name "tanner 5"
        :parameters '(pubertal-stage)
        :model '(constant 5))))
(definterval :name "growth-spurt-rise"
 :constraints
 (list (defconstraint :name "height-acceleration"
        :parameters '(height)
```

```
                    :model '(quadratic (D1 +) (D2 +)))))
                (definterval :name "growth-spurt-fall"
                 :constraints
                 (list (defconstraint :name "height-deceleration"
                        :parameters '(height)
                        :model '(quadratic (D1 +) (D2 -)))))
                (definterval :name "linear-growth-ends"
                 ;; height is constant for the next years.
                 :constraints
                 (list (defconstraint :name "height-constant"
                        :parameters '(height)
                        :model '(constant *))))
                )
        :relations
         '((birth growth-stops (years 17) (years 19))
           ;; Tanner stage intervals span childhood.
           (birth (begin tanner-1) 0 0)
           (consecutive-phase tanner-1 tanner-2)
           (consecutive-phase tanner-2 tanner-3)
           (consecutive-phase tanner-3 tanner-4)
           (consecutive-phase tanner-4 tanner-5)
           (growth-stops (end tanner-5) 0 0)
           ;; From Tanner and Davies charts
           (birth (begin tanner-2) (years 10) (years 13))
           (birth (begin tanner-5) (years 11.5) (years 14.5))
           ((begin early-childhood) birth 0 0)
           ((begin early-childhood) (end early-childhood) (years 2.05) (years 2.95))
           (consecutive-phase early-childhood pre-puberty)
           (puberty-onset (end pre-puberty) 0 0)
           ((begin linear-growth) birth 0 0)
           ((end linear-growth) growth-stops 0 0)
           (consecutive-phase linear-growth linear-growth-ends)
           ((begin linear-growth-ends) (end linear-growth-ends) (years 5) (years 5))
           ;; Tanner and Davies charts constrain the growth spurt interval:
           (birth (begin growth-spurt-rise) (years 10) (years 12))
           (consecutive-phase growth-spurt-rise growth-spurt-fall)
           ;; By definition peak ht velocity is at transition point between two
           ;; growth spurt intervals:
           (peak-ht-velocity (end  growth-spurt-rise) 0 0)
           ;; Puberty onset is equal to onset of tanner 2:
           (puberty-onset (begin tanner-2) 0 0)
           ;; Peak ht velocity within the growth spurt, from Tanner and Davies charts:
           (birth peak-ht-velocity (years 12.5) (years 14.5))
           (growth-stops (end growth-spurt-fall) 0 0)))

        (deftt  "boy-constitutional-delay"
         :landmarks '(birth puberty-onset growth-stops peak-ht-velocity)
```

```
:intervals  (list (definterval :name "early-childhood"
                    :constraints
                    (list  (defconstraint :name "constant build"
                             :func #'bd-infant-normal
                             :parameters '(weight height)
                             :model '(constant *))
                           (defconstraint :name "height weight co-vary"
                             :func #'wt-minus-ht-infant-zscores
                             :parameters '(weight height)
                             :model '(constant *))))
                  (definterval  :name "pre-puberty"
                    :constraints
                    (list  (defconstraint :name "maintain height delayed centiles"
                             :func #'late-sd
                             :parameters '(height)
                             :model '(constant *))
                           (defconstraint  :name "maintain weight centiles"
                             :func #'child-nchs-sd
                             :parameters '(weight)
                             :model '(constant *))
                           (defconstraint
                            :parameters '(bone-age)
                            :func #'chron-age-minus-bone-age
                            :model '(constant 2))
                           ))
                  (definterval :name "linear-growth"
                    )
                  (definterval :name "tanner-1"
                    :constraints
                    (list (defconstraint  :name "tanner 1"
                            :parameters '(pubertal-stage)
                            :model '(constant 1))))
                  (definterval :name "tanner-2"
                    :constraints
                    (list (defconstraint  :name "tanner 2"
                            :parameters '(pubertal-stage)
                            :model '(constant 2))))
                  (definterval :name "tanner-3"
                    :constraints
                    (list (defconstraint  :name "tanner 3"
                            :parameters '(pubertal-stage)
                            :model '(constant 3))))
                  (definterval :name "tanner-4"
                    :constraints
                    (list (defconstraint  :name "tanner 4"
                            :parameters '(pubertal-stage)
                            :model '(constant 4))))
```

```
                (definterval :name "tanner-5"
                 :constraints
                 (list (defconstraint  :name "tanner 5"
                        :parameters '(pubertal-stage)
                        :model '(constant 5))))
                (definterval :name "growth-spurt-rise"
                 :constraints
                 (list (defconstraint :name "height-acceleration"
                        :parameters '(height)
                       :model '(quadratic (D1 +) (D2 +)))
                      (defconstraint
                        :parameters '(bone-age)
                        :func #'chron-age-minus-bone-age
                        :model '(constant 1))))
                (definterval :name "growth-spurt-fall"
                 :constraints
                 (list (defconstraint :name "height-deceleration"
                        :parameters '(height)
                       :model '(quadratic (D1 +) (D2 -)))
                      (defconstraint
                        :parameters '(bone-age)
                        :func #'chron-age-minus-bone-age
                        :model '(constant 0))))
                (definterval :name "linear-growth-ends"
                 ;; height is constant for the next years.
                 :constraints
                 (list (defconstraint :name "height-constant"
                        :parameters '(height)
                        :model '(constant *))))
                )
        :relations
         '((birth growth-stops (years 18) (years 20))
           ;; Tanner stage intervals span childhood.
           (birth (begin tanner-1) 0 0)
           (consecutive-phase tanner-1 tanner-2)
           (consecutive-phase tanner-2 tanner-3)
           (consecutive-phase tanner-3 tanner-4)
           (consecutive-phase tanner-4 tanner-5)
           (growth-stops (end tanner-5) 0 0)
           ;; From Tanner and Davies Charts
           (birth (begin tanner-2) (years 12.5) (years 14))
           (birth (begin tanner-5) (years 14.5) (years 16))
           ((begin early-childhood) birth 0 0)
           ((begin early-childhood) (end early-childhood) (years 2.05) (years 2.95))
           (consecutive-phase early-childhood pre-puberty)
           (puberty-onset (end pre-puberty) 0 0)
           ((begin linear-growth) birth 0 0)
```

```
  ((end linear-growth) growth-stops 0 0)
  (consecutive-phase linear-growth linear-growth-ends)
  ((begin linear-growth-ends) (end linear-growth-ends) (years 5) (years 5))
  ;; Tanner and Davies charts constrain the growth spurt interval:
  (birth (begin growth-spurt-rise) (years 13) (years 13.5))
  (consecutive-phase growth-spurt-rise growth-spurt-fall)
  ;; By definition peak ht velocity is at transition point between two
  ;; growth spurt intervals:
  (peak-ht-velocity (end  growth-spurt-rise) 0 0)
  ;; Puberty onset is equal to onset of tanner 2:
  (puberty-onset (begin tanner-2) 0 0)
  ;; Peak ht velocity within the growth spurt, from Tanner and Davies charts:
  (birth peak-ht-velocity (years 14) (years 16))
  (growth-stops (end growth-spurt-fall) 0 0)))

(deftt  "boy-early-puberty"
 :landmarks '(birth puberty-onset growth-stops peak-ht-velocity)
 :intervals  (list (definterval :name "early-childhood"
                    :constraints
                    (list  (defconstraint :name "constant build"
                            :func #'bd-infant-normal
                            :parameters '(weight height)
                            :model '(constant *))
                           (defconstraint :name "height weight co-vary"
                            :func #'wt-minus-ht-infant-zscores
                            :parameters '(weight height)
                            :model '(constant *))))
                   (definterval  :name "pre-puberty"
                    :constraints
                    (list  (defconstraint :name "maintain height early centiles"
                            :func #'early-sd
                            :parameters '(height)
                            :model '(constant *))
                           (defconstraint  :name "maintain weight centiles"
                            :func #'child-nchs-sd
                            :parameters '(weight)
                            :model '(constant *))
                           (defconstraint
                            :parameters '(bone-age)
                            :func #'chron-age-minus-bone-age
                            :model '(constant -2))))
                   (definterval :name "linear-growth"
                    )
                   (definterval :name "tanner-1"
                    :constraints
                    (list (defconstraint  :name "tanner 1"
                           :parameters '(pubertal-stage)
```

```
                    :model '(constant 1))))
                (definterval :name "tanner-2"
                 :constraints
                 (list (defconstraint  :name "tanner 2"
                    :parameters '(pubertal-stage)
                    :model '(constant 2))))
                (definterval :name "tanner-3"
                 :constraints
                 (list (defconstraint  :name "tanner 3"
                    :parameters '(pubertal-stage)
                    :model '(constant 3))))
                (definterval :name "tanner-4"
                 :constraints
                 (list (defconstraint  :name "tanner 4"
                    :parameters '(pubertal-stage)
                    :model '(constant 4))))
                (definterval :name "tanner-5"
                 :constraints
                 (list (defconstraint  :name "tanner 5"
                    :parameters '(pubertal-stage)
                    :model '(constant 5))))
                (definterval :name "growth-spurt-rise"
                 :constraints
                 (list (defconstraint :name "height-acceleration"
                    :parameters '(height)
                   :model '(quadratic (D1 +) (D2 +)))
                  (defconstraint
                    :parameters '(bone-age)
                    :func #'chron-age-minus-bone-age
                    :model '(constant -1))))
                (definterval :name "growth-spurt-fall"
                 :constraints
                 (list (defconstraint :name "height-deceleration"
                    :parameters '(height)
                   :model '(quadratic (D1 +) (D2 -)))
                   (defconstraint
                    :parameters '(bone-age)
                    :func #'chron-age-minus-bone-age
                    :model '(constant 0))))
                (definterval :name "linear-growth-ends"
                 ;; height is constant for the next years.
                 :constraints
                 (list (defconstraint :name "height-constant"
                     :parameters '(height)
                     :model '(constant *))))
                )
        :relations
```

```
'((birth growth-stops (years 16) (years 18))
  ;; Tanner stage intervals span childhood.
  (birth (begin tanner-1) 0 0)
  (consecutive-phase tanner-1 tanner-2)
  (consecutive-phase tanner-2 tanner-3)
  (consecutive-phase tanner-3 tanner-4)
  (consecutive-phase tanner-4 tanner-5)
  (growth-stops (end tanner-5) 0 0)
  ;; From Tanner and Davies Charts
  (birth (begin tanner-2) (years 9) (years 10.5))
  (birth (begin tanner-5) (years 11) (years 12.5))
  ((begin early-childhood) birth 0 0)
  ((begin early-childhood) (end early-childhood) (years 2.05) (years 2.95))
  (consecutive-phase early-childhood pre-puberty)
  (puberty-onset (end pre-puberty) 0 0)
  ((begin linear-growth) birth 0 0)
  ((end linear-growth) growth-stops 0 0)
  (consecutive-phase linear-growth linear-growth-ends)
  ((begin linear-growth-ends) (end linear-growth-ends) (years 5) (years 5))
  ;; Tanner and Davies charts constrain the growth spurt interval:
  (birth (begin growth-spurt-rise) (years 8.5) (years 10))
  (consecutive-phase growth-spurt-rise growth-spurt-fall)
  ;; By definition peak ht velocity is at transition point between two
  ;; growth spurt intervals:
  (peak-ht-velocity (end  growth-spurt-rise) 0 0)
  ;; Puberty onset is equal to onset of tanner 2:
  (puberty-onset (begin tanner-2) 0 0)
  ;; Peak ht velocity within the growth spurt, from Tanner and Davies charts:
  (birth peak-ht-velocity (years 10.5) (years 12.5))
 (growth-stops (end growth-spurt-fall) 0 0)))


(make-instance 'monitor-set
 :name "boy-growth"
 :tts (list "boy-average-growth" "boy-constitutional-delay"
      "boy-early-puberty")
 :grain (months 3))
```

A.2 Girl Growth Monitor Set

```
(deftt  "girl-average-growth"
 :landmarks '(birth puberty-onset growth-stops peak-ht-velocity)
 :intervals  (list (definterval :name "early-childhood"
                   :constraints
                   (list  (defconstraint :name "constant build"
                          :func #'bd-infant-normal
```

```
          :parameters '(weight height)
          :model '(constant *))
         (defconstraint :name "height weight co-vary"
          :func #'wt-minus-ht-infant-zscores
          :parameters '(weight height)
          :model '(constant *))))
(definterval  :name "pre-puberty"
 :constraints
 (list  (defconstraint :name "maintain height centiles"
          :func #'avg-sd
          :parameters '(height)
          :model '(constant *))
        (defconstraint  :name "maintain weight centiles"
          :func #'child-nchs-sd
          :parameters '(weight)
          :model '(constant *))))
(definterval :name "linear-growth"
 :constraints
 (list (defconstraint :name "normal bone age"
        :func  #'chron-age-minus-bone-age
        :parameters '(bone-age)
        :model '(constant 0))))
(definterval :name "tanner-1"
 :constraints
 (list (defconstraint  :name "tanner 1"
        :parameters '(pubertal-stage)
        :model '(constant 1))))
(definterval :name "tanner-2"
 :constraints
 (list (defconstraint  :name "tanner 2"
        :parameters '(pubertal-stage)
        :model '(constant 2))))
(definterval :name "tanner-3"
 :constraints
 (list (defconstraint  :name "tanner 3"
        :parameters '(pubertal-stage)
        :model '(constant 3))))
(definterval :name "tanner-4"
 :constraints
 (list (defconstraint  :name "tanner 4"
        :parameters '(pubertal-stage)
        :model '(constant 4))))
(definterval :name "tanner-5"
 :constraints
 (list (defconstraint  :name "tanner 5"
        :parameters '(pubertal-stage)
        :model '(constant 5))))
```

```
                (definterval :name "growth-spurt-rise"
                 :constraints
                 (list (defconstraint :name "height-acceleration"
                        :parameters '(height)
                       :model '(quadratic (D1 +) (D2 +)))))
                (definterval :name "growth-spurt-fall"
                 :constraints
                 (list (defconstraint :name "height-deceleration"
                        :parameters '(height)
                       :model '(quadratic (D1 +) (D2 -)))))
                (definterval :name "linear-growth-ends"
                 ;; height is constant for the next years.
                 :constraints
                 (list (defconstraint :name "height-constant"
                        :parameters '(height)
                        :model '(constant *))))
                )
        :relations
         '((birth growth-stops (years 15) (years 18))
          ;; Tanner stage  intervals span childhood.
          (birth (begin tanner-1) 0 0)
          (consecutive-phase tanner-1 tanner-2)
          (consecutive-phase tanner-2 tanner-3)
          (consecutive-phase tanner-3 tanner-4)
          (consecutive-phase tanner-4 tanner-5)
          (growth-stops (end tanner-5) 0 0)
          ;; From Tanner and Davies Charts
          (birth (begin tanner-2) (years 9.5) (years 12.5))
          (birth (begin tanner-5) (years 11) (years 14.5))
          ((begin early-childhood) birth 0 0)
          ((begin early-childhood) (end early-childhood) (years 2.05) (years 2.95))
          (consecutive-phase early-childhood pre-puberty)
          (puberty-onset (end pre-puberty) 0 0)
          ((begin linear-growth) birth 0 0)
          ((end linear-growth) growth-stops 0 0)
          (consecutive-phase linear-growth linear-growth-ends)
          ((begin linear-growth-ends) (end linear-growth-ends) (years 5) (years 5)).
          ;; Tanner and Davies charts constrain the growth spurt interval:
          (birth (begin growth-spurt-rise) (years 8.5) (years 10))
          (consecutive-phase growth-spurt-rise growth-spurt-fall)
          ;; By definition peak ht velocity is at transition point between two
          ;; growth spurt intervals:
          (peak-ht-velocity (end  growth-spurt-rise) 0 0)
          ;; Puberty onset is equal to onset of tanner 2:
          (puberty-onset (begin tanner-2) 0 0)
          ;; Peak ht velocity within the growth spurt, from Tanner and Davies charts:
          (birth peak-ht-velocity (years 10.5) (years 12.5))
```

```
    (growth-stops (end growth-spurt-fall) 0 0)))

(deftt "girl-constitutional-delay"
 :landmarks '(birth puberty-onset growth-stops peak-ht-velocity)
 :intervals (list (definterval :name "early-childhood"
                   :constraints
                   (list (defconstraint :name "constant build"
                          :func #'bd-infant-normal
                          :parameters '(weight height)
                          :model '(constant *))
                         (defconstraint :name "height weight co-vary"
                          :func #'wt-minus-ht-infant-zscores
                          :parameters '(weight height)
                          :model '(constant *))))
                  (definterval :name "pre-puberty"
                   :constraints
                   (list (defconstraint :name "maintain height delayed centiles"
                          :func #'late-sd
                          :parameters '(height)
                          :model '(constant *))
                         (defconstraint :name "maintain weight centiles"
                          :func #'child-nchs-sd
                          :parameters '(weight)
                          :model '(constant *))
                         (defconstraint
                          :parameters '(bone-age)
                          :func #'chron-age-minus-bone-age
                          :model '(constant 2))))
                  (definterval :name "linear-growth"
                   )
                  (definterval :name "tanner-1"
                   :constraints
                   (list (defconstraint :name "tanner 1"
                          :parameters '(pubertal-stage)
                          :model '(constant 1))))
                  (definterval :name "tanner-2"
                   :constraints
                   (list (defconstraint :name "tanner 2"
                          :parameters '(pubertal-stage)
                          :model '(constant 2))))
                  (definterval :name "tanner-3"
                   :constraints
                   (list (defconstraint :name "tanner 3"
                          :parameters '(pubertal-stage)
                          :model '(constant 3))))
                  (definterval :name "tanner-4"
                   :constraints
```

```
                (list (defconstraint  :name "tanner 4"
                        :parameters '(pubertal-stage)
                        :model '(constant 4))))
               (definterval :name "tanner-5"
                :constraints
                (list (defconstraint  :name "tanner 5"
                        :parameters '(pubertal-stage)
                        :model '(constant 5))))
               (definterval :name "growth-spurt-rise"
                :constraints
                (list (defconstraint :name "height-acceleration"
                        :parameters '(height)
                       :model '(quadratic (D1 +) (D2 +)))
                      (defconstraint
                        :parameters '(bone-age)
                        :func #'chron-age-minus-bone-age
                        :model '(constant 1))))
               (definterval :name "growth-spurt-fall"
                :constraints
                (list (defconstraint :name "height-deceleration"
                        :parameters '(height)
                        :model '(quadratic (D1 +) (D2 -)))
                      (defconstraint
                        :parameters '(bone-age)
                        :func #'chron-age-minus-bone-age
                        :model '(constant 0))))
               (definterval :name "linear-growth-ends"
                ;; height is constant for the next years.
                :constraints
                (list (defconstraint :name "height-constant"
                        :parameters '(height)
                        :model '(constant *))))
               )
        :relations
         '((birth growth-stops (years 17) (years 19))
          ;; Tanner stage  intervals span childhood.
          (birth (begin tanner-1) 0 0)
          (consecutive-phase tanner-1 tanner-2)
          (consecutive-phase tanner-2 tanner-3)
          (consecutive-phase tanner-3 tanner-4)
          (consecutive-phase tanner-4 tanner-5)
          (growth-stops (end tanner-5) 0 0)
          ;; From Tanner and Davies Charts
          (birth (begin tanner-2) (years 12) (years 13.5))
          (birth (begin tanner-5) (years 14) (years 16))
          ((begin early-childhood) birth 0 0)
          ((begin early-childhood) (end early-childhood) (years 2.05) (years 2.95))
```

```
    (consecutive-phase early-childhood pre-puberty)
    (puberty-onset (end pre-puberty) 0 0)
    ((begin linear-growth) birth 0 0)
    ((end linear-growth) growth-stops 0 0)
    (consecutive-phase linear-growth linear-growth-ends)
    ((begin linear-growth-ends) (end linear-growth-ends) (years 5) (years 5))
    ;; Tanner and Davies charts constrain the growth spurt interval:
    (birth (begin growth-spurt-rise) (years 11) (years 12))
    (consecutive-phase growth-spurt-rise growth-spurt-fall)
    ;; By definition peak ht velocity is at transition point between two
    ;; growth spurt intervals:
    (peak-ht-velocity (end  growth-spurt-rise) 0 0)
    ;; Puberty onset is equal to onset of tanner 2:
    (puberty-onset (begin tanner-2) 0 0)
    ;; Peak ht velocity within the growth spurt, from Tanner and Davies charts:
    (birth peak-ht-velocity (years 12.5) (years 14))
    (growth-stops (end growth-spurt-fall) 0 0)))

(deftt  "girl-early-puberty"
  :landmarks '(birth puberty-onset growth-stops peak-ht-velocity)
  :intervals  (list (definterval :name "early-childhood"
                     :constraints
                     (list  (defconstraint :name "constant build"
                              :func #'bd-infant-normal
                              :parameters '(weight height)
                              :model '(constant *))
                            (defconstraint :name "height weight co-vary"
                              :func #'wt-minus-ht-infant-zscores
                              :parameters '(weight height)
                              :model '(constant *))))
                    (definterval  :name "pre-puberty"
                     :constraints
                     (list  (defconstraint :name "maintain height early centiles"
                              :func #'early-sd
                              :parameters '(height)
                              :model '(constant *))
                            (defconstraint  :name "maintain weight centiles"
                              :func #'child-nchs-sd
                              :parameters '(weight)
                              :model '(constant *))
                            (defconstraint
                             :parameters '(bone-age)
                             :func #'chron-age-minus-bone-age
                             :model '(constant -2))))
                    (definterval :name "linear-growth"
                     )
                    (definterval :name "tanner-1"
```

```
  :constraints
  (list (defconstraint  :name "tanner 1"
        :parameters '(pubertal-stage)
        :model '(constant 1))))
 (definterval :name "tanner-2"
  :constraints
  (list (defconstraint  :name "tanner 2"
        :parameters '(pubertal-stage)
        :model '(constant 2))))
 (definterval :name "tanner-3"
  :constraints
  (list (defconstraint  :name "tanner 3"
        :parameters '(pubertal-stage)
        :model '(constant 3))))
 (definterval :name "tanner-4"
  :constraints
  (list (defconstraint  :name "tanner 4"
        :parameters '(pubertal-stage)
        :model '(constant 4))))
 (definterval :name "tanner-5"
  :constraints
  (list (defconstraint  :name "tanner 5"
        :parameters '(pubertal-stage)
        :model '(constant 5))))
 (definterval :name "growth-spurt-rise"
  :constraints
  (list (defconstraint :name "height-acceleration"
        :parameters '(height)
       :model '(quadratic (D1 +) (D2 +)))
      (defconstraint
        :parameters '(bone-age)
        :func #'chron-age-minus-bone-age
        :model '(constant -1))))
 (definterval :name "growth-spurt-fall"
  :constraints
  (list (defconstraint :name "height-deceleration"
        :parameters '(height)
       :model '(quadratic (D1 +) (D2 -)))
      (defconstraint
        :parameters '(bone-age)
        :func #'chron-age-minus-bone-age
        :model '(constant 0))))
 (definterval :name "linear-growth-ends"
  ;; height is constant for the next years.
  :constraints
  (list (defconstraint :name "height-constant"
        :parameters '(height)
```

```
                    :model '(constant *)))))
                )
  :relations
   '((birth growth-stops (years 14) (years 16))
     ;; Tanner stage  intervals span childhood.
     (birth (begin tanner-1) 0 0)
     (consecutive-phase tanner-1 tanner-2)
     (consecutive-phase tanner-2 tanner-3)
     (consecutive-phase tanner-3 tanner-4)
     (consecutive-phase tanner-4 tanner-5)
     (growth-stops (end tanner-5) 0 0)
     ;; From Tanner and Davies Charts
     (birth (begin tanner-2) (years 8) (years 10))
     (birth (begin tanner-5) (years 10) (years 11.5))
     ((begin early-childhood) birth 0 0)
     ((begin early-childhood) (end early-childhood) (years 2.05) (years 2.95))
     (consecutive-phase early-childhood pre-puberty)
     (puberty-onset (end pre-puberty) 0 0)
     ((begin linear-growth) birth 0 0)
     ((end linear-growth) growth-stops 0 0)
     (consecutive-phase linear-growth linear-growth-ends)
     ((begin linear-growth-ends) (end linear-growth-ends) (years 5) (years 5))
     ;; Tanner and Davies charts constrain the growth spurt interval:
     (birth (begin growth-spurt-rise) (years 7) (years 8.5))
     (consecutive-phase growth-spurt-rise growth-spurt-fall)
     ;; By definition peak ht velocity is at transition point between two
     ;; growth spurt intervals:
     (peak-ht-velocity (end  growth-spurt-rise) 0 0)
     ;; Puberty onset is equal to onset of tanner 2:
     (puberty-onset (begin tanner-2) 0 0)
     ;; Peak ht velocity within the growth spurt, from Tanner and Davies charts:
     (birth peak-ht-velocity (years 8.5) (years 10.5))
     (growth-stops (end growth-spurt-fall) 0 0)))

(make-instance 'monitor-set
  :name "girl-growth"
  :tts (list "girl-average-growth" "girl-constitutional-delay"
        "girl-early-puberty")
  :grain (months 3))
```

Appendix B Intensive Care Unit Monitor Sets

Below is the computer code that TrenDx interprets to create monitor sets for intensive care unit trends. There are two monitor sets, one for initial blood pressure classification and one for oxygen handbagging. Monitor set definitions follow the definitions of their member trend templates. These monitor sets are also described in section 3.4 and section 5.2.

The code is written in a trend template definition language, implemented on top of the Common Lisp Object System (CLOS) {Steele, 1990 #242}. The higher level trend template language was developed for compact, readable encoding of trend templates. Commands beginning with "`def`" define a trend template object. For example, `deftt` defines a trend template, and `definterval` define an interval within that trend template.

Value constraint definitions include some subset of the following: name, function to apply to data within an interval, parameters (i.e. types of measurements) that the function is applied to, and a model: a low order polynomial with parameters estimated either qualitatively or quantitatively. For example, a constraint `'(constant *)` means constant at any level, while `'(constant 100)` denotes constant at value 100. A constraint that includes `D1` or `D2` constrains the first or second derivative of a polynomial model.

B.1 Blood Pressure Initial Classification Monitor Set

```
(deftt "rising-mean-blood-pressure"
        :landmarks '(now )
        :intervals  (list (definterval :name "stable-pressure"
                          :constraints
                          (list  (defconstraint  :model '(constant *)
                                 :parameters '(mean-arterial-pressure))))
                        (definterval  :name "increasing-pressure"
                          :constraints
                          (list  (defconstraint :model '(linear (D1 +))
                                 :parameters '(mean-arterial-pressure)))))
        :relations
```

```
'(((begin increasing-pressure) (end increasing-pressure) (minutes 1) (minutes 5))
   ((begin stable-pressure) now (minutes 5) (minutes 5))
   (now (end increasing-pressure) 0 0)
   (consecutive-phase stable-pressure increasing-pressure)))

(deftt  "falling-mean-blood-pressure"
 :landmarks '(now)
 :intervals  (list (definterval :name "stable-pressure"
                  :constraints
                  (list  (defconstraint :model '(constant *)
                         :parameters '(mean-arterial-pressure))))
                 (definterval  :name "decreasing-pressure"
                  :constraints
                  (list  (defconstraint :model '(linear (D1 -))
                         :parameters '(mean-arterial-pressure)))))
 :relations '(((begin decreasing-pressure) (end decreasing-pressure)
                (minutes 1) (minutes 5))
           ((begin stable-pressure) now (minutes 5) (minutes 5))
           (now (end decreasing-pressure) 0 0)
           (consecutive-phase stable-pressure decreasing-pressure)))

(deftt  "steady-mean-blood-pressure"
 :landmarks '(now)
 :intervals  (list (definterval :name "stable-pressure"
                  :constraints
                  (list  (defconstraint :model '(constant *)
                         :parameters '(mean-arterial-pressure)))))
 :relations
 '(((begin stable-pressure) now (minutes 5) (minutes 5))
   (now (end stable-pressure) 0 0)))

(make-instance 'monitor-set
 :name "mean-blood-pressure-initial"
 :tts  (list "rising-mean-blood-pressure" "falling-mean-blood-pressure"
          "steady-mean-blood-pressure"))
```

B.2 Handbagging Monitor Set

```
(deftt "adequate-handbagging"
 :landmarks '(handbagging-on handbagging-off)
 :intervals (list (definterval :name "handbag-control"
                  :constraints
                  (list (defconstraint :model '(constant 100)
                        :parameters '(fio2))
                       (defconstraint  :model '(constant 1)
                        :parameters  '(ventilation-mode))))
```

```
                (definterval :name "pressure-control"
                 :constraints
                 (list (defconstraint :model '(constant *)
                        :parameters '(fio2))
                       (defconstraint  :model '(constant 0)
                        :parameters  '(ventilation-mode))))
                (definterval :name "saturating-hemoglobin"
                 :constraints
                 (list (defconstraint  :model '(linear (D1 +))
                        :parameters '(arterial-o2-sat))))
                (definterval :name "saturated-hemoglobin"
                 :constraints
                 (list (defconstraint :model '(constant 100)
                        :parameters '(arterial-o2-sat))))
                (definterval :name  "steady-hemodynamics"
                 :constraints
                 (list (defconstraint :model '(constant *)
                        :parameters '(MEAN-ARTERIAL-PRESSURE))
                       (defconstraint :model '(constant *)
                        :parameters '(ecg-heart-rate)) )))
  :relations '((handbagging-on (begin handbag-control) 0 0)
               (handbagging-off (end pressure-control) 0 0)
               (handbagging-on handbagging-off (minutes 8) (minutes 30))
               ((begin pressure-control) (end pressure-control) *epsilon* *epsilon*)
               (consecutive-phase handbag-control pressure-control)
               (handbagging-on (begin saturating-hemoglobin) 0 0)
               (consecutive-phase saturating-hemoglobin saturated-hemoglobin)
          (handbagging-on (begin saturated-hemoglobin) 0 (minutes 5))
          (handbagging-off (end saturated-hemoglobin) 0 0)
               (handbagging-on (begin steady-hemodynamics) 0 0)
               (handbagging-off (end steady-hemodynamics) 0 0)))

(deftt "compromised-venous-return"
  :landmarks '(handbagging-on handbagging-off)
  :intervals (list (definterval :name "handbag-control"
                    :constraints
                    (list (defconstraint :model '(constant 100)
                           :parameters '(fio2))
                          (defconstraint  :model '(constant 1)
                           :parameters  '(ventilation-mode))))
                   (definterval :name "pressure-control"
                    :constraints
                    (list (defconstraint :model '(constant *)
                           :parameters '(fio2))
                          (defconstraint  :model '(constant 0)
                           :parameters  '(ventilation-mode))))
                   (definterval :name "saturating-hemoglobin"
```

```
                :constraints
                (list (defconstraint  :model '(linear (D1 +))
                      :parameters '(arterial-o2-sat))))
               (definterval :name "saturated-hemoglobin"
                :constraints
                (list (defconstraint :model '(constant 100)
                      :parameters '(arterial-o2-sat))))
               (definterval :name  "steady-blood-pressure"
                :constraints
                (list (defconstraint :model '(constant *)
                      :parameters '(MEAN-ARTERIAL-PRESSURE))))
               (definterval  :name "decreasing-blood-pressure"
                 :constraints
               (list  (defconstraint :model '(linear (D1 -))
                      :parameters '(MEAN-ARTERIAL-PRESSURE))))
               (definterval :name "steady-heart-rate"
                :constraints
                (list (defconstraint :model '(constant *)
                      :parameters '(ecg-heart-rate) )))
               (definterval  :name "increasing-heart-rate"
                 :constraints
                 (list  (defconstraint :model '(linear (D1 +))
                      :parameters '(ecg-heart-rate)))))
 :relations '((handbagging-on (begin handbag-control) 0 0)
          (handbagging-off (end pressure-control) 0 0)
          (handbagging-on handbagging-off (minutes 8) (minutes 30))
          ((begin pressure-control) (end pressure-control) *epsilon* *epsilon*)
          (consecutive-phase handbag-control pressure-control)
          (handbagging-on (begin saturating-hemoglobin) 0 0)
          (consecutive-phase saturating-hemoglobinsaturated-hemoglobin)
          (handbagging-on (begin saturated-hemoglobin) 0 (minutes 5))
          (handbagging-off (end saturated-hemoglobin) 0 0)
          (handbagging-on (begin steady-blood-pressure) 0 0)
          (consecutive-phase steady-blood-pressure decreasing-blood-pressure)
(handbagging-on (begin decreasing-blood-pressure) (minutes 3) (minutes 10))
          (handbagging-off (end decreasing-blood-pressure) 0 0)
          (handbagging-on (begin steady-heart-rate) 0 0)
          (handbagging-off (end increasing-heart-rate) 0 0)
          (consecutive-phase steady-heart-rate increasing-heart-rate)
     ((begin decreasing-blood-pressure) (begin increasing-heart-rate)
               (minutes 0) (minutes 5))

(make-instance 'monitor-set
 :name "response-to-handbagging"
 :tts (list "adequate-handbagging" "compromised-venous-return")
 :grain (minutes 1))
```

Appendix C Discriminatory Power of Value Constraints

Several experiments using data generated from stochastic models determined the discriminatory power of the low-order polynomial models used in trend template value constraints. This appendix fully details the experimental methods and results.

C.1 Experimental Method

Sets of 60 (value, time) pairs were generated at random using data drawn from constant, linear, and quadratic models:

- Constant: (value, time) = $(58 + \varepsilon, n)$.
- Linear: (value, time) = $(6n + 40 + \varepsilon, n)$.
- Quadratic: (value, time) = $(n^2 + 40 + \varepsilon, n)$.

The numbers n were randomly drawn from a uniform distribution from [0 to 6] in increments of .01. Included in each model is an additive white noise ε with mean 0 and a fixed standard deviation (either 1 or 3) in each experiment. The ε's were independent and identically distributed. Data values ranged from 40 to 76. The diagram below illustrates the three distributions from which data were generated.

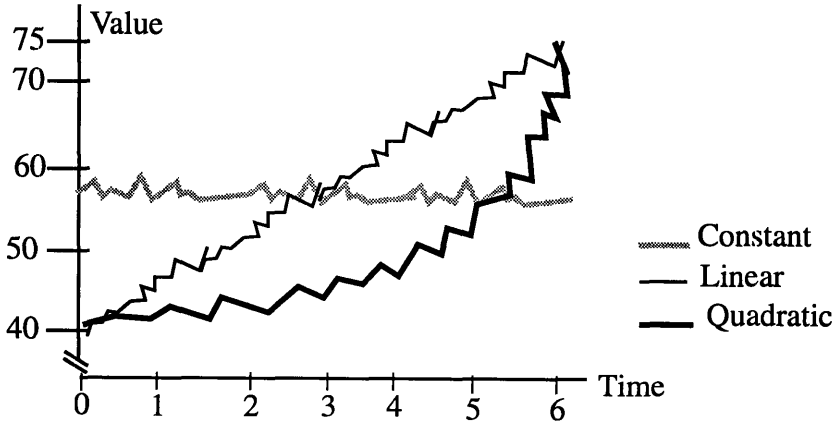


Figure 50 Distributions from which random data were generated.

Data generated from these stochastic models were matched to each of three value constraint models with qualitative derivative constraints:

- constant model (no estimates).
- linear model, positive derivative.
- quadratic model, positive first and second derivatives.

Each experiment included tests on eight different population ratios: all data from one stochastic model (three ratios), equal numbers of data from two of the stochastic models (three ratios), equal numbers from all three models (one ratio), and a set of randomly scattered data over the value range for control (one ratio). Each population ratio called for sixty (value, time) pairs. The different population ratios may be seen on the left side of Table 3.

For each population ratio within each experiment, fifty populations of sixty data points each were drawn according to that ratio. TrenDx computed the matching score, based on mean absolute percentage error (MAPE), between the generated data and each of the three above value constraints. As described in 4.4.4, at times the MAPE is multiplied by a penalty (of 1.25 here) in case of failed qualitative constraints. I calculated means and standard deviations of the matching scores over each set of fifty populations. T-tests compared the mean scores of each pair of value constraint models using the test for unknown and possibly unequal variances [Sachs 1984, page 271]; the number of degrees of freedom in the T-tests was near 50 for all experiments. An analysis of variance (ANOVA) may also be conducted to analyze the results among the three value constrants. However, the pairwise T-tests are instructive for helping the knowledge engineer decide the significance of modeling with different order polynomial models.

Three experiments tested value constraint performance in different conditions:

1. (Value, times) drawn from [0,6] x [40,76] and white noise N(0,1) added.
2. (Value, times) drawn from [0,6] x [40,76] and white noise N(0,3) added.
3. Value, times) drawn from [0,3] x [50,59] and white noise N(0,1) added.

Comparing results between experiments 1 and 2 should indicate effects of increasing noise in a sensor. Comparing results between experiments 1 and 3 should indicate effects of matching to a signal with a smaller change during an interval.

C.2 Results of Initial Experiment

The table below shows the mix of stochastic distributions in the test data for each experiment (at left), as well as the results of the initial experiment (at right). Overall, the errors range from near 1% mean absolute error to just over 20% mean absolute error. Not surprisingly, match scores are highest to the random data. Note that if values were randomly generated from a range of [10, 46] instead of [40, 76], these percentage values would be higher (since the denominators of the MAPE

would be smaller), yet the comparative results and significance values would be quite similar.

SAMPLE (Data per model)			Match Score (% error) for 50 trials					
			Averages			Difference of mean T-scores		
Const	Linear	Quad	E_{con}	E_{lin}	E_{quad}	$T_{con, lin}$	$T_{con, quad}$	$T_{lin, quad}$
60	0	0	1.36	1.57	1.73	0.77	0.73	0.84
0	60	0	16.20	1.41	1.67	9.65**	8.85**	0.77
0	0	60	17.58	4.80	1.60	5.66**	9.81**	2.68**
30	30	0	9.12	8.05	9.25	2.26*	2.11*	1.86*
30	0	30	12.80	10.78	11.73	2.75**	2.63**	2.22*
0	30	30	17.73	6.29	5.87	4.99**	5.16**	1.84*
20	20	20	13.59	9.25	10.20	3.15**	3.00**	2.04*
Random 60			16.19	18.57	20.91	2.65**	2.49**	2.86**

TABLE 3. Matching low-order polynomial models to data valued within [40, 76]. White noise is Normal(0,1). Difference of mean T-scores: * denotes p<.05; ** denotes p<.01.

The results illustrate characteristic features of linear regression. When the sample consisted entirely of points from one polynomial model, the value constraint of the same polynomial matched decidedly better than value constraints of *lower order* polynomials, but not significantly better than the value constraints of *higher* order polynomials. For example, row two of the table shows that when the sample is entirely from the linear model, the linear value constraint has a significantly lower percentage error than the constant model (1.41% to 16.20%; $p < .01$). Yet the difference in scores between linear and quadratic models (1.41% to 1.67%) is not significant.

When the sample is drawn from two distributions, the value constraint representing the higher order polynomial tended to score better, usually at the $p < .05$ level. For example, row six of the table shows that an even mix of linear and quadratic data matches better to the quadratic value constraint than to the linear one(5.87% to 6.29%; $p <. 05$).

C.3 Result of Increasing Sensor Noise

The results of the same experiment for a noisier sensor, of white noise mean 0 and standard deviation 3, appear in Table 4 below. Generally the mean match scores are higher than those with a smaller error. However, most of the statistically significant differences of the first experiment remain. This is essentially because even with larger noise added to the different distributions of Figure 50, the resulting sample is still sufficiently separable. .

SAMPLE (Data per model)			Match Score (% error) for 50 trials					
			Averages			Difference of mean T-scores		
Const	Linear	Quad	E_{con}	E_{lin}	E_{quad}	$T_{con, lin}$	$T_{con, quad}$	$T_{lin, quad}$
60	0	0	4.20	4.69	5.27	1.37	1.29	1.44
0	60	0	16.56	4.27	4.83	5.66**	5.32**	1.37
0	0	60	17.97	6.60	6.40	4.93**	4.92**	1.81*
30	30	0	10.29	8.71	9.75	2.46**	2.32*	1.96*
30	0	30	13.15	11.20	12.17	2.77**	2.66**	2.26*
0	30	30	17.90	7.36	7.10	4.66**	4.74**	1.95*
20	20	20	14.45	9.94	10.61	3.23**	3.13**	2.15*
Random 60			16.29	18.57	20.91	2.65**	2.50**	2.86**

TABLE 4. Matching similar data as above, with white noise as Normal(0, 3), to the three value constraint models. Difference of mean T-scores: * denotes p<.05; ** denotes p<.01.

C.4 Result of Decreasing Net Change Over Interval

In the third experiment, I decreased the spread in the generated data, by using three stochastic models qualitatively similar to those of Figure 50 yet yielding (value, time) pairs from [50 to 59] x [0,3]. The white noise generated with the data was reduced to again be Normal(0,1). The results appear below and are best compared to the results in Table 3.

SAMPLE (Data per model)			Match Score (% error) for 50 trials					
			Averages			Difference of mean T-scores		
Const	Linear	Quad	E_{con}	E_{lin}	E_{quad}	$T_{con, lin}$	$T_{con, quad}$	$T_{lin, quad}$
60	0	0	1.50	1.66	1.89	0.82	0.77	0.85
0	60	0	4.33	1.46	1.63	2.53**	2.39**	0.81
0	0	60	4.46	1.80	1.90	2.35*	2.28*	0.92
30	30	0	2.95	2.56	2.89	1.30	1.22	1.06
30	0	30	3.32	2.90	3.21	1.38	1.31	1.14
0	30	30	4.66	2.04	1.99	2.31*	2.33*	1.02
20	20	20	3.62	2.66	2.86	1.57	1.51	1.11
Random 60			4.16	4.62	5.21	1.37	1.28	1.42

TABLE 5. Matching similar data as in Table 3; data valued within [50, 59]; with white noise as Normal(0, 1). Difference of mean T-scores: * denotes p<.05; ** denotes p<.01.

Many average error scores are somewhat lower than corresponding entries in Table 3, for two artifactual reasons. First, due to the lower spread of values, the goodness of fit to the constant value constraint is virtually certain to improve. Second, in this third experiment more of the randomly generated linear and quadratic points were in a higher value range (over 50) than in the first two experiments (many were between 40 and 50). A larger datum value generally yields a smaller MAPE.

More importantly, many of the statistically significant differences in mean errors from experiment 1 are no longer significant. This is largely due to the added white noise blurring the distinction between the stochastic models. The linear model was $(50 + 3n + \varepsilon)$ for n in [0,3] and the quadratic model was $(50 + n^2 + \varepsilon)$ for n in [0,3]. For n from 0 to 2 the difference in values is rarely more than 2; when white noise of standard deviation 1 is added, values from the different stochastic models may frequently overlap, Thus the data are not nearly as separable into the original models as in experiments 1 or 2.

C.5 Discussion

When choosing a low-order polynomial model for a value constraint that models some signal $\{Y_t\}$, one needs to consider at least three attributes of the signal: the expected spread of values, the expected noise, and the likely distinct qualitative shapes. Generally one may model in competing trend templates different value constraints with each of the expected qualitative shapes. However, if the noise in $\{Y_t\}$ is sufficient to blur the distinction between those shapes, as in experiment 3, then modeling with the highest order polynomials (i.e. quadratic) may yield little discrimination. Whether or not this blur occurs depends largely on the expected noise distribution and the spread of values of $\{Y_t\}$.

References

Allen, J. F. (1983). "Maintaining Knowledge about Temporal Intervals." *Communications of the ACM*, **26**(11): 832-843.

Allen, J. F. (1984). "Towards a General Theory of Action and Time." *Artificial Intelligence*, **23**: 123-154.

Allen, J. F. and J. A. Koomen. (1983). "Planning Using a Temporal World Model." International Joint Conference on Artificial Intelligence (IJCAI-83), 741-747.

Avent, R. K. and J. D. Charlton. (1990). "A Critical Review of Trend-Detection Methodologies for Biomedical Monitoring Systems." *Critical Reviews in Biomedical Engineering*, **17**(6): 621-659.

Bankowitz, R. A., J. R. Lave and M. A. McNeil. (1992). "A Method for Assessing the Impact of a Computer-Based Decision Support System on Health Care Outcomes." *Methods of Information in Medicine*, **31**: 3-11.

Beinlich, I. A., H. J. Suermondt, R. M. Chavez and G. F. Cooper. (1989). *The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks.* Knowledge Systems Laboratory, Stanford University Report, KSL-88-84.

Berk, J. L. and J. E. Sampliner. (1990). *Handbook of Critical Care.*

Berzuini, C., R. Bellazzi, S. Quaglini and D. J. Spiegelhalter. (1992). "Bayesian Networks for Patient Monitoring." *Artificial Intelligence in Medicine*, **4**: 243-260.

Betaque, N. E. and G. A. Gorry (1971). "Automating Judgmental Decision Making for a Serious Medical Problem." *Management Science,* **17** (8): B-421 to B-434.

Blum, R. L. (1984). "Discovery, Confirmation, and Incorporation of Causal Relationships from a Large Time-Oriented Clinical Data Base: The RX Project." In *Readings in Medical Artificial Intelligence*. Reading, MA, Addison-Wesley.

Box, G. E. P. and G. M. Jenkins. (1970). *Time Series Analysis: Forecasting and Control.* Series in Time Series Analysis. Holden-Day.

Cheung, J. T.-Y. and G. Stephanopolous. (1990a). "Representation of Process Trends - Part I. A Formal Representation Framework." *Computers in Chemical Engineering*, **14**(4/5): 495-510.

Cheung, J. T.-Y. and G. Stephanopolous. (1990b). "Representation of Process Trends - Part II. The Problem of Scale and Qualitative Scaling." *Computers in Chemical Engineering*, **14**(4/5): 511-539.

Chui, C. K. and G. Chen. (1991). *Kalman Filtering with Real-Time Applications*. Springer Series in Information Sciences. New York, Springer-Verlag.

Coiera, E. (1990). "Monitoring Diseases with Empirical and Model-Generated Histories." *Artificial Intelligence in Medicine*, **2**: 135-147.

Coiera, E. W. (1989). *Reasoning with Qualitative Disease Histories for Diagnostic Patient Monitoring*. Doctoral thesis in computer science, University of New South Wales.

Console, L., A. J. Rivolin and P. Toraso. (1991). "Fuzzy Temporal Reasoning on Causal Models." *International Journal of Intelligent Systems*, **6**: 107-133.

Cousins, S. B. and M. G. Kahn (1991). "The Visual Display of Temporal Information." *Artificial Intelligence in Medicine,* **3:** 341-357.

Cronk, C., A. C. Crocker, S. M. Pueschel, A. M. Shea, E. Zackai, G. Pickens and R. Reed. (1988). "Growth Charts for Children With Down Syndrome: 1 Month to 18 Years of Age." *Pediatrics*, **81**(1): 102-110.

Das, A. K., S. W. Tu, G. P. Purcell and M. A. Musen. (1994). "An Extended SQL for Temporal Data Management in Clinical Decision-Support Programs." Symposium on Computer Applications in Medical Care, Washington, D.C., 128-132.

Dean, T. and K. Kanazawa. (1988). "Probabilistic Temporal Reasoning." National Conference on Artificial Intelligence (AAAI-88), St. Paul, MN, 524-528.

Dean, T. and K. Kanazawa. (1989). "A Model for Reasoning About Persistence and Causation." *Computational Intelligence*, **5**(142-150): 142-150.

Dean, T. L. and M. Boddy. (1987). "Incremental Causal Reasoning." National Conference on Artificial Intelligence (AAAI-87), Seattle, WA, 196-201.

Dean, T. L. and D. V. McDermott. (1987). "Temporal Data Base Management." *Artificial Intelligence*, **32**: 1-55.

Doyle, R., S. Sellers and D. Atkinson. (1989). "A Focused, Context-Sensitive Approach to Monitoring." *I*nternational Joint Conference on Artificial Intelligence (IJCAI-89), Detroit, MI, 1231-1237.

Draper, N. R. and H. Smith. (1981). *Applied Regression Analysis*. New York, John Wiley & Sons.

Dvorak, D. and B. Kuipers. (1989). "Model-Based Monitoring of Dynamic Systems." International Joint Conference on Artificial Intelligence (IJCAI-89), Detroit, MI, 1238-1243.

Enderton, H. B. (1972). *A Mathematical Introduction to Logic.* New York, Academic Press, Inc.

Evans, R. S. (1991). "The HELP System: A Review of Clinical Applications in Infectious Diseases and Antibiotics Use." *M. D. Computing*, **8**(5): 282-288,315.

Fackler, J. and Kohane, I. (1994) "Hypothesis-Driven Data Visualization: SmartDisplay," submitted to Symposium on Computer Applications in Medical Care, Washington, D.C..

Fagan, L. M., J. C. Kunz, E. A. Feigenbaum and J. J. Osborn. (1984). "Extensions to the Rule-Based Formalism for a Monitoring Task." In *Rule-Based Expert Systems.* Addison-Wesley.

Feldman, M. J. and G. O. Barnett. (1990). "An Approach to Evaluating the Accuracy of DXplain." *S*ymposium for Computer Applications in Medical Care, Washington, D.C., 38-43.

Fischer, U., W. Schenk, E. Salzseider, G. Albrecht, P. Abel and E. J. Freyse. (1987). "Does Physiological Blood Glucose Control Require an Adaptive Control Strategy?" *IEEE Transactions on Biomedical Engineering*, **BME-34**: 575-582.

Forbus, K. D. (1985). "Qualitative Process Theory." In *Qualitative Reasoning About Physical Systems.* Cambridge, MA, MIT Press.

Forsythe, D. E. and B. G. Buchanan. (1992). "Broadening Our Approach to Evaluating Medical Information Systems." Symposium for Computer Applications in Medical Care, Washington, D.C., 8-12.

Friedman, W. J. (1990). *About Time.* Cambridge, MA, MIT Press.

Garn, S. M., W. R. Leonard and V. M. Hawthorne. (1986). "Three Limitations of the Body Mass Index." *American Journal of Clinical Nutrition*, **44**: 996-997.

Greulich, W. W. and S. I. Pyle. (1959). *Radiographic Atlas of Skeletal Development of the Hand and Wrist.* Stanford, CA, Stanford University Press.

Guyton, A. C. (1991). *Textbook of Medical Physiology, eighth edition.* Philadelphia, PA, W.B. Saunders.

Haimowitz, I. J. (1988). *Using NIKL in a Large Medical Knowledge Base.* M.I.T. Laboratory for Computer Science technical memo, TM-348.

Haimowitz, I. J. (1991). "Modeling All Dialogue System Participants to Generate Empathetic Responses," *Computer Methods and Programs in Biomedicine,* **35:** 321-330.

Haimowitz, I. J. and I. S. Kohane. (1991). "Influences on the Performance of Hospital Clinical Event Monitoring." Symposium for Computer Applications in Medical Care, Washington, D.C., 614-618.

Haimowitz, I. J. and I. S. Kohane. (1993a). "Automated Trend Detection with Multiple Temporal Hypotheses." International Joint Conference on Artificial Intelligence (IJCAI-93), Chambery, France, 146-151.

Haimowitz, I. J. and I. S. Kohane. (1993b). "An Epistemology for Clinically Significant Trends." National Conference on Artificial Intelligence (AAAI-93), Washington, DC, 176-181.

Hamil, P. V. V., T. A. Drizd, C. L. Johnson, R. B. Reed, A. F. Roche and W. M. Moore. (1979). "Physical Growth: National Center for Health Statistics Percentiles." *The American Journal of Clinical Nutrition*, **32**: 607-629.

Hayes, P. J. (1985). "The Second Naive Physics Manifesto." In *Readings in Knowledge Representation.* Los Altos, CA, Morgan Kaufman.

Higgins, M. C. (1987). "NAIVE: A method for Representing Uncertainty and Temporal Relationships in an Automated Reasoner." Uncertainty in Artificial Intelligence Workshop, Seattle, WA, 140-147.

Hutchinson, J. H. (1994) *A Radial Basis Function Approach to Financial Time Series Analysis.* M.I.T. Artificial Intelligence Laboratory technical report 1457.

Jang, J.-S. (1993). "ANFIS: Adaptive-Network-Based Fuzzy Inference System." *IEEE Transactions on System, Man, and Cybernetics*, **23**(3): 665-685.

Kahn, M. G., C. A. Abrams, S. B. Cousins, J. C. Beard and M. E. Frisse. (1990). "Automated Interpretation of Diabetes Patient Data: Detecting Temporal Changes in Insulin Therapy." Symposium for Computer Applications in Medical Care, Washington, D.C., 569-573.

Kahn, M. G., L. M. Fagan and L. B. Sheiner. (1991). "Combining Physiologic Models and Symbolic Methods to Interpret Time-Varying Patient Data." *Methods of Information in Medicine*, **30**: 167-178.

Kanazawa, K. (1991). "A Logic and Time Nets for Probabilistic Inference." National Conference on Artificial Intelligence (AAAI-91), Anaheim, CA, 360-365.

Kaplan, S. A. (1990). *Clinical Pediatric Endocrinology.* Philadelphia, PA, W.B. Saunders Company.

Kass, R. and T. Finin. (1988). "The Need for User Models in Generating Expert System Explanations." *International Journal of Expert Systems*, **1**(4): 345-375.

Kassirer, J. P. (1989). "Diagnostic Reasoning." *Annals of Internal Medicine*, **110**(11): 893-900.

Keravnou, E. T. and J. Washbrook. (1990). "A Temporal Reasoning Framework Used in the Diagnosis of Skeletal Dysplasias." *Artificial Intelligence in Medicine*, **2**: 239-265.

Kleinrock, L. (1976). *Queueing Systems, Volumes I and II.* New York, John Wiley and Sons.

Kohane, I. S. (1987). *Temporal Reasoning in Medical Expert Systems.* MIT Laboratory for Computer Science Technical Report, TR-389.

Kohane, I. S. (1992). "Maintaining Alternate Interpretations of Data from Multiple Sources in a Clinical Event Monitoring System." MEDINFO-92, Geneva, Switzerland, 483-489.

Kohane, I. S. and I. J. Haimowitz. (1993). "Hypothesis-Driven Data Abstraction with Trend Templates." Symposium for Computer Applications in Medical Care, Washington, D.C., 444-448.

Konstantinov, K. B. and T. Yoshida. (1992). "Real-Time Qualitative Analysis of the temporal Shapes of (Bio)process Variables." *AIChE Journal*, **38**(11): 1703-1715.

Kuipers, B. (1983). "Commonsense Reasoning About Causality: Deriving Behavior from Structure." *Artificial Intelligence*, **2**: 201-203.

Kuipers, B. and D. Berleant. (1988). "Using Incomplete Quantitative Knowledge in Qualitative Reasoning." National Conference on Artificial Intelligence (AAAI-88), St. Paul. MN, 324-329.

Kukich, K. (1983). "Knowledge-Based Report Generation: A Technique for Automatically Generating Natural Language

Reports from Databases." Sixth Annual International ACM SIGIR Conference, Bethesda, MD, 246-250.

Kumar, P. R. and P. Varaiya. (1986). *Stochastic Systems: Estimation, Identification, and Adaptive Control.* Information and System Sciences Series. Englewood Cliffs, NJ, Prentice-Hall.

Kume, H. (1987). *Statistical Methods for Quality Improvement.* Japan, Association for Overseas Technical Scholarship.

Kvanli, A. H., C. S. Guynes and R. J. Pavur. (1989). *Introduction to Business Statistics.* New York, West Publishing Company.

Laird, J. E. and P. S. Rosenbloom. (1992). "In Pursuit of Mind: The Research of Allen Newell." *AI Magazine,* **13:** 18-45.

Larizza, C., A. Moglia and M. Stefanelli. (1992). "M-HTP: A System for Monitoring Heart Transplant Patients." *Artificial Intelligence in Medicine,* **4**(2): 111-126.

Leong, T.-Y. (1993). "Dynamic Decision Modeling in Medicine: A Critique of Existing Formalisms," Symposium for Computer Applications in Medical Care, Washington, D.C., 478-484.

Lindgren, Bernard W. (1976). *Statistical Theory.* New York, Macmillan.

Ljung, L. (1987). *System Identification - Theory for the User.* Englewood Cliffs, N.J., Prentice Hall.

Lundbye-Christensen, S., P. Winkel and M. Christensen. (1991). *Serial Measurements Applied in Clinical Practice.* University of Aalborg Institute for Electronic Systems Report, R 91-31.

Lyon, A. J., M. A. Preece and D. B. Grant. (1985). "Growth Curve for Girls with Turner Syndrome." *Archives of Disease in Childhood*, **60**: 932-935.

McCallie, D. P. J., D. M. Margulies, I. S. Kohane, R. Stalhut and B. Bergeron. (1990). "The Children's Hospital Workstation." Symposium for Computer Applications in Medical Care, Washington, D.C., 755-759.

McKeown, K. (1985). "Discourse Strategies for Generating Natural Language Text." *Artificial Intelligence*, **27**(1): 1-42.

McPherson, G. (1990). *Statistics in Scientific Investigation.* New York, Springer-Verlag.

Miller, P. L. (1986). *Expert Critiquing Systems: Practice-Based Medical Consultation by Computer.* New York, Springer-Verlag.

Nakhimovsky, A. (1988). "Aspect, Aspectual Class, and the Temporal Structure of Narrative." *Computational Linguistics*, **14**(2): 29-43.

Nelson, K. S. and G. D. Hadden. (1992). "Trend Recognition and Failure Prediction of the Attitude Determination and Control System of the Space Station Freedom." 15th Annual AAS Guidance and Control Conference, Keystone, CO, 1-13.

Newell, A. (1981). "The Knowledge Level." *AI Magazine*, 1-33.

Nichols, D. G., J. J. McCloskey and M. C. Rogers. (1992). "Adult Respiratory Distress Syndrome." In *Textbook of Pediatric Intensive Care*. Baltimore, MD, Williams and Wilkins.

Ostrom, C. W. J. (1990). *Time Series Analysis: Regression Techniques*. Quantitative Applications in the Social Sciences. Beverly Hills, CA, Sage Publications.

Parsons, T. (1986). *Voice and Speech Processing*. McGraw-Hill.

Passonneau, R. J. (1988). "A Computational Model of the Semantics of Tense and Aspect." *Computational Linguistics*, **14**(2): 44-60.

Patil, R. S. (1981). *Causal Representation of Patient Illness for Electrolyte and Acid-Base Diagnosis*. M.I.T. Laboratory for Computer Science technical report, TR-267.

Pauker, S. G., G. A. Gorry, J. P. Kassirer and W. B. Schwartz. (1976). "Towards the Simulation of Clinical Cognition: Taking a Present Illness by Computer." *The American Journal of Medicine*, **60**: 981-996.

Percival, D. B. and A. T. Walden. (1993). *Spectral Analysis for Physical Applications*. Cambridge, England, Cambridge University Press.

Pindyck, R. S. (1981). *Econometric Models and Economic Forecasts*. New York, McGraw-Hill.

Podolsky, D. K. and K. J. Isselbacher. (1991). "Diagnostic Tests in Liver Disease." In *Harrison's Principles of Internal Medicine, Twelfth Edition*. McGraw Hill.

Priestly, M. B. (1981). *Spectral Analysis and Time Series*. New York, Academic Press.

Rabiner, L. R. and B. H. Juang. (1986). "An Introduction to Hidden Markov Models." *I.E.E.E. A.S.S.P. Magazine* **3**(1): 4-16.

Rind, D. M., C. Safran, R. S. Phillips, W. V. Slack, D. R. Calkins, T. L. Delbanco, and H. L. Bleich (1991). "The Effect of Computer-Based Reminders on the Management of Hospitalized Patients with Worsening Renal Function," Symposium for Computer Applications in Medical Care, Washington, D.C., 28-32.

Rummelhart, D. E., G. E. Hinton and R. J. Willimas. (1986). "Learning Internal Representations by Error Propagation." In *Parallel Distributed Processing*. Cambridge, MA, MIT Press.

Russ, T. (1985). *Temporal Control Structure Reference Manual.* MIT Laboratory for Computer Science technical memo, TM-331.

Russ, T. A. (1993). *Reasoning with Time Dependent Data.* MIT Laboratory for Computer Science technical report, TR-545.

Rutledge, G. W., S. K. Andersen, J. X. Polaschek and L. M. Fagan. (1990). "A Belief Network Model for Interpretation of ICU Data." Symposium for Computer Applications in Medical Care, Washington, D.C., 785-789.

Ryan, T. P. (1989). *Statistical Methods for Quality Improvement.* Wiley Series in Probability and Mathematical Statistics. New York, Wiley.

Sachs, L. (1984). *Applied Statistics.* New York, Springer-Verlag.

Schalkoff, R. (1992). *Pattern Recognition: Statistical, Structural and Neural Approaches.* New York, John Wiley and Sons.

Sculley, R. E. (1986). "Case records of the Massachusetts General Hospital." *New England Journal of Medicine*, **314** (January 2): 39-49.

Shahar, Y., S. Tu and M. Musen. (1992). "Knowledge Acquisition for Temporal Abstraction Mechanisms." *Knowledge Acquisition*, **1**(4): 217-236.

Shoham, Y. (1988). *Reasoning About Change.* Cambridge, MA, MIT Press.

Shrobe, Howard E. (1988). "Symbolic Computing Architectures," in *Exploring Artificial Intelligence*, H. E. Shrobe, ed., Morgan Kaufman.

Sittig, D. F., K.-H. Cheung and L. Berman. (1992). "Fuzzy Classification of Hemodynamic Trends and Artifacts: Experiments with the Heart Rate." *International Journal of Clinical Monitoring and Computing*, **9**: 251-257.

Smith, B. E., P. H. King and L. Schlain. (1992). "Clinical Evaluation - Continuous Real-Time Intra-Arterial Blood Gas Monitoring During Anesthesia and Surgery by Fiber Optic Sensor." *International Journal of Clinical Monitoring and Computing*, **9**: 45-52.

Sox, H. C., Jr., M. A. Blatt, M. C. Higgins and K. I. Marton. (1988). *Medical Decision Making*. Boston, MA, Butterworths.

Steimann, F. (1994). *Diagnostic Monitoring Based on Fuzzy State Transitions*. Department of Medical Computer Sciences, University of Vienna technical report, January 1994.

Steimann, F. and K.-P. Adlassnig. (1994). "Clinical Monitoring with Fuzzy Automata." *Fuzzy Sets and Systems*, **61**(1): 37-41.

Stock, J. H. and M. W. Watson. (1988). "Variable Trends in Economic Time Series." *Journal of Economic Perspectives*, **2**(3): 147-174.

Stodley, K. D. C., D. R. Walker, A. D. Crew and J. S. Marshall. (1992). "Problems in the Development of a Computerized Ward Monitoring System for a Pediatric Intensive Care Unit." *International Journal of Clinical Monitoring and Computing*, **8**: 281-287.

Strang, G. (1988). *Linear Algebra and its Applications (third edition)*. San Diego, CA, Harcourt Brace Jovanovich.

Stroud, A. H. (1971). *Approximate Calculation of Multiple Integrals*. Englewood Cliffs, NJ, Prentice-Hall.

Szolovits, P. and S. G. Pauker. (1976). "Research on a Medical Consultation System for Taking the Present Illness." Third Illinois Conference on Medical Information Systems, Chicago, IL, 299-320.

Tanner, J. M. (1990). *Foetus into Man: Physical Growth from Conception to Maturity*. Cambridge, MA, Harvard University Press.

Tanner, J. M. and P. S. W. Davies. (1985). "Clinical Longitudinal Standards for Height and Height Velocity for North American Children." *The Journal of Pediatrics*, **107**(3): 317-329.

Terano, T., K. Asai and M. Sugeno. (1992). *Fuzzy Systems Theory and its Applications*. Boston, MA, Academic Press.

Thissen, D. and R. D. Bock. (1990). "Linear and Nonlinear Curve Fitting." In *Statistical Methods in Longitudinal Research, Volume II: Time Series and Categorical Longitudinal Data*. Academic Press, Inc.

Tsay, R. S. (1988). "Outliers, Level Shifts, ann Variance Changes in Time Series." *Journal of Forecasting*, **7**: 1-20.

Uckun, S. and B. M. Dawant. (1992). "Qualitative Modeling as a Paradigm for Diagnosis and Prediction in Critical Care Environments." *Artificial Intelligence in Medicine*, **4**(2): 127-144.

Vilain, M. and H. Kautz. (1986). "Constraint Propagation Algorithms for Temporal Reasoning." National Conference on Artificial Intelligence (AAAI-86), Philadelphia, PA, 377-382.

Voss, L., J. Walker, H. Lunt, T. Wilkin and P. Betts. (1989). "The Wessex Growth Study: First Report." *Acta Paediatrica Scandinavia (Supplement)*, **349**: 65-72.

Warner, H. R. (1979). *Computer-Assisted Medical-Decision Making*. New York, Academic Press.

Washington, R. and B. Hayes-Roth. (1989). "Input Data Management in Real-Time AI Systems." International Joint Conference on Artificial Intelligence (IJCAI-89), Detroit, MI, 250-255.

Wellman, M. P. (1988). "Book Review of *P. L. Miller, Expert Critiquing Systems: Practice-Based Medical Consultation by Computer*," *Artificial Intelligence,* **35**: 273-276.

Wellman, M. P., M. H. Eckman, C. Fleming, S. L. Marshall, F. A. Sonnenberg and S. G. Pauker. (1989). "Automated Critiquing of Medical Decision Trees." *Medical Decision Making*, **9**: 272-284.

Werth, B. (1991). "How Short is Too Short?: Marketing Human Growth Hormone." *The New York Times Magazine* June 16, 1991.

Whitely, J. R. and J. F. Davis. (1992). "Knowledge-Based Interpretation of Sensor Patterns." *Computers in Chemical Engineering*, **16**(4): 329-346.

Williamson, M. and S. Hanks. (1993). "Exploiting Domain Structure to Achieve Efficient Temporal Reasoning." International Joint Conference on Artificial Intelligence (IJCAI), Chambery, France, 152-157.

Wilson, D. (1988). "Autoregressive Growth Curves and Kalman Filtering." *Statistics in Medicine*, **7**: 73-86.

Winkel, P. (1990). "A Programming Language and a System for Automated Time and Laboratory Test Level Dependent Decision-Making During Patient Monitoring." *Computers and Biomedical Research*, **23**: 426-446.

Wittink, D. R. (1988). *The Application of Regression Analysis.* Boston, MA, Allyn and Bacon, Inc.

Wyatt, J. and D. Spiegelhalter. (1992). "Field Trials of Medical Decision-Aids: Potential Problems and Solutions." Symposium for Computer Applications in Medical Care, Washington, D.C., 3-7.

Yeh, A. (1991). "Automatically Analyzing a Steadily Beating Ventricle's Iterative Behavior Over Time." *Artificial Intelligence in Medicine*, **3**(6): 313-323.

Zadeh, L. A. (1965). "Fuzzy Sets." *Information and Control*, **8**: 338-353.

Subject Index

A

ABEL 138
accumulators for trend significance 93
adequate handbagging 83
AIS 121
ALARM 126
alarms 20, 95, 132
ANA 138
anchor point 52
applicability of TrenDx 132
assignment of data to intervals 41, 54
autoregressive model 43
autoregressive moving average model 116
average normal growth 29, 45, 73

B

baroreceptor reflex 21, 32, 46
beam search of hypotheses 66
beam size comparison 86
blood pressure trends 48
bone age 27
BUNYAN 100

C

Clinician's Assistant 54, 123
closed-loop monitoring 20
Common Lisp Object System (CLOS) 35
consecutive phases 41, 45
CONSISTENT-TEMPORAL-CONTEXTS algorithm 59
constitutional delay 30, 73
constraint-based TrenDx 103
context sensitive monitoring 21
context, temporal 53, 70
craniopharyngioma 30
Current Time landmark point 46
CUSUM for trend significance 93

D

data overload 50
data uncertainty 136
data visualization 95
densely sampled data 22
diabetic patient monitoring 49, 128
diagnostic process monitoring 19, 33
diagnostic systems 18
diagnostic theory 49
discriminatory power of value constraints 65, 161
disease histories 120
distribution assumptions 62

E

early puberty 73
economics 33
expertise of process operators 19, 50
explanation 36, 118
exponential model 42
exponentially weighted moving average 93
expressiveness of representation 35

F

false positive alarms 32
family history 27
fault model 48
filtering input data 126
financial trading 33
fluents 124
framework for diagnostic monitoring 131
fuzzy logic 124

G

general purpose usage 36
growth charts 27
growth disorders 30
growth hormone 27, 30

growth template 45
Guardian 126
guidelines for monitor sets 97

H
height age 27
HELP 20, 25
hemodynamic fault 32, 38, 83
Henderson-Hasselbach equation 42
hidden Markov model 116
hierarchical trend templates 47, 62
histories 120
hypotheses of TrenDx 41, 54

I
industrial process monitoring 33
initial classification 49
instantiating a monitor set 51
intensive care units 23, 126
intervals of trend templates 39

J
jackknifing 65

K
Kalman filters 38, 136
knowledge engineering 36, 41
knowledge level 36

L
landmark point 39, 98
learning trend templates 100
linear regression 42
liver disorder template 105
logarithmic model 42

M
macroeconomic monitoring 23
macroeconomics 33
matching data to predictions 20
mean absolute percentage error 60
measurement error 37
M-HTP 123
MIMIC 126
monitor set 48, 51
monitor set critiquer 95, 100
multiple disorders 48, 138

N
natural language 138
nearest satisfier value constraint 64
neural networks 117
normal model 48
Now landmark point 46

O
open-loop monitoring 20
ordinary least squares 60
oxygen handbagging 38

P
parallel hypothesis evaluation 110
parameter 37
parameter estimation 36
pathophysiologic model 23
persistence 37, 123, 128
persistence heuristic 110
pharmacokinetic models 126, 128
phase transition estimates 86, 110
polynomial model 42, 43
power plant monitoring 33
prediction 20, 22
prediction granularity 68, 87
PREMON 126
Present Illness Program 18, 104, 122
primitives 36
probabilistic networks 124
process histories 120
process monitoring 19
PROCESS-DATA-CLUSTER algorithm 55
pruning hypotheses 20

Q
QSIM 120, 126
qualitative estimate 43

Qualitative Process Theory 120, 126
qualitative simulation 126
qualitative superposition 120, 138

R

recommending tests 137
reference intervals, problems 18
regression analysis 114
removing outliers in regression 65
repeated iteration of chronologies 111
RESUME 128
runs for trend significance 93
RX 123

S

sensitivity of alarms 35, 77, 83, 109
sexual development 27
significance of a trend 91
SIMON 126
Skeletal Dysplasia Diagnostician (SDD) 121
sound inference 66
sparsely sampled data 22
specificity of alarms 32, 36, 77, 82
spectral analysis 116
standard error 136
state-based feature recognition (SBFR) 127
subjective trend evaluation 139
system identification 113

T

Tanner stages 27
TCS 128, 129
telecommunication monitoring 34
temporal abstraction 20, 123
temporal constraints 39
temporal distances 40
TEXT 138
thresholds for trend significance 92
THRIPHT 122
thyroid hormone 27
time intervals 39
Time Map Manager (TMM) 57, 123
time series analysis 115
TOPAZ 128, 138
training data 22
trend template 35
trend, definition 17
TrenDx 48, 51
TrenDx for trend significance 94
triggering disorders 104
trigonometric model 42
triple-logistic curve 114
truth maintenance system 129
TUP 40, 122, 123, 129, 137

U

ultimate etiologies 50

V

value constraints 41, 103
value intervals 136
VentPlan 126
VM 127

Z

z-score 29, 45