Diagnostic Process Monitoring with Temporally Uncertain Models

by

Steven M. Bull

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2002

Author . . . .
Department of Electrical Engineering and Computer Science
May 24, 2002

Certified by . . . .
Peter Szolovits
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Diagnostic Process Monitoring with Temporally Uncertain Models

by

Steven M. Bull

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2002, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

This thesis develops a real-time trend detection and monitoring system based on previous work by Haimowitz, Le, and DeSouza [3, 5, 2]. The monitor they designed, TrenDx, used trend templates in which the temporal points where data patterns change are variable with respect to the actual process data. This thesis uses similar models to construct a monitoring system that is able to run in real time, based on a continuous, linearly segmented process data input stream. The instantiation of temporally significant template points against the process data is determined through a simulated annealing algorithm. The rankings of competing hypotheses in the monitor set is based on the distance of these template points from their expected temporal values, along with the area between the process data measurements and the value constraints placed on those parameters. The feasibility of the real-time monitor was evaluated in the domain of pediatric growth, particularly in comparison to previous versions of TrenDx, using an expert gold standard of the diagnoses of pediatric endocrinologists. Real-time TrenDx shows promise in its monitoring abilities and should be evaluated in other domains which are more suited to its continuous data stream input model.

Thesis Supervisor: Peter Szolovits
Title: Professor of Computer Science and Engineering

This work was supported (in part) by DARPA contract F30602-99-1-0509, Adaptive Knowledge-Based Monitoring for Information Assurance, administered through the Air Force Research Laboratory.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In many fields, problem solving can be described as observing a situation, identifying what the problem is (or determining if a problem is even present), and applying an appropriate method to solve it. Depending on the particular field, each of these steps introduces various amounts of difficulty. Often only an expert in the field has the required skills and knowledge to diagnose the situation accurately.

To quantify the observation stage, it is reasonable to assume that there is a set of measurable properties related to the situation. In many interesting domains, these properties are able to change over time. Experts are able to recognize which properties of the system are important, how they relate to one another, and if they are changing in a significant way. Using these observations and knowledge of the dynamics of the system, experts are able to differentiate among different problem conditions and identify possible causes of the problem. Their decisions are supported with evidence from the measured properties and models of the system behavior.

Many systems exist that can be used to monitor a process and control its evolution, provided that there is a model of the dynamics of the process that is well-specified in time. In many domains, however, processes do not have well-defined models. One cause of uncertainty may be due to the fact that a process may start in one of several initial states and behave according to a model that is dependent on its initial state, but there is no clear way to determine the initial state in a reasonable amount of time. In other situations, different phases of a process may be well understood (and have different models), but there is no way to determine when the process moves from one phase to the next. There is a need for a process monitor that can handle each of these situations.

In recent years, progress has been made by Haimowitz, Le, and DeSouza [3, 5, 2] in developing a monitoring system called TrenDx. TrenDx is able to be used in domains in which trends of process parameters are described with qualitative terms and points of temporal significance that may vary from one data set to the next. The previous implementations of TrenDx have shown promise in their diagnostic abilities. This thesis introduces revisions to the hypothesis-scoring algorithm of TrenDx, while maintaining the trend template description language introduced by Haimowitz. The monitoring revisions are intended to improve the efficiency of TrenDx and make it usable in a real-time processing context. The following section explains the importance of a process monitor such as TrenDx, and discusses many of the considerations involved in the initial design of TrenDx.

1.1 Importance

The goal of this thesis is to further develop a system that is able to monitor processes that evolve with time and diagnose potential problems that the processes may be experiencing. In particular, the monitor must be able to detect multivariate trends in time-ordered process data based on inexact models of the process. The models may consist of descriptions of how the process progresses in each of several phases that are ordered relative to one another in time. Within each phase, the model may describe different trends that the process data may follow, based on the state of the process during that phase. Furthermore, the expected length of each phase may be dependent on the current state of the process, and therefore dependent on the history of all states of the process. The monitor must be able to adjust to match different trends during different intervals in spite of the fact that the interval endpoints are not precisely specified in time. The monitor needs to be able to detect landmark events that may signal the beginning or end of an interval, and adapt its future reasoning about the data once one of these landmark events has been detected.

The processes that the monitor will be used to diagnose may be modeled with inexact descriptions of the way variables behave temporally and the ways in which they interact with one another. In many domains, these characteristics make it difficult for novices (and even for experts) to differentiate among some similar trends. If the monitor is to be trusted in these domains, it will need to be able to justify the reasoning it used to draw its conclusions. This explanation should be thorough enough that an expert is satisfied that no other diagnosis of the process is a particularly better match.

A monitor that satisfies these requirements would be of great use in a variety of fields. It could be used as a reference aide to help train specialists in a domain. It could be used for supporting evidence in situations in which non-experts are responsible for proposing a diagnosis. It could be used in situations overlooked by experts, when too few experts are available to closely monitor all of the simultaneously evolving processes. In essence, a monitor of this type could save time and effort for many people, which is one of the great benefits of technology in general.

1.1.1 Difficulty of solving the problem

Representation considerations

There are several difficulties that arise in an attempt to design a diagnostic process monitor like that described above. Some of the biggest decisions that need to be made are the determinations of how knowledge of the process as well as knowledge of the models is to be represented. A large benefit of this monitor as opposed to other monitors is that it is able to handle temporally uncertain events. This introduces constraints on the structures that can be used to represent time. The time representation must allow for the input of time-ordered (and usually discretely time-stamped) data. It must be flexible enough to allow for the models to contain landmark events in trends, but allow those events to be temporally uncertain. The time structure needs to have enough power to express both discrete events and events that span a length of time. Models may contain trends in which some variables are to vary in a particular fashion within certain phases of the process, while other variables should follow trends over several phases of the process. The time representation must take each of these requirements into account.

A related representation consideration arises during the need to express models. There should be no loss of knowledge between the description of a process model and its representation in the monitor. If order for this requirement to be satisfied, the model representation must be able to express how variables interact with one another as well as how they vary over time. The model may have particular upper and lower bounds on data values, requiring the monitor to have a representation for a description of the ways in which data variables can vary from a trend but still be considered to match the model. The model descriptions should generally be high-level, but the monitor may need to do precise computations with models: the model representation needs to allow for these low-level calculations while only being described in high-level terms.

Complexity

Finding the best diagnosis for a monitored process breaks down into finding the closest match of the available models to the set of measured data. With a fixed number of models in which all temporal boundaries are fully specified, this search is tractable, and there are efficient methods of finding the best diagnosis. When the temporal boundaries are allowed to be unspecified, however, the problem becomes much harder. In a process that contains only 3 phases and one monitored variable, with constraints such that first phase boundary is between any of the first $\frac{n}{2}$ data points and the second phase boundary falls between any of the last $\frac{n}{2}$ data points, the number of possibilities to search is already $(\frac{n}{2})^2$ per model template. For multiple phases and multiple possible behaviors of the variables within each phase, the problem to find the best match quickly becomes combinatorial in complexity, which is computationally intractable. Another way to think about this problem is from a constraint-satisfaction point of view. For each temporal phase boundary, there will be some set of constraints. For the data within each phase, there is another set of constraints for each model. This can be considered as a multi-dimensional constraint satisfaction problem. Unfortunately, this view does not ease the computational complexity of the problem.

Applicability

It is important that the framework of the monitor that is developed be useful in a wide variety of domains. An obvious reason for this is that it is impractical to require the development of a new monitor system for each domain in which process monitoring is required. Generally, the people with expertise in a domain are not adept at designing an automated monitor such as this. Furthermore, experts may not be able to express their knowledge in algorithmic form without an appropriate description language as a basis. Generally, it is more practical to have a person with expert knowledge use their skills in the ways they have been trained than to employ them as diagnostic process monitor system developers.

In order to be of use in a variety of applications, the monitor must facilitate the conversion of knowledge into the representations that the monitor understands. Experts should be able to add knowledge to the monitor with ease as their understanding of the process improves. Without efficient representations of time and models, the monitor will be ineffective in both diagnostic ability and usability in application domains.

1.1.2 Application domains

There are several domains in which a monitor with these capabilities would be useful. One of the most obvious areas in which this monitor would have great use is the field of medicine. Often, medical conditions are described in an imprecise language, such as, "If Y increases while X increases, and then Y decreases after X starts to decrease, then condition Z is present." The temporal boundaries on the significant events in descriptions such as this are not specified, and multiple diagnoses are often possible, so these situations would receive great benefit from the proposed monitoring system. The medical field also demonstrates two different situations in which this monitor would be helpful. In intensive care units, process data are measured nearly continuously. The monitor should be able to diagnose the situation in real time, and either alert medical professionals or modify treatment when particular conditions are detected. On the other hand, the monitor would be useful in settings such as a pediatrician's practice. In general, children's height and weight are recorded a few times each year. Growth dysfunctions are often difficult to detect, but it is not reasonable to send every child to a growth specialist. The monitor could alert the pediatrician when additional investigation into the growth condition of the child should be made.

Another domain in which this sort of monitoring would be helpful is the maintenance of computer networks. It may be desirable that certain activities are avoided on the network, such as an abnormally high amount of traffic during busy network times being accountable to one particular machine. Furthermore, a monitor of this type could be used to detect security problems and potential attacks on a network or on a machine, based on the rate and type of information traveling across the network. Additionally, a monitor may be used to detect when a machine has been infected with a computer virus or is overloaded with inessential tasks, based on the performance statistics of the computer. Similarly, applications can be found in economics, industrial processes, and many other areas in which crude models of behavior are known, but the complexity of the system make it such that the points of temporal significance are not precisely determined.

1.2 Aims of research

The intent of this thesis is to reformulate TrenDx, a diagnostic process monitor that was developed with many of the previously mentioned qualities in mind. The original implementation of TrenDx was applied in the medical domains of Intensive Care Unit monitoring and pediatric growth monitoring. The results of the experiments were promising in both domains. However, it took TrenDx too long to process data to be feasible for use in an Intensive Care Unit, where new data are constantly being acquired and decisions are very time critical. TrenDx was further developed and tested in the domain of monitoring children's growth.

The goal of this research was to develop a real-time version of TrenDx, without losing the template description capabilities of the original monitor. TrenDx could express both temporal constraints and value constraints in its trend templates, and these models were to be preserved. However, the input model to TrenDx was revised to a streaming data input model, and therefore new ways of scoring competing behavioral models needed to be developed. TrenDx was written over the course of several years by at least 3 different people, resulting in code that is not well organized. A large part of the work involved in this thesis was rewriting TrenDx according to the principles of software engineering.

The revised version of TrenDx was evaluated in the domain of children's growth monitoring. It was compared to previous versions of TrenDx, both in terms of efficiency and in accuracy of its results.

1.3 Guide to this thesis

The remainder of this thesis is organized as follows. The original TrenDx monitor system, along with the enhancements previously made to it, is described in Chapter 2. The revisions made to TrenDx as part of this thesis follow in Chapter 3, which discusses the changes made to the models used in TrenDx, and Chapter 4, which describes how monitoring is performed with these revisions. Chapter 5 introduces the domain of pediatric growth, and discusses the results of the evaluation of the revised monitor in this domain. Finally, Chapter 6 presents the conclusions of this thesis, describing related research and suggestions for future work on TrenDx.

Chapter 2

TrenDx

A monitoring system with many of the desired characteristics described previously has been developed over the last several years. This system, TrenDx, was used as the basis for the development of the monitoring system presented in this thesis. The trend template models presented in the original implementation of TrenDx have been largely preserved in the revised version of the monitor. These models provide the ability to describe both qualitative and quantitative constraints on the process data in addition to temporally uncertain interval boundary points. The modifications made to TrenDx significantly changed the methods by which competing trend templates were matched and ranked against the process data, which were the primary cause of the inefficiencies in previous versions of TrenDx. This thesis develops real-time methods of matching the trend template models of TrenDx to the process being monitored.

For a more complete description of TrenDx, see the original description by Haimowitz [3], and improvements made by Le [5] and DeSouza [2]. The aspects of TrenDx that are most relevant to this thesis are discussed below.

2.1 Template representation

The largest contribution of the previously implemented versions of TrenDx to the monitoring system presented in this thesis comes in the representation language of trend templates. Trend templates are competing descriptions of the states of the monitored process. They contain the knowledge that an expert uses to diagnose the process, expressed in the form of models that a machine can use for computation.

There are two categories of knowledge that are contained in trend templates. The first of these is a description of temporal relations which specify when transitions occur among various phases of a process. The other type of knowledge is an explanation of what values each of the measured parameters is expected to take during each phase of the possible behavioral states in which the process may currently be. A description of the way in which TrenDx modeled these two types of knowledge is described in this section.

2.1.1 Temporal intervals

Temporal Utility Package

The manner in which TrenDx represents time is based heavily on the Temporal Utility Package (TUP) designed by Kohane [4]. TUP was designed with the idea in mind of separating the temporal reasoning component of an expert system from the remainder of the system. TUP includes structures that represent both points in time and temporal intervals. TUP can represent quantitative and qualitative temporal relations among these time structures. Additionally, relations among points and intervals may have different values depending on the context in which the relation is interpreted. TUP is able to make temporal inferences through the use of constraint propagation. TUP uses temporal-clustering heuristics to maintain its computational feasibility as a temporal reasoning tool.

Interval relations

TrenDx defines two temporal structures: temporal intervals and landmark events. Temporal intervals correspond to lengths of time in TUP. Intuitively, each interval has a left ("begin") and right ("end") boundary point. The length of a temporal interval is described in terms of its boundary points. Relations are defined between a pair of boundary points in terms of a minimum and maximum temporal distance between the pair, for example,

```
((begin interval_A) (end interval_A) 5 7)
```

expresses that `interval_A` has a length of between 5 and 7 time units. Furthermore,

```
((end interval_A) (begin interval_B) 0 4)
```

expresses that the end of `interval_A` and the beginning of `interval_B` are separated by at most 4 time units (and `interval_A` ends before `interval_B` begins). These relations are depicted graphically in Figure 2-1.

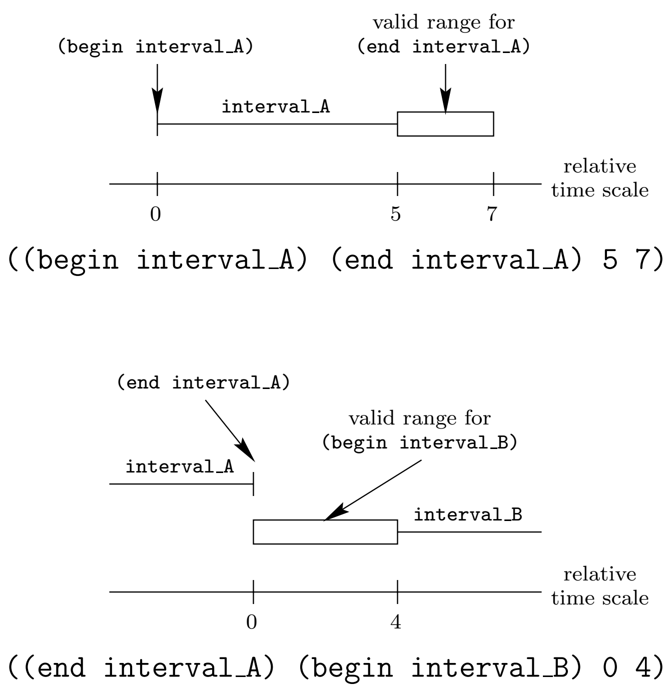


Figure 2-1: Interval relation examples.

Landmark events

Landmark events correspond to time points in TUP. The reasoning engine in TUP considers time points as a subset of temporal intervals, particularly temporal intervals with zero length. In this way, landmark events and temporal intervals share many temporal relation expression capabilities. In addition to being related to the boundary of a temporal interval, landmark events may be fixed in time with relation to the monitored process. For example, the landmark event of `birth` can be fixed to time 0.0 in the process data. Additionally, a special landmark event of `now` may exist in trend templates. `now` is always associated with the most recently acquired data point.

Landmark events participate in relationships in the same way as boundary points. An example of a temporal constraint involving landmark events is

`(start (end interval_B) 14 19)`

which describes that relation that `interval_B` ends between 14 and 19 time units after the landmark event `start`. Figure 2-2 displays this relation in graphic form.

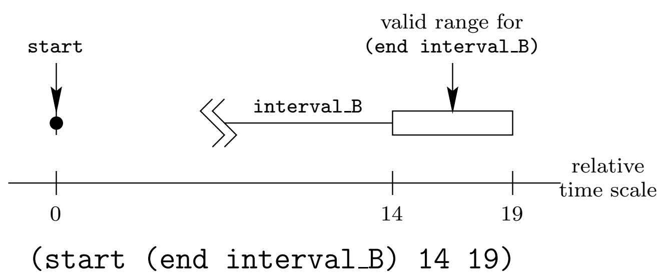


Figure 2-2: Landmark event relation example.

2.1.2 Value constraints

Value constraints are what TrenDx uses to describe the expected levels of the measured parameters when the process is behaving according to a particular model. Value constraints are not independent; each must be associated with a specific temporal interval. With this combination of constraints, the expected trends of each parameter during each phase of a behavioral pattern can be described.

TrenDx provided the ability to express value constraints in terms of functions of the measured process parameters in addition to the measurements themselves. For instance, a value constraint could express a pattern describing the trend of the average value of all of the measured parameters. This ability is useful because many models are described in terms such as these, that is, they constrain parameters that are not readily available from standard measuring equipment, but which may be derived from those measurements.

TrenDx allowed value constraints to be expressed in terms of low-order polynomials: constant, linear, or quadratic. Examples of how these constraints could fit to a set of data are shown in Figure 2-3. The linear and quadratic constraints required at least a qualitative parameter to describe the desired trend. A linear constraint must be described as either increasing or decreasing, while a quadratic constraint would additionally require a specification of concavity. Constant and linear constraints were also permitted to specify an exact constraint parameter to describe the trend. This additional descriptiveness allowed the expression of constraints such as "constant at 98.6 degrees" or "(linearly) increasing at a rate of two percent per year".

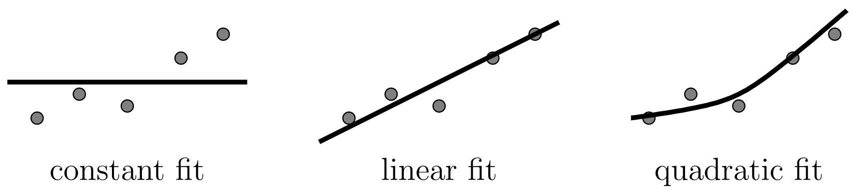


Figure 2-3: Various polynomial constraints against a data set.

2.2 Monitoring paradigm

The monitoring scheme of TrenDx was based on the idea that if competing models were constructed, representing different diagnoses of the monitored process, the model for the diagnosis which accurately described the process data would prove to be a better match to the data than the other competing model templates. In the language of TrenDx, each competing trend template is called a hypothesis. Naturally, only the hypotheses applicable to the particular process would be included in the monitoring set of hypotheses for that process. The primary computational work of TrenDx is accounted for in determining how well hypotheses fit the measured process data, in comparison to the other hypotheses in the monitor set.

The trend template representation described in the previous section was preserved during the revising of TrenDx for this thesis. The areas of TrenDx that were modified to improve efficiency are described in this section. The scoring procedures for value constraints were changed, but closely resemble the original scoring methods. The hypothesis scoring and searching techniques were greatly modified, but are presented here for completeness and comparison.

2.2.1 Value constraint scoring

In order to rank the competing hypothesis in the monitor set, TrenDx assigns a score to each hypothesis. This score is produced by first using TUP to generate all valid mappings of data points to temporal intervals defined in the hypothesis. Each of these mappings is then given an error score. The error score is computed using the value constraints on the data in each temporal interval. If the value constraint is not completely specified, a polynomial regression is computed to choose the best parameters for the constraint. For example, if the value constraint specifies a constant trend, but does not indicate that a particular constant value is desired, the best value is chosen to minimize the total least-squares error over all of the data points assigned to that temporal interval. Furthermore, if a linearly increasing trend is prescribed, but a polynomial regression dictates a line with negative slope, the slope of the constraint is set to zero. This would be the best possible slope of a trend line that does not violate the qualitative value constraint placed on the data.

After the value constraint is fully specified, the data points assigned to the interval are used to accumulate an error score for that particular constraint. The most straightforward calculation of this type is the least squares error calculation:

$$\text{ordinary least squares error} = \sum_i (x_i - \hat{x}_i)^2$$

where the sum is over each data point i assigned to the interval, x_i is the data point value, and $\hat{x}_i$ is the value predicted by the value constraint. A slightly more meaningful error measure that is used by TrenDx is the residual mean square error, in which the squared error is scaled by the number of degrees of freedom in the regression fit:

$$\text{residual mean square error} = \frac{1}{\textsc{DegreesOfFreedom}} \sum_i (x_i - \hat{x}_i)^2$$

The degrees of freedom in a regression fit is calculated as the difference between the number of data points being fit and the number of parameters being estimated in the regression. For example, there are 2 parameters estimated in a linear regression (corresponding to slope and y-intercept). If there are 5 data points being fit by this model, the number of degrees of freedom is equal to 3. In the case that the degrees of freedom calculation is less than 1, the regression fit will match the data perfectly, and any error score should be zero, in which case the value given by the residual mean square error formula is meaningless.

Another error measure used by TrenDx is the mean absolute percentage error. This error measure is useful for combining error measurements of parameters with varying magnitudes.

$$\text{mean absolute percentage error} = \frac{1}{\textsc{DegreesOfFreedom}} \sum_i \left| \frac{x_i - \hat{x}_i}{x_i} \right|$$

The mean absolute percentage error normalizes the variances of variables with different magnitudes. Unfortunately, for many value constraints, $\hat{x}_i$ is zero for at least one i (a common example being a desired constant trend at a measurement value of zero). Furthermore, many data measurements return values close to zero, while a value constraint may prescribe a nonzero value. This can cause artificially large error values. For example, assume a value constraint of a constant at a level of 1, and the process data have a variance of 2. Suppose also that two data points, x_0 and x_1 were assigned to the temporal interval containing this constraint. Assume x_0 = -0.5 and x_1 = 0.01. Let $\textsc{Error}_i = \left|\frac{x_i - \hat{x}_i}{x_i}\right|$. The calculations of the mean absolute percentage error would proceed as follows:

$$\begin{aligned}
\text{mean absolute percentage error} &= \frac{1}{\textsc{DegreesOfFreedom}} \sum_i \left| \frac{x_i - \hat{x}_i}{x_i} \right| \\
&= \frac{1}{\textsc{DegreesOfFreedom}} (\textsc{Error}_0 + \textsc{Error}_1)
\end{aligned}$$

$$\textsc{Error}_0 = \left| \frac{x_0 - \hat{x}_0}{x_0} \right| = \left| \frac{-0.5 - 1}{-0.5} \right| = \left| \frac{-1.5}{-0.5} \right| = 3$$

$$
\begin{aligned}
\textsc{Error}_1 &= \left|\frac{x_1 - \hat{x}_1}{x_1}\right| = \left|\frac{0.01 - 1}{0.01}\right| = \left|\frac{-0.99}{0.01}\right| = 99 \\
\text{mean absolute percentage error} &= \frac{1}{\textsc{DegreesOfFreedom}}\left(\textsc{Error}_0 + \textsc{Error}_1\right) \\
&= \frac{1}{2}(3 + 99) = \frac{1}{2} \cdot 102 = 51
\end{aligned}
$$

In this example, the value of x_1 (0.01) is significantly closer to the desired value of 1 than x_0 (-0.5), but x_1 contributes a disproportionate amount to the total error score.

Because of these inadequacies in the mean absolute percentage error calculation, TrenDx used a combination of the mean absolute percentage error and the residual mean square error metrics in determining an error score for a particular value constraint. TrenDx allowed the user to specify which metric it should use for each constraint.

In addition to these error metrics, TrenDx had the ability to pre-process data values before fitting them to a value constraint. This ability was most commonly used to express a constraint in the form of a desired range of values that the process data should take. Many constraints are described in language such as "the temperature should be kept between 35 and 45 degrees". With constraints like this, it is generally intended that a measurement in the middle of the desired value range is a good fit to the constraint, while measurements closer to the boundaries of the range are a poorer fit, and those outside the range do not match the constraint. To incorporate constraints of this nature, TrenDx was constructed to allow the domain expert to express a constraint of this type via pre-processing the data with an error curve similar in shape to the curve in Figure 2-4.

When applying this error function, a data measurement that is exactly in between the upper and lower bounds of the range gets mapped to the value of 0. Data measurements outside the desired value range get mapped to a value of 1. Within the desired value range, the distance from the data measurement to the midpoint of the range is raised to the fourth power, after being normalized so that the function

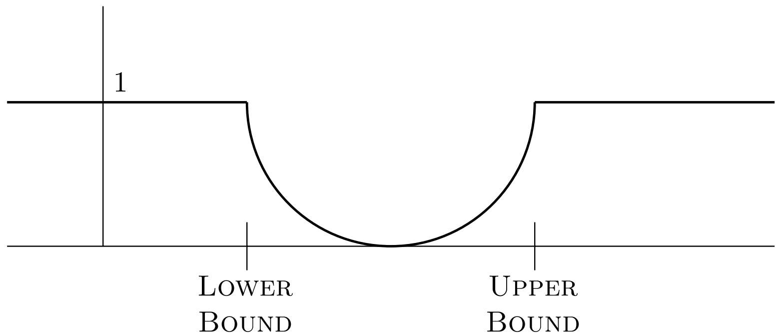


Figure 2-4: Pre-processing error function.

is continuous at the range boundaries. The resulting mapped data values are then commonly used with a desired value constraint of a constant value of 0.

2.2.2 Hypothesis scoring

To produce an error score for an entire hypothesis, TrenDx computes a weighted average of the error scores produced by the fitting of each value constraint in the hypothesis. The weight used in this average is equal to the fraction of the total number of degrees of freedom accounted for by each value constraint.

$$\text{hypothesis score} = \frac{\sum_j \textsc{DegreesOfFreedom}_j \cdot \textsc{ValueConstraintError}_j}{\sum_j \textsc{DegreesOfFreedom}_j}$$

The sums are over all value constraints j in the hypothesis trend template. DegreesOfFreedom$_j$ is the number of degrees of freedom in value constraint j, and ValueConstraintError$_j$ is the error score resulting from the fitting of value constraint j. As described previously, the error score of a value constraint is either a residual mean square error value or a mean absolute percentage error value. In either of those two calculations, DegreesOfFreedom occurs in the denominator. In the combined hypothesis score, DegreesOfFreedom occurs in the sum in the numerator. These two occurrences result in a term-by-term cancellation in the sum in the numerator of the hypothesis score. Therefore the degrees of freedom is only important in the denominator of the hypothesis score equation, which can lead to more efficient comparisons of competing hypotheses, if the total degrees of freedom are expected to be equal across hypotheses.

This form of value constraint error weighting tends to normalize the importance of each value constraint in the combined hypothesis score. In many cases, this is the desired behavior when constraints are placed on varying parameters in different intervals. However, some situations require disproportionate weighting of different value constraints. For example, running low in windshield washer fluid in a car is generally not a particularly serious condition, but having an abnormally high engine temperature requires immediate care (otherwise total engine failure may result). For situations such as this, TrenDx allows the knowledge engineer to provide explicit weights for each value constraint as part of the value constraint definition.

2.2.3 Ranking hypotheses

The score used by TrenDx in ranking competing hypotheses is the best (least error) score found for each hypothesis. The search space of all possible valid assignments of data points to temporal intervals can be overwhelmingly large, particularly when many data points are present. To compensate for this problem, TrenDx uses a beam search to prune the set of assignments of data points to temporal intervals that it considers. When a new data point is processed, TrenDx first produces all valid assignments of that data point to temporal intervals, based on the assignments of previously encountered data points that it has already decided. It then retrieves the entire hypothesis score for each of these potential assignments. At this point in the computation, these competing scores for different temporal data assignments within the same hypothesis are ranked, and the pruning beam is applied. TrenDx uses a default beam width of 3. The assignments that score the best are kept to be used as the basis for possible valid assignments when the next data point is processed. The very best scoring of these assignments is used to produce the score for the hypothesis as a whole.

To rank competing hypotheses in a monitor set, TrenDx compares the score of each hypothesis. Hypotheses with lower scores are considered better fits to the process data than those with higher scores. To translate this to a monitoring decision, several metrics may be used. The procedure that was used primarily in the development of TrenDx has been to classify each hypothesis as normal or abnormal. The score of the best-fitting normal hypothesis was then compared to a threshold value, which would determine whether or not to trigger a warning signal for the user.

2.2.4 Efficiency

The original implementation of TrenDx left much room for improvement. Originally, TrenDx took several hours to process just a handful of data points and produce meaningful results. Since the original implementation, it has been substantially improved. One of the primary factors increasing its speed was the general improvement in computer hardware technology over the course of a decade. Improved programming techniques also trimmed the time required to process data by reducing the computations performed when the data fit the constraints in a hypothesis trivially. However, TrenDx continued to suffer from a few setbacks that prohibited its incorporation into other monitoring systems.

One of the most significant problems that TrenDx suffered from was its failure to memoize results of computations that it had previously evaluated. When TrenDx scored a data set, it began from the earliest point in the data set and used the beam search technique with hypothesis scoring, expanding by one data point at a time, until it had reached the last data point, at which point it would produce an output. This required that all data points were input before the computation began. If an additional data point were to be added to the data set, TrenDx was not able to use the results of a calculation based on a smaller data set and incorporate the new data. Instead, TrenDx would reset itself and start from the first data point in the set, assigning each point to an interval and scoring each hypothesis as before. This technique is extremely inefficient for a process in which input data are continually being acquired.

Another area of concern in TrenDx was the technique it used to determine the best assignment of data to intervals in a hypothesis. As described previously, a beam search was used. Just as in hill-climbing search techniques, beam searches can become trapped in sub-optimal paths, which would only lead to a locally minimum error score as opposed to the globally minimum error score for the hypothesis. Unfortunately, the alternatives available to explore all possible data assignments in order to find the optimum solution add significant computational time to the search. Dynamic programming techniques can be used to find the best solution in time on the order of $(N + I)^3$, where there are N data points and I intervals in the hypothesis. In contrast, beam search typically runs in time linear in $(N + I)$.

Correcting these inefficiencies in the monitoring program is the primary goal of this thesis.

Chapter 3

Model Revisions to TrenDx

During the development of an improved version of TrenDx, a few revisions were made to the problem models used in the original implementation of TrenDx. The largest conceptual change to the problem description was made in the model of the input process data. Additionally, a few changes were made in the trend template description model. The modifications to the trend template descriptions were small, and were primarily necessary for implementation of the real-time processing algorithm. These changes are extensions to the trend template models introduced by Haimowitz [3] for purposes of computation, but do not do much to enhance the descriptive power of the original TrenDx trend templates.

All of these changes are discussed in this chapter.

3.1 Input process data model

The most significant change in the input process data model made in the revised TrenDx is the assumption that the input data are continuous. Furthermore, the program was constructed with a model of streaming input, as opposed to requiring all data to be stored in a file before running the program. The primary work of this thesis involved designing a hypothesis scoring framework under this revised input data model.

3.1.1 Continuous data stream

In the original TrenDx program, process data were input at discrete points in time. Since that time, it was determined that there is a greater need for a monitor whose input is modeled by a continuous function. There are relatively few types of measured process data that only have meaning at discrete points in time as opposed to continuously. That is, most parameters are (at least conceptually) measurable at any point in time, not purely at discrete time points. In general, the measurements of these parameters result in continuous values, if the measurements are taken at a fine enough temporal granularity. Occasionally, however, true discontinuities may occur in the process parameters that are of interest to the monitoring system. In order to fit these discontinuities into a model framework of continuous input data, a discontinuity can be modeled by an arbitrarily small temporal interval during which the magnitude of the slope of the data is arbitrarily large. In this context, “arbitrarily small” and “arbitrarily large” are intended to mean that the differences between a true discontinuity and this continuous construct are negligible to the monitoring system.

It should be noted that digital measurements of process data are required in order to use the data in computations. Digital measurements are intrinsically discrete. The principal revision made to the input model in terms of continuity is the assumption that the data points provided to TrenDx are sufficient to accurately represent the underlying continuous process. The previous versions of TrenDx did not make this assumption, but instead relied on the design of the trend templates for a particular domain to take into account the sampling frequency of the process parameters.

The streaming data model modification is important to note because the revised TrenDx is intended to be a real-time monitoring system. This means that it can regularly be receiving new data and not require breaks to do its computations. It should be able to use each new data point to update its state without restarting its calculations from the beginning. As discussed previously, the original TrenDx suffered from performance slowdowns due to its static data input model.

One of the advantages of a continuous data model is that continuous functions can be well approximated with a series of simple functions. In particular, within a given error value, all continuous functions can be modeled through linear segments over sufficiently short temporal intervals. Additionally, most higher-order functions of interest can be well described by linear segments between special points on the original functions, such as local extrema, inflection points, and even at the temporal locations of the zero crossings of higher-order derivatives of some functions. The following section describes how this ability is taken advantage of in the improved version of TrenDx.

3.1.2 Data stream segmentation algorithm

The revised version of TrenDx uses a linearly segmented description of the input process data in order to perform hypothesis scoring. Due to this fact, before the data are ready to be scored, they must be broken into linear segments. The input data model discussed above calls for a continuous data input stream. Dr. William Long has developed a method for producing a series of continuous linear segments from a series of data points in real-time. His segmentation algorithm is the focus of this section.

Dr. Long's segmentation algorithm takes a series of data points as input. In practice, many measuring instruments return updated digital values at a rate primarily restricted by the hardware limitations of the measuring device. If a continuous analog device is being used to measure process parameters, an analog-to-digital conversion may be performed on the input measurements to produce data points appropriate for the segmentation algorithm. In essence, sampling a continuous data stream produces a discretely time-stamped data stream. If this is done, care should be taken as to not lose too much information during the sampling process.

The driving force behind the development of a segmentation algorithm that operates without knowledge of templates is the desire to detect temporally significant behavioral changing points before template matching is attempted. To do this, Dr. Long's algorithm uses accumulated values to determine when one linear segment no longer matches the data points well enough from the previously determined anchor point. This decision is based on an error threshold. If the error in a linear regression would exceed the threshold, a new anchor point is found. The temporal location of this new anchor point is determined by considering making an anchor at the temporal location of each data point since the most recent anchor. An error score is produced by fitting the data points since the previous anchor to two linear segments that intersect temporally at the location under consideration. The new anchor is inserted at the location that provides the lowest error score. The series of anchor points produced in this manner serve to break the data, at temporally interesting points, into linear segments that accurately describe the data set. Dr. Long's algorithm is depicted in Figure 3-1.

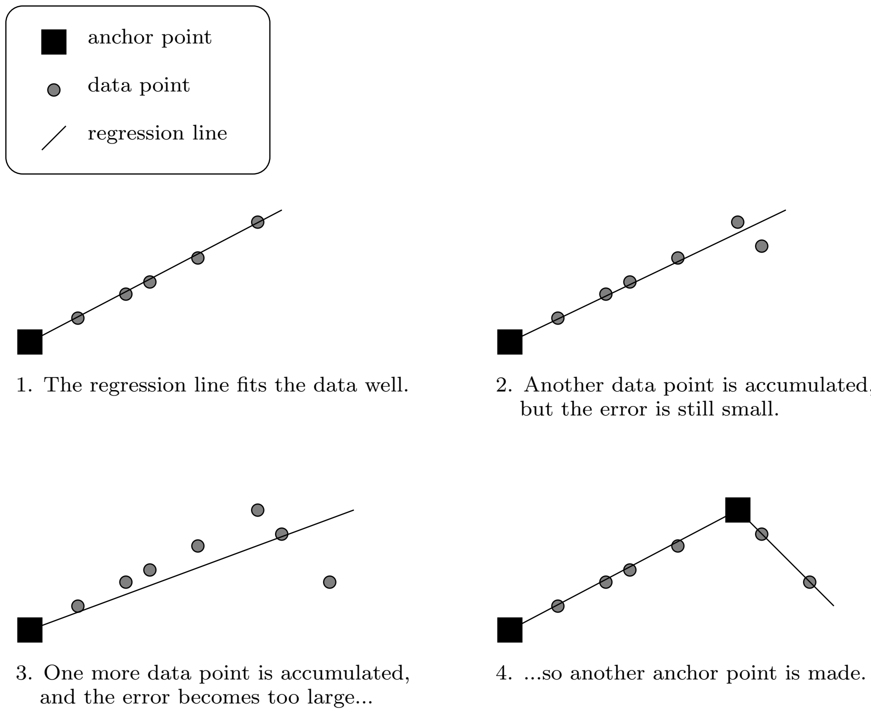


Figure 3-1: Segmentation algorithm.

This segmentation algorithm is intended to be used for real-time trend detection purposes. A constant amount of work is required for each new data point encountered until the error threshold has been broken. This is due to the fact that a list of values is accumulated as points are gathered, and these accumulated values are used to determine when the error threshold has been broken. Unfortunately, the amount of work required to find the best new anchor point is quadratic in the number of points encountered since the temporal location of the most recent anchor point. However, computations can be performed in a manner such that linear work is done while processing each point to avoid the quadratic load all at once. With improving processing speeds, this is becoming increasingly feasible to be used in real-time computations, particularly due to the latency between availability of new data points. Additionally, in many situations, improved searching methods could be used to reduce the load at each point from linear to logarithmic. Furthermore, this processing time can be bounded by restricting the number of data points allowed to accumulate before introducing a new anchor point. This technique bounds the computation requirements to a constant, which certainly results in a real-time segmentation algorithm.

3.2 Trend templates

In addition to modifications to the process data input model, a few revisions were made to the trend template description requirements during the updating of TrenDx. These changes were necessary for the computations in the new version of TrenDx, but they do not reduce the power of the TrenDx model. One new requirement is the need for each interval boundary point to be explicitly declared. The other required change is the need for a distinction among which of these points are related to each other, either directly or indirectly. Both of these changes help to clarify the trend template model. The model descriptions are still primarily identical to those developed by Haimowitz [3], with the minor changes added. These changes are discussed in depth in this section.

3.2.1 Temporal interval boundary points

One change in the trend template model from that of the original implementation of TrenDx is the need to explicitly declare boundary points of temporal intervals. The original model for temporal interval descriptions was discussed in section 2.1.1. In the new version of TrenDx, each boundary point of a temporal interval is declared and named similarly to landmark points in the original implementation. In the Java implementation of the revised TrenDx, these points are now instances of the class `TemplatePoint`.

`TemplatePoint`s in the new implementation share all of the features of landmark points in the original implementation. Their locations can be fixed in time or variable in relation to the process data. In constructing a trend template, the locations of `TemplatePoint`s can be specified as absolute or relative to the locations of other `TemplatePoint`s. The special landmark point of `now` is not lost in the new implementation. It is a `TemplatePoint` whose relative distance from the most recent data point is zero. There are two types of conversions needed to replace trend template models from the original TrenDx description language with descriptions in the new implementation, which pertain to relations between adjacent intervals and relations between non-adjacent intervals.

Adjacent intervals

The conversion from adjacent consecutive intervals in the original TrenDx description language to the new description language is straightforward. A frequently-used construct in the original TrenDx was the following:

```
(consecutive-phase interval_A interval_B)
```

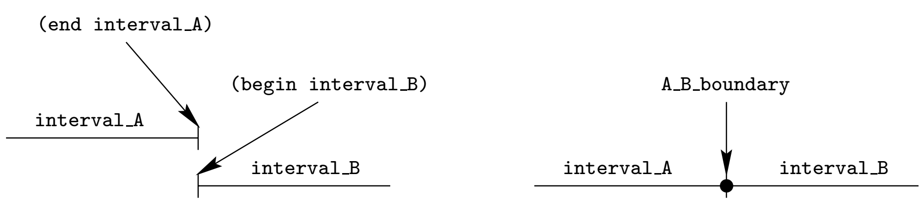
This was a macroexpression which would expand to the relation

```
((end interval_A) (begin interval_B) 0 0)
```

as temporal relations were described in section 2.1.1[1]. In the new implementation, this consecutive-phase relation would be expressed as follows, in the object-oriented syntax of Java:

```
TemplatePoint A_B_boundary = new TemplatePoint();
interval_A.setEnd(A_B_boundary);
interval_B.setBegin(A_B_boundary);
```

As can be seen, `interval_A` and `interval_B` share the new `TemplatePoint`, `A_B_boundary`, as their boundary point. This simple addition of one new `TemplatePoint` was sufficient to express the consecutive-phase relation, in which intervals are separated with zero length. A graphical depiction of this difference can be seen in Figure 3-2.



Original description. Revised description.

Figure 3-2: Adjacent temporal intervals modification.

Related non-adjacent intervals

The conversion of descriptions of intervals that are related but are not separated by zero length from the original to the new syntax of TrenDx is slightly more complicated. Relations may exist between intervals of the following forms:

```
((end interval_A) (begin interval_B) 4 4)
((end interval_B) (begin interval_C) 2 7)
```

[1]The relation created by consecutive-phase would actually create upper and lower bounds of length `*epsilon*` (instead of `0`) between the interval boundaries, but this detail is not important.

In these situations, the boundaries of the intervals are separated with non-zero and/or uncertain temporal distances. In the new implementation of TrenDx, these relationships require the creation of two new `TemplatePoint`s as well as an additional `TemplateInterval` (temporal interval). These are constructed as follows:

```
TemplatePoint end_A = new TemplatePoint();
TemplatePoint begin_B = new TemplatePoint();
TemplateInterval A_to_B = new TemplateInterval();
interval_A.setEnd(end_A);
A_to_B.setBegin(end_A);
A_to_B.setEnd(begin_B);
interval_B.setBegin(begin_B);
```

The new `TemplatePoint`s are the boundary points for the original intervals, and the new interval is sandwiched between them. The temporal distance between these boundary points can then be set by adjusting the length of the new interval, which is equivalent to specifying relative distance between the boundary points themselves. The new interval that was created does not contain any value constraints. This behavior is the same as that which was modeled in the relation between the original two intervals with the old TrenDx template description language. The new interval acts purely as a temporal restriction. This modification is displayed pictorially in Figure 3-3.

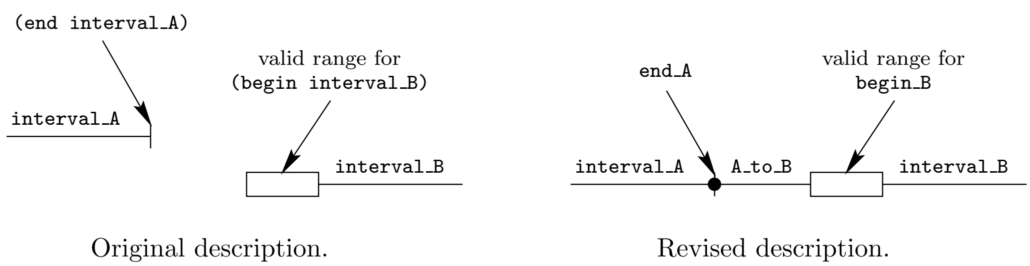


Figure 3-3: Related non-adjacent temporal intervals modification.

As discussed previously, the requirement to add these new template points explicitly does not result in any loss of information from the original TrenDx templates. These changes are syntactic, and the translations from the older descriptions to the new syntax is well-specified. Explicitly declaring these points does aid in the visualization of the trend templates, as there are no longer any “gaps” in the progression of temporal intervals in a template (no data points are able to fall in the holes between intervals, as there is always a temporal interval to contain them). The explicit declaration of these boundary points is useful for the real-time scoring algorithm of the revised TrenDx program, which will be discussed in chapter 4.

3.2.2 Interval chains

The other difference in the trend template models between the original version of TrenDx and the updated program comes in the form of interval chains. In the new trend template model, hypotheses are composed of distinct, independent interval chains. An interval chain is composed of inter-dependent temporal units that are related to one another. In this context, temporal points are related if they have a direct relation to one another or if they are both related to another temporal point. That is, related is a both a transitive and commutative property. If `point_A` is related to `point_B`, and `point_B` is related to `point_C`, then `point_C` is related to `point_A`. Since temporal intervals are defined by their boundary points, and the two boundary points of a temporal interval are inherently related, only template points need be discussed to define interval chains.

Interval chains can be discovered from the original trend template descriptions as follows. Initially, each template point is in its own set. For each relation encountered, join (via set union) the sets that contain those two points. Once this has been done for every relation in the trend template description, the non-intersecting sets will form distinct interval chains. In essence, interval chains are sets of temporal points whose location depend on the locations of other points in the chain. Points from distinct interval chains do not depend on the locations of each other, and therefore distinct interval chains are independent. Due to this independence, each interval chain may be processed independently of the other interval chains in a hypothesis. The hypothesis score is an accumulation of the scores of the independent interval chains. Distinct interval chains are useful for expressing trends over several process data streams whose temporally significant breakpoints are unrelated. In the original implementation of TrenDx, these types of trends were not considered. Distinct interval chains are an extension to the TrenDx trend template description language.

Chapter 4

Monitoring Paradigm

To satisfy the goals of this thesis, the TrenDx monitoring procedure needed to be reformulated. The original TrenDx monitoring program was not targeted at being a real-time monitor system, which allowed it to do excessive computations without much concern about practical applicability. As discussed in the previous chapter, a few modifications were made to the computational models of TrenDx. These modifications help to make TrenDx suitable for real-time processing. This chapter discusses how these changes are used to perform hypothesis scoring, which is the core of the TrenDx monitoring system.

4.1 Error value calculations

The hypothesis scoring procedure of the original TrenDx involved calculating a weighted average of value constraint scores for each temporal interval in the hypothesis. The revised version of TrenDx also computes an error score for each value constraint in the hypothesis, albeit in a different fashion, and uses a weighted average of these scores to form a score for the hypothesis. Additionally, however, the new version of TrenDx computes error scores for each `TemplatePoint`, and adds these scores into the total score for the hypothesis. This feature was not present in the original implementation of TrenDx.

4.1.1 Temporal constraint error

As mentioned above, each `TemplatePoint` is given an error score. `TemplatePoint`s contain all of the temporal constraint information of a trend template, so these error scores represent how well a model matches the process data in terms of the temporal locations of its significant behavioral changing points. The original implementation of TrenDx used temporal constraints in a threshold-based fashion to prune possibilities to explore in terms of the intervals to which data points may be assigned. One problem with this approach is that points just outside of the interval boundaries are not considered as possibly being assigned to the interval. This is often not the behavior desired by the person designing the template. Instead, they would prefer that points toward the center of the interval be considered very likely candidates for being in that interval, and the possibility of a point being assigned to an interval decreases as the temporal distance from the center of the interval increases, in a continuous fashion. This problem is analogous to the value constraint whose intent is to describe a "desired range of values" as discussed in section 2.2.1.

The way in which the new version of TrenDx determines the temporal locations of interval boundary points also required revision due to the new model of input process data. The original version of TrenDx modeled the input as a sequence of discrete data points. The new data model describes the input to the monitor as a linearly segmented continuous data stream. A continuous data stream translates to an infinite number of discrete points to be handled by the original TrenDx monitor, which creates an intractable problem. If the anchor points of the linear segments were instead considered in the same fashion as discrete points in the original monitoring scheme, problems would arise when value constraint scores were to be calculated. If the original value constraint scoring method were used, the information in the process data between anchor points would be lost.

The process of determining where the `TemplatePoint`s are assigned temporally in the data stream is described in section 4.3. Both temporal and value constraint error scores are used. It is easier to describe the error given to a `TemplatePoint` if it is assumed that the location of the `TemplatePoint` in the data stream has already been determined. The error calculation used in the new TrenDx program starts with an expected temporal location of the `TemplatePoint`. This expected value is derived from the trend template. Every interval chain contains some fixed `TemplatePoint` (such as "`birth`" or `now`). The expected temporal location for a point with a direct relation to one of these points is the midpoint of the specified range of the point in the trend template model. `TemplatePoint`s that have relations to other non-fixed points are assigned an expected value based on the determined or expected value of the related point, in that order. For instance, given the relation

```
(begin_A end_A 5 7)
```

the expected temporal location of `end_A` would be 6 time units after the determined location of `begin_A`, but if `begin_A` were not yet determined, `end_A`'s expected location would be 6 time units after the expected temporal location of `end_A`.

Given the expected temporal location of the `TemplatePoint` along with its determined temporal location, the error calculation is straightforward. The error is simply the squared difference between the expected value and the actual value, normalized so that the borders of the specified range receive an error of 1 (and the center of the range receives an error of 0).

$$\text{temporal error} = \left(\frac{2 \cdot (\textsc{DeterminedTime} - \textsc{ExpectedTime})}{\textsc{UpperBound} - \textsc{LowerBound}}\right)^2$$

In general, $\textsc{ExpectedTime} = \frac{1}{2}(\textsc{UpperBound} + \textsc{LowerBound})$. This error function has the desired characteristics of increasing error as temporal distance from the center of the range increases, and it continues to grow relatively fast outside of the desired range, which will cause a large error score for temporally significant points in the process data that do not match the locations of those in the hypothesis well. A graph of this error function is displayed in Figure 4-1.

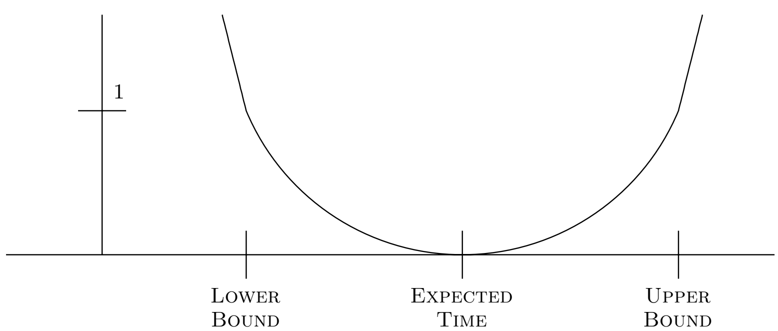


Figure 4-1: Temporal error function.

4.1.2 Value constraint error

Due to the new process data input model, the way in which TrenDx computes the error of each value constraint required revision. In the original program, value constraint errors were determined with either the residual mean square error or the mean absolute percentage error calculation. These calculations use a sum over each data point contained in the temporal interval. Since the data model is no longer composed of discrete points, but instead of continuous linear segments, a different error measure must be used. The error measure used for value constraints in the revised TrenDx monitoring algorithm is a measure of the area between the process data and the trend template. This is conceptually analogous to the error measure used in the original discrete-point program, but expanded to handle continuous input.

When a value constraint is fully specified by the trend template, the error calculation is straightforward. The area between the value constraint and the process data is easily calculated, since the process data are presented to the monitoring algorithm as a sequence of continuous linear segments. In general, the area between the portion of a linear segment within the interval and the value constraint would be computed with an integration. The trend templates used in the evaluation of the revised version of TrenDx contained only linear or constant value constraints, which simplifies this calculation. With constraints of this type, the area between a segment and the constraint is either in the shape of a trapezoid, if the segment and the constraint do not intersect, or two triangles, if they do intersect. The non-intersecting area calculation looks like the following:

$$\begin{aligned}
\text{non-intersecting area} &= \text{trapezoidal area} \\
&= \frac{1}{2} \cdot \textsc{Height} \cdot (\textsc{Base}_{\mathrm{L}} + \textsc{Base}_{\mathrm{R}}) \\
&= \frac{1}{2} \cdot (t_{\mathrm{R}} - t_{\mathrm{L}}) \cdot (|x_{\mathrm{L}} - \hat{x}_{\mathrm{L}}| + |x_{\mathrm{R}} - \hat{x}_{\mathrm{R}}|)
\end{aligned}$$

The subscript L indicates the left side of the section, and R indicates the right side, with the understanding that time increases toward the right. t indicates the time value, x indicates the value of the data, and $\hat{x}$ indicates the value expected by the trend template. The area calculation for the case in which the constraint and the data intersect proceeds as follows:

$$\begin{aligned}
\text{intersecting area} &= \text{left triangular area} + \text{right triangular area} \\
&= \frac{1}{2} \cdot \textsc{Base}_{\mathrm{L}} \cdot \textsc{Height}_{\mathrm{L}} + \frac{1}{2} \cdot \textsc{Base}_{\mathrm{R}} \cdot \textsc{Height}_{\mathrm{R}} \\
&= \frac{1}{2} \cdot (t_{\mathrm{I}} - t_{\mathrm{L}}) \cdot |x_{\mathrm{L}} - \hat{x}_{\mathrm{L}}| + \frac{1}{2} \cdot (t_{\mathrm{R}} - t_{\mathrm{I}}) \cdot |x_{\mathrm{R}} - \hat{x}_{\mathrm{R}}|
\end{aligned}$$

Here, t_{I} is the temporal value of the intersection of the trend template and the data segment. These area calculations are pictured in Figure 4-2.

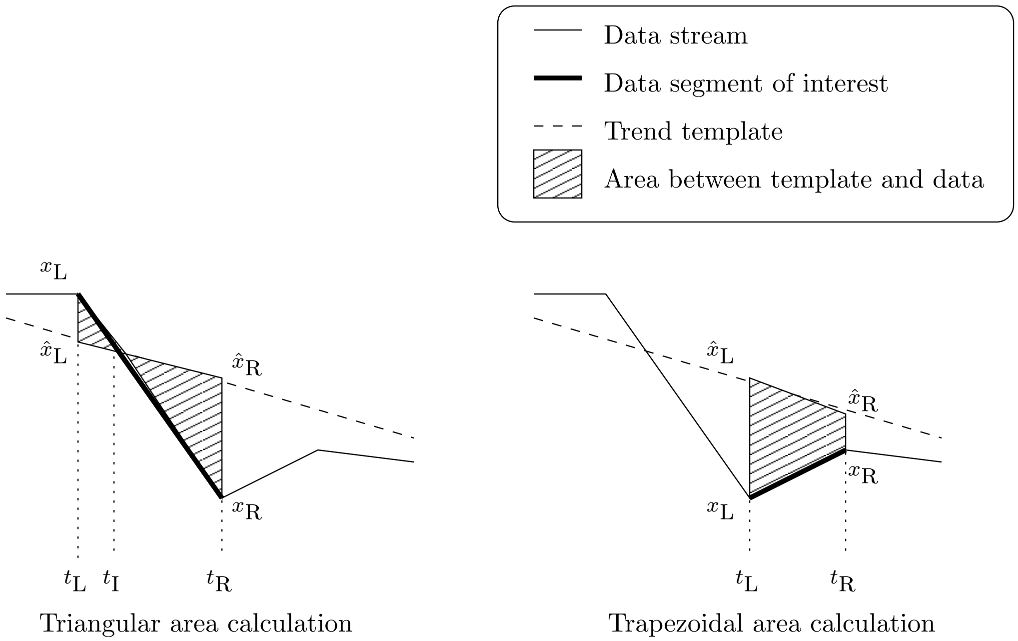


Figure 4-2: Trend template vs. data segment areas.

In each of these equations, the only variables that are not obvious from the information in the data stream and the trend template may be the expected values of the constraint at the endpoints of the section ($\hat{x}_L$ and $\hat{x}_R$) and the intersection point, t_I. When the value constraint is a specified constant, these variables take on the value of that constant, and t_I is at the point in the data stream where the segment achieves this constant value, if any. If the constraint is an unspecified constant, the value for the constraint to take on is first determined by finding the average value of the process data over the temporal interval, and using that as the value of the constant constraint. For linear constraints with an unspecified slope, the constraint is made to be the line segment constructed between the data values at the beginning and end of the temporal interval. For linear constraints with a specified slope, the constraint is positioned so that it bisects this line segment. These constraint determinations are depicted graphically in Figure 4-3.

The method of determining the free parameters of value constraints that are not fully specified does not minimize the area between the constraint and the data stream over all possible values of the free parameters, particularly for unspecified linear constraints. However, the desired behavior of these area calculations is that the area tends to decrease as the interval boundaries approach the corresponding points of temporal significance in the actual data stream. This behavior is desired for the template point determination algorithm described in section 4.3. Furthermore, the exact values of the area calculations are not as important as are the comparisons of scores produced by competing trend templates. In general, those value constraints that more closely match the pattern of the data stream will produce smaller areas using these methods, resulting in hypotheses that score better than others when their trend templates more accurately describe the input process data.

Although these triangular and trapezoidal area calculations are only valid for linear value constraints, it is advantageous to recall that the majority of functions that are used in trend modeling are easily broken into linear segments, which is part of the rationalization for the new TrenDx data input model. So although a true integration would be ideal in order to describe a trend with any function, these calculations should suffice for most circumstances. Under the new TrenDx framework, however, the program could be easily extend to perform more complex area calculations.

As was the case in the original TrenDx, problems may arise when values derived from parameters of varying magnitudes are used in an accumulation. In order to normalize the percent of error that each value constraint contributes to the total error score for a temporal interval, an idea similar to the mean absolute percentage error is used. The error score is scaled by the average value of the process data in the interval for the parameter associated with the constraint. However, problems similar to those encountered when using the mean absolute percentage error in the original version of TrenDx arise when the average value is close to zero, so the new program allows use of the unscaled error as well. Furthermore, each value constraint may be assigned a weighting factor that specifies its relative importance in relation to other constraints in the hypothesis.

A full description of how TrenDx assigns a score to a hypothesis, using both temporal error scores and value constraint error scores, appears later in this chapter.

4.2 Real-time processing

The primary goal of this thesis is to develop a real-time monitoring system using the templates of TrenDx. The previous section discussed how error scores are produced for temporal and value constraints. The next section describes how the temporal location of a template point is determined in the data stream. This section describes how the input data travel through the monitoring system to produce real-time results.

Recall that the input process data are modeled as a continuous data stream. Furthermore, this stream is broken into continuous linear segments before it is presented to the monitoring algorithm. The segmentation algorithm described in section 3.1.2 can encapsulate the information needed to reconstruct this sequence of segments via the anchor points it determines. In the new TrenDx program, these anchor points are called `Pivots`. Therefore, the input process data are viewed in the program as a pivot stream. When the monitor receives a new pivot from the segmenting algorithm, it notifies each hypothesis. According to the new model in which hypotheses contain independent interval chains, the hypothesis propagates the notification to each of the interval chains it contains. The objects to which this information is most relevant are the template points themselves. Once the template points are assigned a temporal location in the data stream, the interval chain and therefore the hypothesis can produce a final score. So the interval chain must notify each template point in the chain of the newly encountered pivot.

The only template points that need to receive updates of new pivots are those template points whose temporal location has not been determined[1]. When the data stream information reaches an undetermined template point, the monitor decides whether that template point is able to be determined. This decision is based on the time stamp of the incoming pivot along with the temporal constraints of the template point. The current method used makes a decision that a template point is ready to be determined if the time stamp of the pivot exceeds the upper bound of the temporal constraint range on the template point. The template point determination process is described in the next section.

One of the problems with the original implementation of TrenDx was that it was not able to accumulate its results into a summary that could be used for future computation. In the revised program, determined template points contain the necessary information for further processing. Once a template point is determined, its error score will not change, and so its score can be memoized for fast look-up. In addition to the temporal error score of the template point itself, the error scores of each of the value constraints in temporal intervals which are bounded on the right by the template point will no longer change (template points may not be determined to be beyond the most recently encountered pivot). So template

[1]Template points whose positions are specified relative to "now" are never really determined, as their positions are recalculated with each new pivot (so these template points also require notification of new pivots). Template points whose position is fixed to positive infinity have similar requirements, as the error scores of the intervals they bound need to be updated.

points can contain all of the necessary error information for intervals in the past. Using these properties, an interval chain score is produced by summing the error scores of the determined template points (this value can, itself, be memoized) as well as the error scores associated with undetermined template points. Scores for undetermined template points are produced using the expected temporal location of the template point, as described in section 4.1.1, and as much of the process data as is available at the time of computation. The error scores of undetermined template points and of value constraints in intervals bounded by these points require updating every time additional pivots are received.

The amount of time required by the monitor to process each pivot from the data stream is represented by the equation

$$\begin{aligned} \text{pivot time} &= (\text{previously determined template point time}) \\ &\quad +(\text{newly determined template point time}) \\ &\quad +(\text{undetermined template point time}) \end{aligned}$$

The time used on previously determined template points is a constant, based on the number of interval chains and hypotheses in the monitor set, because of the memoization of error values discussed above. The time used on newly determined template points is equal to the time required to determine a template point multiplied by the number of template points that are able to be determined due to that pivot. The time required to determine a template point is discussed in the next section. The number of points able to be determined is dependent on the characteristics of the trend templates and the process data, but should generally be bounded by a small constant. The time used on undetermined template points is dependent on the number of value constraints bounded by these template points. This is also a property of the monitor set and process data, which should again be bounded by a constant.

4.3 Template point determination

One of the most important parts of the revised version of TrenDx is the methodology it uses to determine the temporal locations of template points in the process data input. In the original version of TrenDx, the hypothesis score was determined by finding the best score among an easily determined finite set of possible interval boundary locations. The set of possibilities was finite because the only distinction that had an impact in the error score was to which side of an interval boundary the discrete data points were assigned. In the new version of TrenDx, the continuous data model results in an infinite number of possible locations for each template point. The locations of these template points has a large impact on the scores given to hypotheses in any scoring system that does not discard information from the input stream. Because of this impact, the template point determination method should not be blind to the hypothesis scoring algorithm. Rather, its placement of each template point should strive to produce the best possible score for the hypothesis. These considerations contributed largely to the method of error score determination described in section 4.1. This section describes the method by which the locations of template points are determined.

4.3.1 Simulated annealing

The algorithm used for template point determination is a form of the simulated annealing approach to global optimization. Simulated annealing is based on the way in which a heated substance (particularly a metal) cools slowly to form a crystalline structure. Each individual molecule in the substance strives to achieve its lowest possible energy state. However, based on the total energy of all of the molecules in the substance (i.e., the temperature), there is a possibility that a molecule may be forced into a higher energy state. As time progresses, the substance loses energy, the temperature decreases, and the molecules become less and less likely to move from the locally minimum energy states that each has found. This process continues until the temperature has reached "freezing", at which point the substance is locked into its current state and molecules are no longer able to move. When cooled slowly enough, this process is guaranteed to result in an optimum placement of molecules to achieve to lowest possible energy state.

The implementation of this method proceeds as follows. An initial energy is assigned to the initially proposed solution to the problem, and the process is given an initial temperature. The solution is given a random perturbation, and the energy of the resultant state of the solution is calculated. If the perturbation resulted in a lower energy state, the change is accepted. If the perturbation resulted in a higher energy state, the change is accepted with probability $p = \exp \frac{\Delta E}{T}$, where ΔE is the change in energy, and T is the current temperature. The proposed change is discarded and the system is returned to its previous state with probability $(1 - p)$. The temperature is then lowered, and the process continues. When the temperature reaches a specified minimum, the process stops, and the current state is returned as the solution.

4.3.2 Application to template point determination

To apply this method to determining the temporal locations of template points, a mapping must be found for the parameters in the simulated annealing implementation. The revised TrenDx program uses the error between the trend template and the process data as the energy of the current state. This error is calculated as described in section 4.1, assuming that the temporal location of the template point is the one currently being proposed. The random perturbation in the current solution is mapped to a change in the temporal location of the template point that is being determined. This change is randomly chosen from a uniform distribution between -1.0 and +1.0. The proposed solution is the location of this template point. In general, there is no natural mapping of the temperature parameter to solving optimization problems. The manner in which temperature is handled contributes to the effectiveness of a solution and the time required to achieve the solution in such problems. In the revised TrenDx implementation, the temperature is initially set at 1.0, and it is decreased logarithmically by a factor of 0.9 at each iteration, i.e., $T_{i+1} = 0.9 \cdot T_i$. The algorithm stops when the temperature reaches 0.1, at which point the lowest energy state encountered thus far is returned as the solution to the problem, and the corresponding position of the template is fixed from that point onwards. A demonstration of the template point determination algorithm is shown in Figure 4-4.

This approach to determining the location of template points is directly related to the hypothesis scoring method, in that they both use the same error calculations. This process loops for a fixed number of iterations, determined by the initial temperature, the freezing temperature, and the temperature reduction factor. During each iteration, the number of calculations required is determined by the number of value constraints in intervals affected by the move in the location of the template point in conjunction with the number of segments of the process data stream assigned to these intervals. In general, these factors should be bounded by a small constant. Therefore, the total time required to determine each template point will be bounded by a constant using this method. Each template point requires determination exactly once, after which the error values affected by its location are memoized for fast hypothesis scoring.

There are several advantages to using this technique to locating the positions of temporal points in the data stream. Due to its constant-bounded running time, the template point determination can be performed in real time. In contrast to many other search techniques, the simulated annealing method does not become trapped in search paths that only lead to local extrema. This is a very important property of any searching method to be used in determining the locations of template points. Finally, the template points may be positioned in between the pivots of the data stream with this determination method. The combination of temporal and value constraint errors used to produce a hypothesis score make it possible for the best score for a hypothesis to be achieved when the template points are located anywhere along a linear segment of the data stream as opposed to being restricted to the pivot points. This is an important ability for the determination mechanism to have, particularly when data stream pivots are spread relatively far apart in relation to temporal constraints on the template points. Additionally, this ability partially compensates for a detail of the segmentation algorithm described in section 3.1.2. The segmentation algorithm only allows pivots to be placed temporally at locations where input data points occur, when in fact a more accurate segmentation scheme might choose to put a pivot at any temporal location. This fact becomes more important as the discrete data values are spread further apart in time.

This completes the description of the revised TrenDx monitoring system. Hypotheses are scored by combining the scores of their component interval chains, and interval chains scores are determined through the scores of their component template points. After hypotheses are scored, there are many ways to interpret the meanings of the scores. The revised version of TrenDx follows the same model that the previous version of TrenDx used. A decision to trigger a signal that a process is in an abnormal state is based on the lowest score of the applicable trend templates of normal process behavior. If a threshold score is surpassed, the signal is triggered, indicating to the user that the process requires special attention. In the following chapter, this monitoring scheme is analyzed in the domain of children's growth.

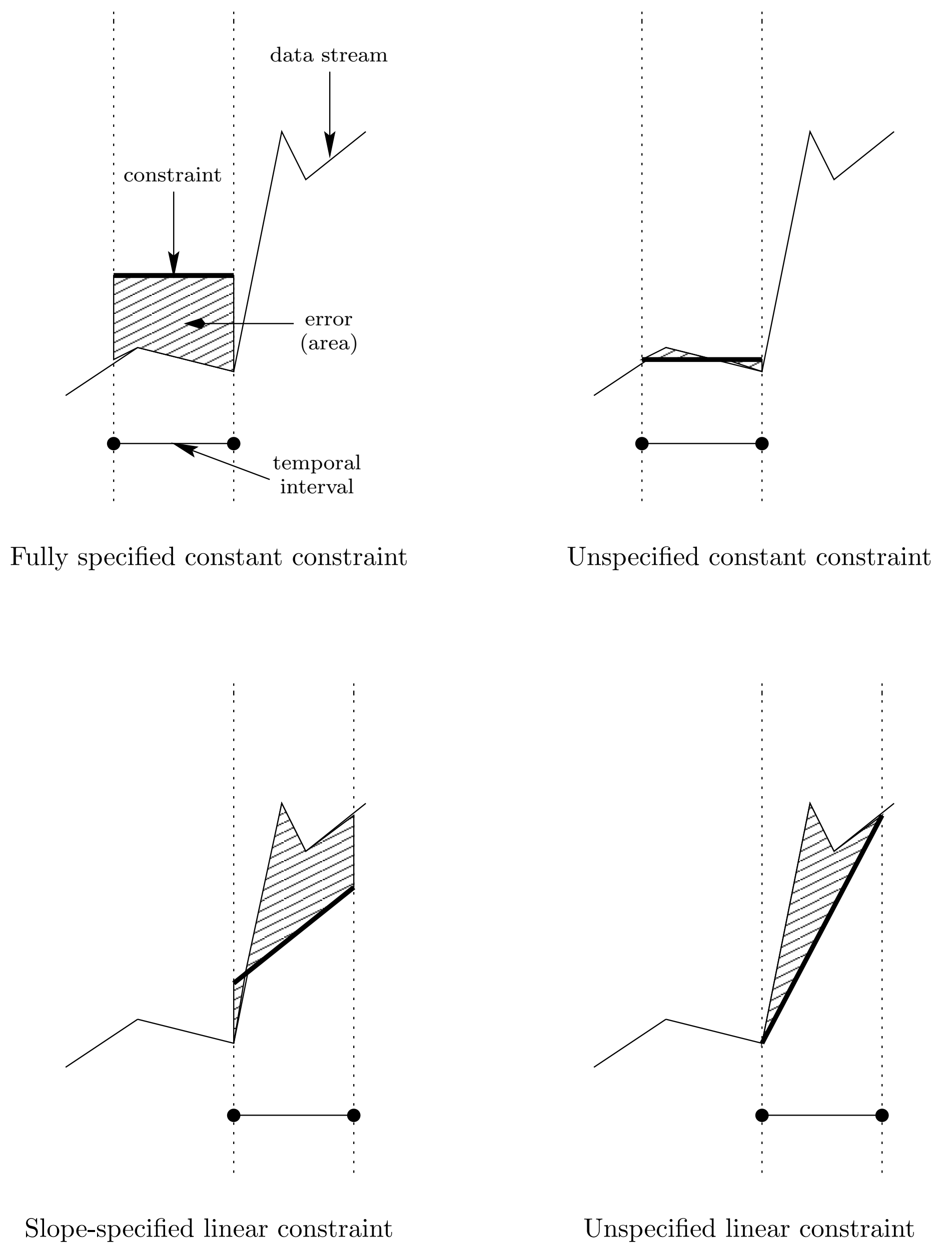


Figure 4-3: Value constraints against a data stream.

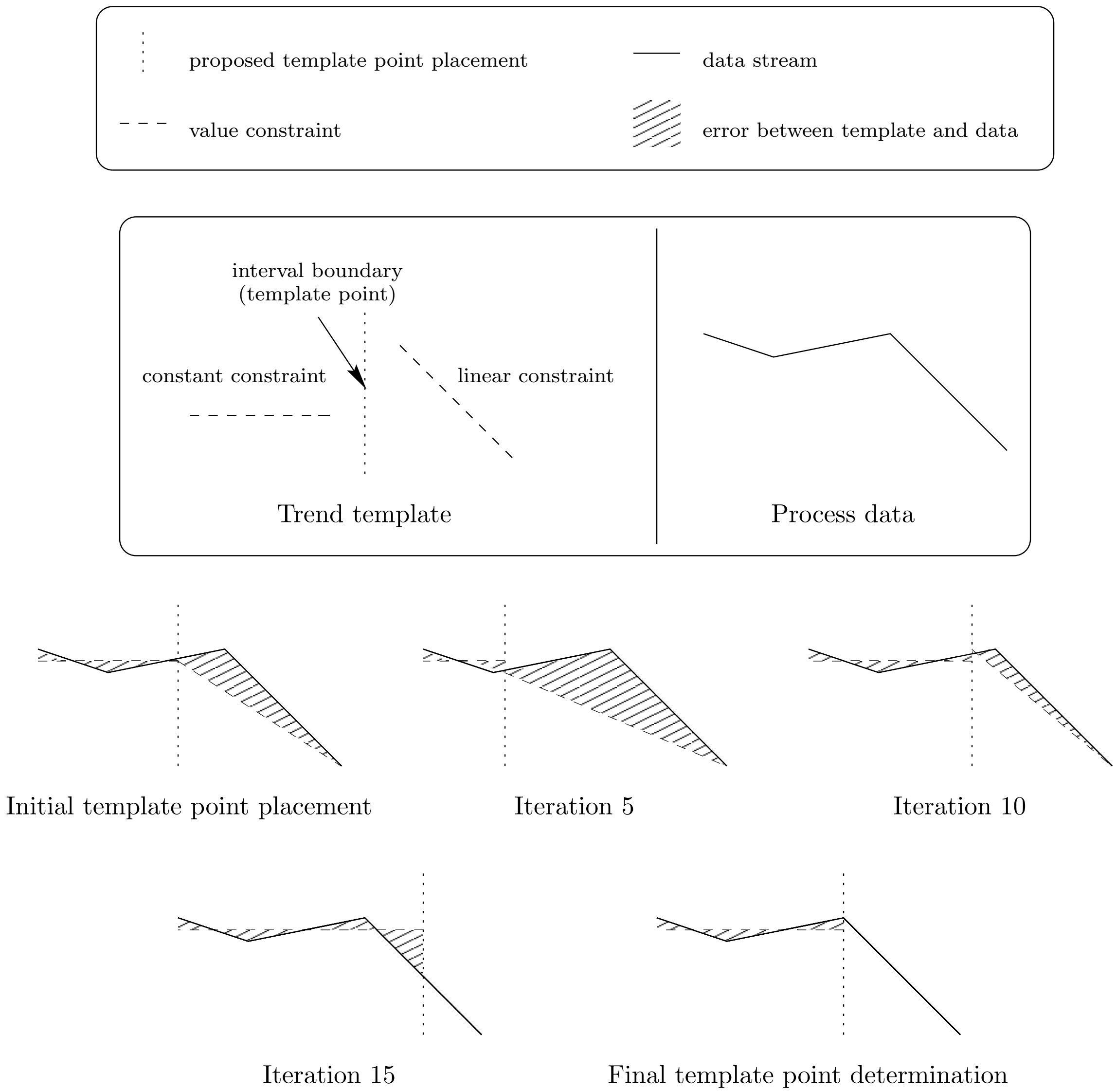


Figure 4-4: Demonstration of template point determination.

Chapter 5

Pediatric Growth

The revised version of TrenDx described in this thesis was compared to previous versions of TrenDx to determine its feasibility for use as a process monitor in the same domains as the original TrenDx. This was necessary because the process data input model was significantly modified in the revised version of TrenDx, as well as its hypothesis scoring mechanisms. The methods and results of these comparisons are discussed in this section.

The previous versions of TrenDx were most thoroughly tested in the domain of pediatric growth. There are many reasons why developing a monitor for this domain is important. Several growth abnormalities can be detected in this domain by observing a small number of parameters, particularly height and weight measurements. However, a correct diagnosis based on these measurements is often quite difficult for general pediatricians. The decision by a pediatrician to refer a child to a growth specialist is generally based on relatively few height and weight data points. Such a referral would be called for only when the child is suspected of abnormal growth behavior. Unfortunately, there are many types of growth patterns that are considered to be normal growth. These various patterns result from the differences in age at which children enter puberty, family history, and other influences.

The most common way to detect growth abnormalities is through plotting the height and weight measurements on a growth chart, on which several centiles for these statistics have been drawn based on data collected by the National Center for Health Statistics (NCHS). Variations of this chart are available for males and females in different subpopulations. Other than being able to detect when the growth pattern is far from the standard development rate, doctors are able to notice when a child's measurements are moving to different centile channels using these charts, which may indicate abnormal growth. Unfortunately, doctors do not have enough time to carefully examine each patient's records, and must make judgement calls based on quick reviews of data. It is often difficult to differentiate between normal and abnormal growth patterns even using these charts, as the centile channels on the charts are modeled by complex functions, there are several charts to consider for each patient, and children mature at different rates. Pediatricians must be careful not to refer too many children without growth disorders to specialists, as the specialists would then become overwhelmed and would not be able to treat each child as well as possible. On the other hand, those children who do have growth disorders should be referred to a specialist as soon as possible, because early detection and treatment produces the best results in the long run.

Experts in the domain of children's growth disorders are significantly more adept at achieving the correct clinical diagnoses with height and weight information, but their skills are in too great demand to be asked to review every child's measurements. With additional information, such as measurements of bone age and sexual development, their diagnoses become much more accurate. It would be a great asset to children's health care if this expert knowledge could be programmed into a monitor system that could be used at a general pediatrician's office to assist in referral decisions.

5.1 Trend Templates

One aspect necessary to the proper function of TrenDx is the programming of competing trend templates. These trend templates must be developed by a knowledge engineer in such a way that they are able to distinguish between different diagnoses of the process being monitored. For the evaluations of TrenDx in the pediatric growth domain, trend templates were defined in three categories: normal templates, abnormal templates, and supplemental templates. The normal templates were originally designed by Haimowitz [3], and this template set was refined and expanded by Le [5].

5.1.1 Normal templates

The majority of the templates used in this domain followed a temporal interval model similar to that depicted in Figure 5-1. There are three hypotheses that indicate normal growth behavior: average growth, constitutional delay, and early puberty.

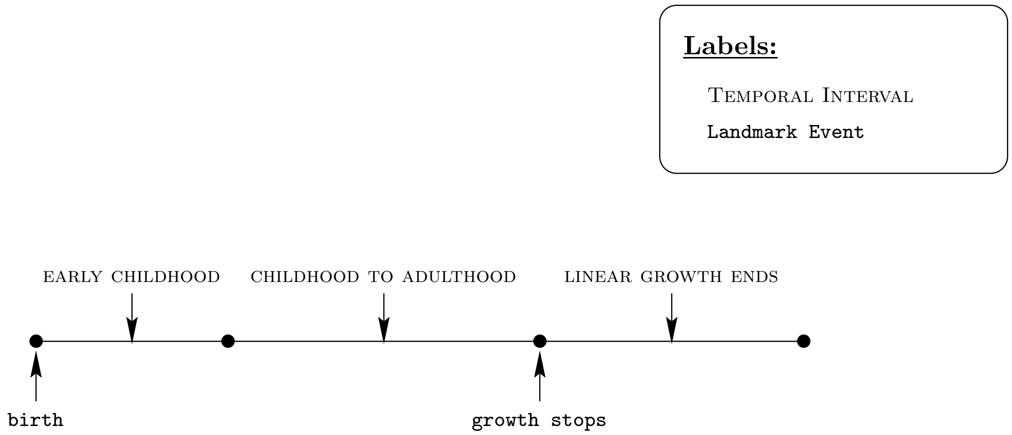


Figure 5-1: Temporal interval model for growth trend templates. Interval and landmark event names as in Le [5].

Average Growth is the trend template intended to match most closely to the growth patterns of a child following the mean behavior of the NCHS charts. This should be the lowest-scoring template for the majority of children seen by a pediatrician.

Constitutional Delay is a common condition in which pubertal onset occurs later than in the average growth scenario, and therefore the end of the growing stage occurs at a later age. Adult height and weight are equivalent to those achieved by average growth.

Early Puberty is the hypothesis which depicts a child whose pubertal onset occurs before that in average growth. Consequently, growth stops at an earlier age as well, but adult height and weight achieved are equivalent to those in the average growth model.

None of the previous three templates are harmful growth conditions, and therefore do not require referral to a specialist.

5.1.2 Abnormal templates

There were four abnormal trend templates defined in the growth monitor: congenital growth hormone deficiency, short bone syndrome, acquired growth hormone deficiency, and precocious puberty. Each of these diagnoses is a disorder that requires treatment by a growth specialist.

Congenital Growth Hormone Deficiency is a condition in which a child has an inability to produce or respond to growth hormone. The child's skeletal development is greatly delayed, much more so than in the constitutional delay model, and therefore the child is significantly short for his/her age.

Short Bone Syndrome models a growth disorder in which children are even shorter than those suffering from growth hormone deficiency, but their skeletal age is not significantly delayed.

Acquired Growth Hormone Deficiency is marked by normal growth followed by a significant growth deceleration, which occurs after the onset of this condition. The temporal intervals of this trend template are described by a model that differs from the rest of the normal and abnormal hypotheses, and is depicted in Figure 5-2.

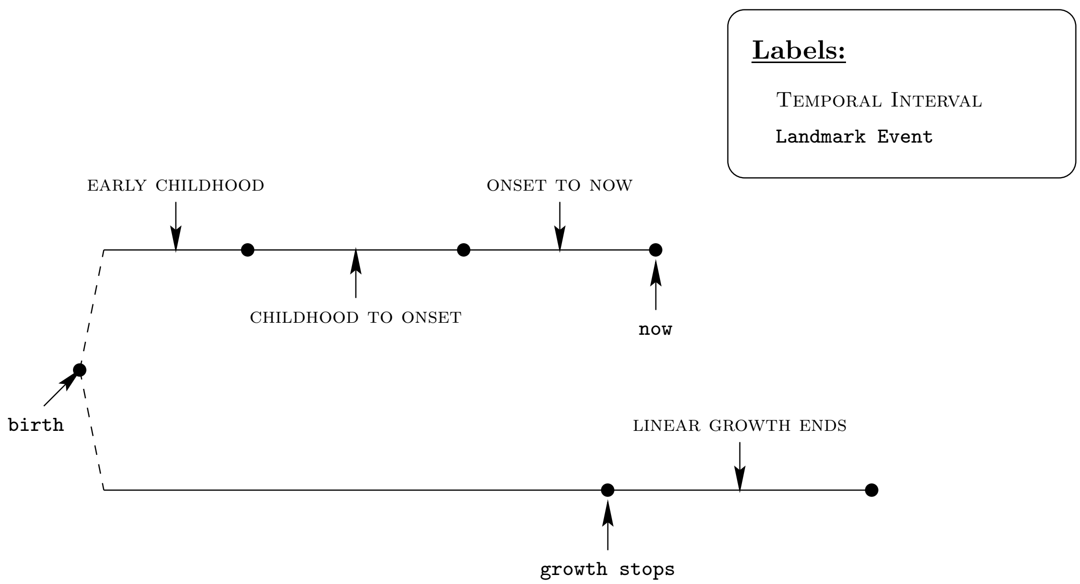


Figure 5-2: Temporal interval model for Acquired Growth Hormone Deficiency. Interval and landmark event names as in Le [5].

Precocious Puberty describes an extremely early development of a child. The child is generally significantly taller than his/her peers due to advanced skeletal development.

5.1.3 Supplemental templates

In addition to the diagnoses presented above, information about the build of a child is useful to pediatricians and growth specialists. These trend templates are fairly simple, with a temporal interval model of that displayed in Figure 5-3.

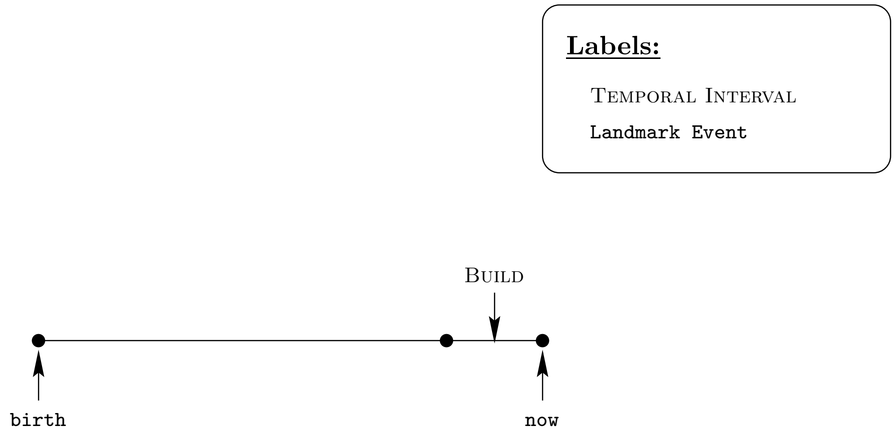


Figure 5-3: Temporal interval model for build templates.

Three trend templates use this model: normal build, obese, and malnourished. The lowest scoring of these templates may be used as additional information to support particular diagnoses.

5.2 Evaluation methods

The evaluations of previous versions of TrenDx were largely based on patient data collected by Le [5]. The patient records were collected from the Endocrine Division at the Boston Children's Hospital. In total, 95 patient records were used for the evaluations.

5.2.1 Expert Gold Standard

A gold standard used in the evaluations of TrenDx was the diagnosis of a pediatric endocrinologist for each of the patients in the study. The diagnosis was based on the height and weight data of the patient that were available prior to that patient's visit to the clinic. These height and weight data were presented on NCHS growth charts for ease of visualization. The growth specialist was asked to either recommend or deny a referral to the growth clinic for each patient. The referral decision of the pediatric endocrinologist would be considered the correct course of action for a general pediatrician to take. Of the 95 patient records used in the evaluations, the pediatric endocrinologist decisions were to refer 59 patients to a growth clinic, and to deny referrals for 36 of the patients based on the information available.

5.2.2 Physician base line

Le conducted a study in which human participants were asked to diagnose the patients based on the same information that was provided to the pediatric endocrinologists. The participants (henceforth "physicians") were required to have medical training (one registered nurse participated, along with students in medical school as well as post-residency physicians). They were asked to determine, based on the height and weight data presented on an NCHS growth chart, whether or not a patient should be referred to a growth clinic. The accuracy of their responses, compared to the expert gold standard of the pediatric endocrinologists, formed the base line to which the results of TrenDx were compared.

5.2.3 TrenDx diagnoses

To compare the results of TrenDx to the human diagnoses, the height and weight data points were entered into the program for each patient in the study. TrenDx was used to monitor these points, and the best score for each hypothesis in the monitor set was recorded. To make a referral decision in previous versions of TrenDx, the score of the lowest-scoring (closest matching) normal growth template (see section 5.1.1) was compared against a threshold value. If this score was above the threshold value the referral would be made, otherwise it would be denied. To perform a similar evaluation with the version of TrenDx described in this thesis, the hypothesis scores needed to be modified before a threshold comparison could be used. This is due to the fact that as the length of time over which data points are spread increases, the area (which is used in the error score) between the data and the model generally increases. To compensate for this fact, the score for each hypothesis was scaled by the length of time over which the data points were collected. This scaled value was then compared against a threshold to determine the referral decision. In this evaluation, a threshold value of 0.663 was used.

5.3 Evaluation results

The revised version of TrenDx was evaluated in the same manner and using the same test cases as previous implementations were evaluated. The results of the new evaluation are compared to the results of the previous evaluations in this section. It is important to compare the reformulated version of TrenDx to previous versions of the monitor to ensure that the revisions did not cripple its diagnostic ability. At the time of this evaluation, the development of real-time TrenDx is at a similar stage as was the version of TrenDx used by Le [5].

5.3.1 Evaluation metrics

There were two metrics used in the previous evaluations of TrenDx to determine its ability to diagnose patients in the growth domain correctly. The metrics used were sensitivity and specificity. These metrics are typically used to describe the accuracy and effectiveness of a diagnostic test. They are defined in terms of "positive" and "negative" test results. In the context of this evaluation, positive is to mean that the decision of the test was to refer the patient to a growth clinic, while negative means that the decision made was not to refer the patient.

A summary of the results of the evaluations of all of the versions of TrenDx with results published using these metrics is presented in Table 5.1. This table also includes the data of the expert gold standard and the physician base line provided by Le.

Sensitivity

The sensitivity of a testing mechanism is a measure of how accurate the test is in producing positive results. It is the ratio of the number of test subjects that both should and do test positive to the total number of test subjects that should test positive. In terms of the test used in this thesis,

$$\text{sensitivity} = \frac{\text{number of patients referred by both gold standard and test method}}{\text{number of patients referred by gold standard}}$$

In the gathering of opinions from medically trained physicians conducted by Le, the number of responses collected varied greatly from patient to patient. This was largely due to the methods used to conduct the study. A total of 217 diagnoses were collected from the human participants. Each diagnosis applied to one of the 95 patient test cases used in Le's study, and each physician responded with up to 10 diagnoses for different patients. 142 of these were diagnoses of patients who should have been referred to a growth clinic, according to the pediatric endocrinologist gold standard. There were 91 instances in which the opinion of the physician correctly resulted in a referral according to the gold standard. Consequently, 51 diagnoses by physicians disagreed with the expert gold standard and denied a referral that should have been given.

$$\text{Physician sensitivity} = \frac{91}{142} = 0.64$$

Le was also the first to evaluate a version of TrenDx in this fashion. His version of TrenDx, henceforth referred to as $\text{TrenDx}_{\text{LE}}$, correctly referred 36 of the 59 patients who were referred by the pediatric endocrinologist. Likewise, 23 patients were denied referrals by $\text{TrenDx}_{\text{LE}}$ who should have been referred according to the expert gold standard.

$$\text{TrenDx}_{\text{LE}} \text{ sensitivity} = \frac{36}{59} = 0.61$$

Following Le's evaluation, more development was done on TrenDx by DeSouza [2]. This newer version of TrenDx, referred to as $\text{TrenDx}_{\text{DESOUZA}}$, performed better on the patient records collected by Le than did $\text{TrenDx}_{\text{LE}}$, as compared to the expert gold standard. Of the 59 referrals diagnosed by the pediatric endocrinologist, $\text{TrenDx}_{\text{DESOUZA}}$ correctly produced 38 of there referrals, while denying referrals to 21 patients who should have been referred.

$$\text{TrenDx}_{\text{DESOUZA}} \text{ sensitivity} = \frac{38}{59} = 0.64$$

The revised version of TrenDx described in this thesis, $\text{TrenDx}_{\text{BULL}}$, was tested on the same set of 95 patient records gathered by Le. $\text{TrenDx}_{\text{BULL}}$ correctly referred 36 of the 59 patients referred by the expert gold standard, and it mistakenly denied referrals to 23 of those 59 patients. This matches the results of $\text{TrenDx}_{\text{LE}}$.

$$\text{TrenDx}_{\text{BULL}} \text{ sensitivity} = \frac{36}{59} = 0.61$$

Specificity

In a complementary nature to the sensitivity of a testing mechanism, the specificity of a test measures the accuracy of that test in producing negative results. It is the ratio of the number of test subjects that both should not and do not test positive to the total number of test subjects that should not test positive (those that should test negative). In terms of the tests of this thesis,

$$\text{specificity} = \frac{\text{number of patients denied referrals by both gold standard and test method}}{\text{number of patients denied referrals by gold standard}}$$

75 of the responses from the physicians gathered by Le pertained to patients who did not need to be referred to a growth clinic according to the pediatric endocrinologist. 56 responses in this category correctly denied referrals to the patients, while 19 of the physicians diagnoses disagreed with the expert gold standard and suggested referrals for these patients.

$$\text{Physician specificity} = \frac{56}{75} = 0.75$$

Le's version of TrenDx denied referrals to 19 of the 36 patients who did not need to be referred according to the expert gold standard. TrenDx$_{\text{LE}}$ referred 17 of these 36 patients in disagreement with the referral decision of the pediatric endocrinologist.

$$\text{TrenDx}_{\text{LE}} \text{ specificity} = \frac{19}{36} = 0.53$$

In the version of TrenDx that was improved after Le's evaluation, TrenDx$_{\text{DESOUZA}}$, the specificity of the program was increased along with the sensitivity. TrenDx$_{\text{DESOUZA}}$ agreed with the pediatric endocrinologist on 25 of 36 patients in denying referrals. For 11 patients, TrenDx$_{\text{DESOUZA}}$ produced referrals, while the expert gold standard decision was not to refer the patient.

$$\text{TrenDx}_{\text{DESOUZA}} \text{ specificity} = \frac{25}{36} = 0.69$$

TrenDx$_{\text{BULL}}$, the new version of TrenDx, agreed with the expert gold standard in not referring 19 patients of the 36 non-referral patients of the standard. TrenDx$_{\text{BULL}}$ decided to refer 17 patients to whom the pediatric endocrinologist denied referrals.

$$\text{TrenDx}_{\text{BULL}} \text{ specificity} = \frac{19}{36} = 0.53$$

Diagnosis method	Correct Positive Referrals	Correctly Denied Referrals	Sensitivity	Specificity
Pediatric Endocrinologist Expert Gold Standard*	59	36	———	———
Physicians*	91 (of 142)	56 (of 75)	0.64	0.75
$\text{TrenDx}_{\text{LE}}$*	36	19	0.61	0.53
$\text{TrenDx}_{\text{DeSouza}}$†	38	25	0.64	0.69
$\text{TrenDx}_{\text{Bull}}$	36	19	0.61	0.53

∗: results from Le [5]

†: results from DeSouza [2]

Table 5.1: Summary of evaluation metrics.

5.3.2 Discussion of evaluation results

At first glance, the sensitivity and specificity of TrenDx, not to mention those of the physicians, seem rather low. However, it is important to note that the patient records used in this evaluation were taken from an endocrine clinic, a place where patient records would only be kept for children who were referred for a particular reason. This implies that their growth patterns may not follow behaviors of normal children, which a program like TrenDx would understand, but instead may follow a special scenario that was not incorporated into the growth trend templates. Experts (pediatric endocrinologists) may have special experience with these scenarios, therefore resulting in a specificities that are significantly lower than that which would be achieved from a sample of patients more closely resembling the whole population. The fact that the metrics for physicians and TrenDx are similar, however, is an encouraging sign that TrenDx may be useful in a clinical setting.

It is worthwhile to compare $\text{TrenDx}_{\text{Bull}}$ to $\text{TrenDx}_{\text{LE}}$. Each of these monitors were initial attempts at using the TrenDx trend templates to diagnose the state of a process. However, they differ greatly in their input models, and in the ways they score hypotheses. The first evaluation performed with these trend template descriptions in use was done with $\text{TrenDx}_{\text{LE}}$. The sensitivity and specificity of $\text{TrenDx}_{\text{LE}}$ were determined to be good enough to try to improve the monitoring system and further develop TrenDx. $\text{TrenDx}_{\text{Bull}}$ is at a similar stage in its development as $\text{TrenDx}_{\text{LE}}$ was at the time of its first evaluation. Both the sensitivity and specificity of $\text{TrenDx}_{\text{Bull}}$ are identical to the sensitivity and specificity of $\text{TrenDx}_{\text{LE}}$, which suggests that TrenDx$_{\text{BULL}}$ is also worth further development and evaluation.

When compared to TrenDx$_{\text{DESOUZA}}$, TrenDx$_{\text{BULL}}$ is less accurate. It is important to note, however, that TrenDx$_{\text{DESOUZA}}$ was specifically tailored to perform well in this domain, and with knowledge of the problems encountered by TrenDx$_{\text{LE}}$ during its evaluation. TrenDx$_{\text{BULL}}$ is a significant deviation from this process of iterated improvement, and therefore the drops in sensitivity and specificity back to the levels of TrenDx$_{\text{LE}}$ are not very worrisome.

The sensitivity of TrenDx$_{\text{BULL}}$ is comparable to the sensitivity of the physicians. However, the specificity of TrenDx$_{\text{BULL}}$ is significantly lower than that of the physicians. None of the versions of TrenDx have been able to match the specificity of the physicians without substantial costs in the sensitivity. There is a trade-off between sensitivity and specificity when the decision threshold is varied. Higher sensitivities may be desirable to warn doctors of patients at risk for certain disorders while suffering only minor costs in specificities. Conversely, higher specificities may be desirable to avoid over-referring patients to growth clinics, which would put to great of a demand on the time and skills of the specialists.

For TrenDx$_{\text{BULL}}$ to achieve a sensitivity of 0.64, the same sensitivity of TrenDx$_{\text{DESOUZA}}$ and of the physicians, the threshold value that is used to make referral decisions must be lowered to 0.650. With this threshold, the specificity of TrenDx$_{\text{BULL}}$ decreases to 0.47. However, if the threshold is lowered further to 0.645, the sensitivity increases to 0.68 while the specificity remains constant at 0.47. On the other hand, in order to achieve a specificity of 0.69 for TrenDx$_{\text{BULL}}$, the threshold value must be increased to 0.750. At this threshold, the sensitivity is drops dramatically to 0.34. However, as previously mentioned, the specificity in these evaluations does not accurately reflect the standard interpretation of specificity, due to the skewed sample set of patients.

Chapter 6

Conclusions

The results of the evaluation of TrenDx$_{\text{BULL}}$ in the pediatric growth domain were promising, and comparable to those of TrenDx$_{\text{LE}}$. There are several points of interest to note about this evaluation which suggest that TrenDx$_{\text{BULL}}$ may fare even better in a different domain.

6.1 Evaluation difficulties

The largest conceptual change introduced in TrenDx$_{\text{BULL}}$ was in the model of process data input. In previous versions of TrenDx, the data model consisted of a sequence of discrete data points. In TrenDx$_{\text{BULL}}$, however, the input model is a continuous data stream (i.e., a sequence of data points that accurately describes the underlying continuous process, as discussed in section 3.1.1). The previous versions of TrenDx were designed to perform well under their assumptions on the data, and the evaluation in the growth domain shows the feasibility of TrenDx in a domain with sparse and irregularly-spaced data points. An evaluation with this data set is not well suited to determining the monitoring abilities of TrenDx$_{\text{BULL}}$.

One of the reasons a domain in which the data points were sparse was used in the evaluations of previous versions of TrenDx was the fact that TrenDx was not able to process a large amount of data in a reasonable amount of time. Although the speed of TrenDx improved as computational technology advanced, it still suffered from the need to explore all possible discrete placements of temporal interval boundary points. During the evaluation of TrenDx$_{\text{DESOUZA}}$, it was noted that "On average, it takes a few minutes to process a patient" [2, page 14]. TrenDx$_{\text{BULL}}$, on the other hand, is able to process several patients (up to 10) per minute. The characteristics of the machine which was used for the trials of TrenDx$_{\text{DESOUZA}}$ are not known. It is suspected that the processor of the computer was a Pentium Pro running at approximately 200 MHz. The trials of TrenDx$_{\text{BULL}}$ were run on a SPARC-based CPU at a speed of 270 MHz. The increase in speed appears to be significant, though an exact comparison cannot be performed. It is possible that a large portion of the improvement is due to optimized architecture of the computer used as well as other technological advances since the trials of TrenDx$_{\text{DESOUZA}}$. Trials involving a larger number of data points are desirable to test the efficiency improvements of real-time TrenDx.

Another disadvantage encountered in evaluating TrenDx in the growth domain is the fact that the data samples are decidedly insufficient to reconstruct the entire growth pattern of the patient. The assumption made in the evaluation of TrenDx$_{\text{BULL}}$ was to linearly interpolate between height and weight measurements. With this assumption, the pivot stream presented to the monitoring algorithm consisted of the data points themselves. This is likely not the best reconstruction of growth patterns from the data points given. Further difficulties arose in the evaluation on this data set due to the absence of some values at different time points. Part of the assumption of the monitor is that some measurement would be present for the various parameters at each point in time, but this was not the case in the data collected by Le. For example, many patients had height measurements but no weight measurements at a few ages. Again, these values were linearly interpolated wherever possible for the evaluation.

Regardless of these difficulties, TrenDx$_{\text{BULL}}$ still performed with an accuracy identical to that of TrenDx$_{\text{LE}}$. In comparison to TrenDx$_{\text{DESOUZA}}$, the new version of TrenDx did not fare as well. However, there is a concern that TrenDx$_{\text{DESOUZA}}$ was engineered to over-fit the evaluation data. Its improved results were accomplished one patient case at a time, although the theoretical implications of each improvement were carefully considered prior to implementation.

In addition to the development of the monitoring system of TrenDx previously being conducted under the assumption of a discrete data model, the trend templates were engineered to produce good results with this input model. With the revised data model, it is likely that there could be some changes made to the growth trend templates that would improve the results of TrenDx$_{\text{BULL}}$. One particular area for concern is due to the allowance of TrenDx to apply arbitrary functions to its input before the value constraints are used. The monitoring system of $\text{TrenDx}_{\text{BULL}}$ expects linearly segmented parameter streams to score against value constraints. This caused the functions to be applied to the data points, producing another parameter, and using the values of that parameter to construct another data stream. Even based on a linear data point stream, there is no indication that this new parameter stream would be accurately described by linear segments between the data point pivots, since the functions applied to the data to produce the new parameter are not necessarily linear transformations. This would skew the error measure for the value constraints on those parameters. A domain expert was not on hand to assist with the effort of re-designing trend templates, but even still $\text{TrenDx}_{\text{BULL}}$ performed well without this additional improvement.

It would be desirable to evaluate $\text{TrenDx}_{\text{BULL}}$ in a domain more in line with this monitor's model of process data. Due to time limitations, however, an evaluation of this sort was not able to be performed. There is considerable effort involved in gathering the data needed to perform such an evaluation, along with the need to secure the services of a domain expert to assist in the engineering of the trend templates and to provide a gold standard for comparison.

6.2 Related research

The areas of knowledge that are related to the ideas in this thesis are quite wide in scope, ranging from time series analysis to medical informatics.

6.2.1 Time series analysis

Time series analysis techniques are typically used to study processes that are in some way periodic. The models used in time series analysis often consist of four components: seasonal components, cyclical components, trend lines, and stochastic components [6]. Although these models do contain trend lines as a component, the analysis techniques are generally more concerned with noise filtering and prediction of future values than they are geared toward diagnosis. Time series analysis techniques may be quite useful in monitoring domains, however, and in particular in conjunction with TrenDx, to pre-process data before the monitoring algorithm is employed.

The focus of time series analysis involves accurately modeling the stochastic component of the process [8], which may then be extracted from the process. In a monitoring capacity, the diagnosis of a process may be dependent on the type of interference to which it is subjected, or the process may become easier to diagnose with the interference removed. Furthermore, some techniques developed through time series analysis, such as analyses in the frequency and wavelet domains [7], are useful in detecting different characteristics of a process. These characteristics may be used in a monitoring capacity, such as in the form of parameters on which trend templates may place value constraints. Time series analysis is not particularly concerned with diagnosing processes, but may be used as part of a monitor system such as TrenDx to enhance the descriptive capabilities and diagnostic power of the system.

6.2.2 Neural nets

In recent years, neural nets have been used to perform recognition tasks [12]. Using a network of nodes, each of which performs a threshold-like function, a neural net can be trained through back-propagation to simulate any desired function. Unfortunately, the design of neural nets is rather difficult. One of the difficulties involves the representation of the data with which you wish the neural net to operate. Another drawback of neural nets is that they often require large amounts of training data. For the domains in which TrenDx is well-suited, there need not be much training data available, but instead a description of different diagnoses by an expert. If the size of the neural net in use is not appropriate, the function desired may be unlearnable (if the net is too small), or it may easily overfit the training data and not work well in test cases (if the net is too large). Furthermore, it is not clear how to apply a neural net to diagnose evolving processes such as those which TrenDx was designed to monitor.

6.2.3 Temporal reasoning

There have been several proposed methods for representing and using time in computational settings. Most methods used to represent time have been specific to the task which they are trying to solve. A few of the time representation structures used in applications related to diagnostic process monitoring are discussed below.

Temporal Control Structure

One attempt to incorporate time-ordered data into diagnostic monitoring modeled time through the use of memory variables. In the Temporal Control Structure (TCS) [10], all data are associated with either a point or an interval in time and are stored in point or interval variables. TCS maintains a historical database of data values over time, using propagation of data through memory variables to draw conclusions about the state of the process. However, TCS is limited in that it does not allow uncertain endpoints of temporal intervals and it does not allow variables to vary within a particular interval.

Knowledge-Based Temporal-Data Abstraction

Another approach to the problem of using data that varies over time in a computational setting is proposed by the Knowledge-Based Temporal-Data Abstraction method (KBTA) [11]. The goal of KBTA is to create interval-based temporal abstractions from a set of time-ordered data with a domain-independent methodology. KBTA decomposes the task of temporal abstraction into 5 subtasks, each of which is solved by mechanisms that depend on 4 domain-specific knowledge types. Only these knowledge types are domain-specific, and they are formally defined. This approach to abstracting qualitative temporal behaviors of time-oriented clinical data emphasizes the explicit representation of knowledge required for such a task.

6.2.4 Constraint programming

There has been interest recently in the development of improved constraint satisfaction solvers [9]. The trend detection problem discussed in this thesis can be viewed as a multi-dimensional constraint satisfaction problem (temporal constraints providing one dimension, value constraints providing additional dimensions). Constraint satisfaction solvers have been used with success in some commercial applications, resulting in a desire to develop a standard framework for representing and computing with constraints. Emphasis has been placed on developing constraint-programming libraries for common programming languages. Additionally, work has continued in improving the search-based methods that have been successful. A significant drawback of constraint programming is the difficulty of representing models that involve trends and temporal uncertainties with current available tools. Furthermore, many of these tools are not geared toward real-time problem solving, which is one of the goals of this thesis.

6.2.5 Medical expert systems

Several diagnostic programs have been developed in recent years that are commercially available. Many expert systems are developed around a rule-based paradigm. Due to the qualitative nature of descriptions of symptoms for various disorders in the medical domain, however, strict rule-based systems tend not to fare very well with their dependence on quantitative tests. Most systems use probability distributions in one way or another, as exhibited in the following comparison of four medical diagnostic programs [1]:

> Iliad and Meditel use Bayesian logic, but they differ in the assignment of prior probabilities, in specific decision rules, and in the use of expert judgment. Dxplain and QMR use non-Bayesian algorithms, but they incorporate semi-quantitative scales to express the probabilistic association of findings (signs and symptoms) with particular diagnoses, and they use these scales to derive a weighted assessment of the patients' combined signs and symptoms.

TrenDx does not have prior probability distributions inherent in is design, but through the engineering of trend templates and by weighting value constraints appropriately, similar biases to particular diagnoses can be achieved. There is no clear way to decide what the best design for a medical expert system will be, but through continued research the goal of attaining a reliable health monitoring system may well be achieved.

6.3 Future work

There are several areas to investigate in terms of further development of real-time TrenDx. Perhaps the most informative future work on TrenDx would be to perform an evaluation of real-time TrenDx in a domain that is more aptly suited for the monitor's abilities than the pediatric growth domain. Originally, TrenDx was perceived as being useful in an intensive care unit context. In that domain, the process data would more closely resemble the continuous data stream that real-time TrenDx expects. With the appropriate form of data input, the matching algorithm of real-time TrenDx is likely to perform better than on the sparse data set of the growth evaluation. Evaluating TrenDx in an intensive care unit setting would also be a good test to determine the efficiency of TrenDx and its practicality as a real-time monitoring tool.

The future development of real-time TrenDx is expected to be more enticing than the development of previous versions of TrenDx. Real-time TrenDx runs much faster than the older versions of the monitor. The previous versions of TrenDx were implemented in various dialects of the Lisp programming language. From version to version, the implementation had to be updated depending on the Lisp interpreter that was being used in conjunction with the type of operating system of the computer on which the program was being run. Real-time TrenDx was implemented in the Java language. Java was designed with portability and quick development considerations in mind. Under this design, Java programs are compiled into byte code that does not need to be modified when transferring files from one operating system to another, which is intended to make the programs platform independent. These advantages of the Java language should make it easier to maintain real-time TrenDx over evolving operating systems and computer technology.

Another area to look into for future development concerns the problem discussed above in which arbitrary functions may be applied to process data before applying a value constraint. As discussed above, this can cause skewed error measures to result from value constraints on parameters of this type. The Java language does allow arbitrary functions to be dynamically applied to variables, so it is possible that the type of function could be analyzed to determine an appropriate scaling mechanism for use with these constraints.

As discussed in section 4.1.2, the methods by which unspecified value constraints are fit to the data stream are sub-optimal in terms of minimizing the area between the segments. An optimal method of free parameter determination should be explored and incorporated into the monitor, although the current method was sufficient for use in this thesis. One unsettling fact of the method described for constraint fitting is that an unspecified constant constraint may be instantiated differently than a linear constraint with a specified slope of zero would be fit to the data stream.

Furthermore, it may be desirable for real-time TrenDx to allow the description of value constraints in terms of higher-order functions than simply first-order polynomials. Linear value constraints were used in this implementation due to the straightforward computations that they allow in finding areas, especially in combination with the process data input model.

Furthermore, the trend templates in existence did not specify any higher-order functions, and therefore linear constraints were sufficient for the evaluation. The mechanism required for generalized function constraints would be an integrator (to find the area between curves), which was not readily available. The implementation of real-time TrenDx was designed in such a way that expansion of the available types of constraints would be relatively easy for the developer. An integrator should be built into the monitor to enable generalized function constraints, which would not be a difficult task.

An important part of any system that is to be used by a variety of people in different fields is a well-designed user interface. Part of this user interface is for the end user of the monitoring system. For this person, it would be important for the monitor to provide a customizable display of its current state, including the rankings of the possible diagnoses of the process along with the reasoning it used to achieve those results. For the domain expert, it would be desirable to have an intuitive interface that she may use to encode her knowledge into trend templates. Neither of these interfaces have been developed. In the medical domain, there is a particular need for an explanatory display to be available to support a diagnosis.

Finally, in addition to exploring the applicability of TrenDx in the medical world, the feasibility of real-time TrenDx should be explored in other areas. A natural domain for a software-based monitoring system such as this is in the observation of computer systems themselves. In particular, TrenDx may be used for network monitoring purposes, to determine when a machine has been compromised. This has not been fully explored, but many areas such as this seem promising to demonstrate the usefulness of TrenDx.

Bibliography

[1] Eta S. Berner, et.al. Performance of four computer-based diagnostic systems. *NEJM*, 1994.

[2] Mary DeSouza. Automated Medical Trend Detection. Master's thesis, Massachusetts Institute of Technology, 2000.

[3] Ira J. Haimowitz. *Knowledge-Based Trend Detection and Diagnosis*. PhD thesis, Massachusetts Institute of Technology, 1994.

[4] Isaac S. Kohane. Temporal Reasoning in Medical Expert Systems. Technical report, Massachusetts Institute of Technology Laboratory for Computer Science, 1987.

[5] Phuc V. Le. A Clinical Trial of TrenDx: An Automated Trend-Detection Program. Master's thesis, Massachusetts Institute of Technology, 1997.

[6] G. McPherson. *Statistics in Scientific Investigation*. Springer-Verlag, 1990.

[7] Guy P. Nason and Rainer von Sachs. Wavelets in time series analysis. *Philosophical Transactions of the Royal Society of London A*, 357(1760):2511–2526, 1999.

[8] M. B. Priestley. *Spectral Analysis and Time Series*. Academic Press, 1981.

[9] Pierre Roy and Francois Pachet. Reifying constraint satisfaction in smalltalk. *Journal of Object-Oriented Programming*, 10(4):43–51, 63, 1997.

[10] Thomas A. Russ. *Reasoning with Time Dependent Data*. PhD thesis, Massachusetts Institute of Technology, 1991.

[11] Yuval Shahar. *A Knowledge-Based Method for Temporal Abstraction of Clinical Data*. PhD thesis, Stanford University, 1994.

[12] Patrick Henry Winston. *Artificial Intelligence.* Addison-Wesley, third edition, 1992.