Temporal Pattern Discovery in Course-of-Disease Data

Jorge C. G. Ramirez[1,2], Diane J. Cook[1], Lynn L. Peterson[1], Dolores M. Peterson[2]

[1]Department of Computer Science & Engineering
University of Texas at Arlington
PO Box 19015, Arlington, TX 76019-0015
Tel: 817-272-3620; Fax: 817-272-3784
ramirez@cse.uta.edu

[2]HIV Clinical Research Group
Division of General Internal Medicine
University of Texas Southwestern Medical Center

Abstract

The goal of the research being reported is the discovery of useful concepts in temporal medical databases. Building on previous experiments, we introduce TEMPADIS, the Temporal Pattern Discovery System, which uses our Event Set Sequence approach to discover patterns in this type of data. Results are presented for a database of Human Immunodeficiency Virus (HIV) patients. We discuss the overall process of knowledge discovery including data cleaning and preprocessing, data reduction and projection, and matching goals to a particular data-mining method.

Introduction

With the recent explosion of research in the area of knowledge discovery in databases (KDD), advances in, as well as problems with, KDD and data mining are being researched and documented. Fayyad, Piatetsky-Shapiro, and Smyth [1] give a good overview of the KDD process, the role of data mining, and associated application issues. According to them, "KDD refers to the overall process of discovering useful knowledge from data, and data mining refers to a particular step in this process. Data mining is the application of specific algorithms for extracting patterns from data." [2] They enumerate nine steps in the KDD process:

1. Understanding application domain/identifying goal of KDD process
2. Creating a target data set
3. Data cleaning and preprocessing
4. Data reduction and projection
5. Matching goals to a particular data mining method
6. Exploratory analysis/model and hypothesis selection
7. Data mining
8. Interpreting mined patterns
9. Acting on the discovered knowledge

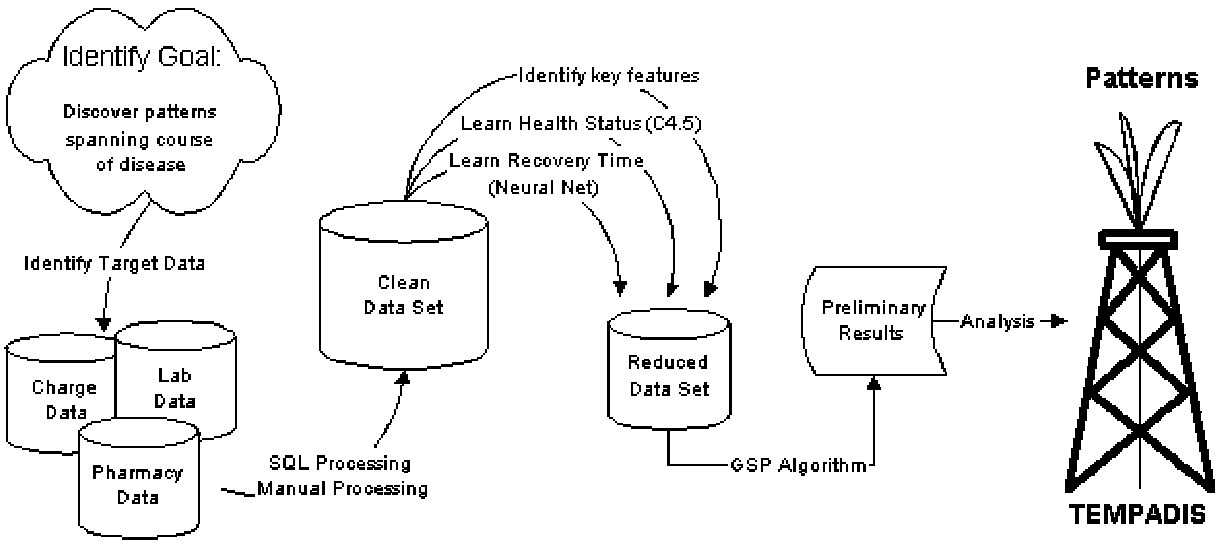

Figure 1. Overview of development of TEMPADIS.

We use the development of our Temporal Pattern Discovery System (TEMPADIS) to help understand the overall process of knowledge discovery in a medical database environment. Figure 1 diagrams the steps in the knowledge discovery process using TEMPADIS. We started with the hypothesis that, given a database of clinical data for patients who had the same catastrophic or chronic illness, we could discover that subsets of those patients had a similar experience during the course of that disease. Once we found a database suitable to test this hypothesis, we embarked on the process of knowledge discovery.

Identifying the Goal

We started with the hypothesis stated above. Given a database that contains clinical data for patients diagnosed with the same catastrophic or chronic illness, we are interested in discovering patterns in that data that show that groups of patients had similar experiences during the course of the disease. The motivations for such research are many. With advances in medical technology have come many methods for treating such illnesses. Analysis of the course of such diseases is beneficial from multiple points of view, including enhancement of provision of care, prognosis, monitoring, outcomes research, cost/benefit analysis, and quality assurance. This type of research is also beneficial for development of techniques of pattern discovery for other data collections that have similar characteristics.

Understanding the Domain

One of the important issues in the KDD process is availability of sufficient data to discover complex patterns in data, particularly when there are a large number of data fields. Our domain is HIV disease. Our data collection is the Jonathan Jockush HIV Clinical Research Database which was established in 1987 at the University of Texas Southwestern Medical Center at Dallas. The database contains data for over 8,500 patients of the Acquired Immune Deficiency Syndrome (AIDS) Clinic at Parkland Memorial Hospital, also in Dallas. The database consists of data collected from the hospital charge system, the pharmacy system, and the laboratory information system. In addition, some data is entered directly from patients' day sheets and charts. Choosing what specific data to consider is a complex task.

The data available for our analysis is a combination of binary, numeric, symbolic and text data fields. Beyond the variety of data, there are several other aspects of the data to consider. First, especially where laboratory, diagnosis, and therapy data are concerned, most of the data is temporal, i.e., there are multiple instances of the same data field with different dates and values for each instance.

Second, the importance of the temporal aspect depends on the specific type of data. For some of the data (e.g., clinic or emergency room visits, or lab test results) the occurrence of the event and the order of occurrence relative to other events is important. For other data (e.g., diagnoses or drug therapies), the duration of an event is just as important as the relative order of occurrence.

Third, the set of data fields that exist for each patient not only varies greatly between collection dates for any given patient, but also varies between patients even for similar medical events. We are studying methods of discovering useful patterns in this temporal, non-standard form, variable data-field, medical data.

Creating a Target Data Set

The objective of this phase is to select a data set or focus on a subset of variables or data samples. The objective for us was to have enough data to discover significant patterns, but to focus on as small a set of variables as would produce useful results. We decided to select a subset of the patients, and then select a subset of the available variables.

We examined the database to see how long patients had been monitored, and how many times they had been seen. Of the over 8,500 patients in the database, there are many who had been monitored for only a short period of time. Several hundred have only been seen for a month or less. On the other hand there are several hundred that have been monitored for seven years or more. We needed to find a group of patients on whom data had been collected for a significant length of time and often enough to have a significant detectable pattern. Approximately 1,100 of the patients have been monitored by the Parkland system for at least 4 years, with a minimum of 30 distinct dates when at least one type of event (i.e., charge, pharmacy, lab test result, etc.) was recorded. We have randomly selected groups of patients from these 1,100 patients for the results given in this paper. The number of patients used for any given task is discussed as that task is discussed.

Next, we looked at the available variables. Given the mass of data that is available, it is important to focus on finding some minimum number of variables that still is sufficient to represent the concepts that are to be discovered. To help us focus, we used an initial goal of identifying a dozen key variables, although we finally used twenty. This subset of the available data includes all encounters with patients, a subset of the laboratory results, and a subset of the pharmacy data. Later, in step four of the KDD process, data reduction and projection, we add the final two variables that are measures of the patient's overall health status.

An encounter with a patient is represented by two variables. The first variable is the type of encounter. The three types are clinic visits, emergency room (ER) visits, and hospital stays. The second variable is the level of severity of the encounter. For example, it is not uncommon for patients to go to the ER, when they could have gone to the clinic had it been open. This type of visit would not be considered as severe as an ER trauma visit (e.g., gunshot wound).

Conversely, patients will sometimes go to the clinic when they really need to be hospitalized. Further, when hospitalized, the severity is different if the patient requires intensive care rather than placement on one of the regular wards. Therefore, each type of encounter has graded levels of severity.

There are literally thousands of different medical laboratory tests. While examining this data we noted that there were some tests that were ordered significantly more times than others. We also noted that the database contains duplicate data for several different time periods, i.e., that particular data was in fact loaded into the database twice during those time periods.

We then examined the pharmacy data. In order to see what types of drugs were being used on these patients, we randomly selected 100 patients, and found that 71 different drugs had been given to them that were specific to HIV and/or HIV-related illnesses. We also discovered that there were many errors in how the various prescriptions were recorded in the pharmacy's computer system. In an extreme example, one drug was coded 46 different ways throughout the database.

Data Cleaning and Preprocessing

The purpose of this step is to remove noise from the data or develop a method for accounting for noise in the data, look at strategies for handling missing data, and handle any other known changes that need to be made. The business of cleaning up the data is a long and tedious task. In our case, some identified problems could be cleaned up as a group by processing the database with SQL statements. One example would be to correct the same misspelling and/or miscoding of a drug in the pharmacy data that appeared multiple times. Other problems required a manual search of the data to remove duplicate records. In one example, a charge for a single service was entered five times, and then corresponding records were entered to reverse the charge four times, with a net result of nine records for a single charge event.

Since the problems that could be cleaned up with SQL statements were easy, they were corrected for the entire database. However, handling the manual problems in the entire database would have been too time-consuming. Therefore, we randomly selected a subset of 400 patients from our 1,100 described above in order to have a significant base of patients to work from without losing sight of the original goal of our project. It still took approximately 3 man-months to clean up those 400 patients' data. It is important to note that we successfully corrected the obvious errors. However, errors due to missing or incorrectly recorded data were not fixed.

Data Reduction and Projection

The purpose of this step is find useful features to represent the data depending on the goal. This can involve using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or finding invariant representations of the data. Our goal was to use only a minimum number of key variables that well represented the data. First, we reexamined the lab test results. We focused on variables that were most likely to be recorded on any given encounter and were important indicators of the health status of an HIV-infected patient. The six variables chosen are White Blood Cell (WBC), Hematocrit (HCT), Platelets (PLT), CD4 Absolute (CD4A), CD4 Percent (CD4P), and Lymphocytes (LMPH).

Knowing that the “normal” value of these lab results would vary from patient to patient, we developed a methodology for normalizing all the values for each patient so that comparisons could be made with other patient’s data [3]. Each variable was treated separately for each patient. All were normalized to a range of integers from –4 to +4. In this range, 0 is normal, and both -4 and +4 are indicative of severe illness and are roughly equivalent to the number of standard deviations away from normal for that person. The methodology is based on statistical norms for the general population with adjustments made for the fact that immune-compromised patients tend to have lower than normal values. However, the methodology takes into account the fact that the normal value for any given patient may or may not be different, compared to the general population.

Choosing a subset of the pharmacy data proved to be a more challenging task. Since we are trying to find patterns that represent similar experiences during the course of disease, the most obvious solution was to group the drugs into categories according to the reason they were being prescribed. This yielded the following ten categories: Nucleoside Analogs, Protease Inhibitors, Prophylaxis Therapies, Intraveneous antibiotics, Anti-virals, Anti-pneumocystis pneumonia/anti-toxoplasmosis, Anti-mycobacterials, Anti-wasting syndrome, Anti-fungals and Chemotherapies. We decided that simply tracking whether or not a patient is on a drug in a particular category on a given day is sufficient information for tracking the patient’s overall course of disease.

At this point we have 18 variables as shown in Table 1. In our terminology, an “event” is the occurrence of one of these variables together with its value on a given day.

Table 1. The 18 Data Elements Extracted Directly From the Clean Data Set.

Patient Encounter	Drug Categories
1. Event Type	9. Nucleoside Analogs
2. Event Severity	10. Protease Inhibitors
	11. Prophylaxis Therapies
Blood Tests	12. Intraveneous antibiotics
3. White Blood Cell (WBC)	13. Anti-virals
4. Hematocrit (HCT)	14. Anti-pneumocystis pneumonia/ anti-toxoplasmosis
5. Platelets (PLT)	15. Anti-mycobacterials
6. CD4 Absolute (CD4A)	16. Anti-wasting syndrome
7. CD4 Percent (CD4P)	17. Anti-fungals
8. Lymphocytes (LMPH)	18. Chemotherapies

The next data we considered is also the group of data in the database collected in the least automated way, the diagnosis data. This data was being manually entered directly from the patient day sheets each time the patient visited the clinic. However, the resources allocated to this task were not sufficient, and this data is rather incomplete. As a result, we chose not to use the available diagnosis data for our discovery purposes. We instead looked for another way to obtain equivalent information.

After discussions with the clinicians in the HIV Clinical Research Group, we decided to use the pharmacy data to learn about the current state of a patient's health. Further, we decided that the level of detail needed for health status was not fine-grained, and that the five health status categories, shown in Table 2, would be appropriate.

Table 2. Health Status Categories.

1. Asymptomatic, not on any therapy
2. Asymptomatic, only on anti-HIV therapy
3. Immune system significantly damaged, on prophylactic therapy
4. Active opportunistic infection/illness
5. Severe/Life-threatening illness

We used a machine learning technique called decision tree induction to develop rules for determining the health status value for any given patient on any given day. It is important to note that even though we used a machine learning technique to do this data reduction, this is not the data-mining step. In order to have a single variable that represents the general health status, we used decision tree induction to learn how to determine this value from a larger set of data.

Decision tree induction is commonly used for classification problems of this type. Decision trees are built using varying rules about the information gained by splitting the remaining unclassified examples on each of the remaining unused attributes. In order to induce the tree for determining the correct health status value, we developed a set of test data by randomly selecting four days from each of 100 patients and listing all drugs being taken on those days. Three clinicians then rated the health status for each day, according to the categories listed in Table 2, based solely on the drug information. If there was a discrepancy among the clinicians' ratings we went back to them and got a consensus.

We then used C4.5 [4] to induce the decision tree. The inputs were the standard files required by C4.5. The names file contained the classes (which were the health status categories listed in Table 2), the attributes (the various drugs), and the attribute values (1 if currently on the drug and 0 otherwise). The data file contained the 400 samples with the attribute values and the class. We ran C4.5 in iterative mode using the default parameters, except that we specified a 95% confidence level on the pruned tree. The resulting tree was converted to rules that were used to determine health status values. Then a health status (HS) field was inserted in each day of each patient's data as part of the data reduction preprocessing.

Now that we had a measure of health status, which gave us an idea of the current state of the patient, we needed a measure that gave us a feel for how long the patient might remain in that state. The health status only tells what is currently wrong with the patient, but it gives no indication of how severe the current problem is. For example, health status could be 1, 2 or 3 and the patient could have the flu. A severity measure would provide a way to differentiate between that patient and another that has the same health status but no current illness. Further, a patient could have health status 5 and be near death or could be well on the way to recovery. Again, a severity measure would provide a means to differentiate these two states. The combined information provided by the health status measure and a severity measure can provide a significant increase in the meaning not only of the current state of a patient, but also of a discovered pattern. The severity of the current event can be measured by determining how long it would take to recover from that event. However, this is not the type of information that appears in the database. Once again, we turned to a machine learning technique to inject information not already explicitly present to enhance the discovery process.

We chose a neural network to learn the recovery time function. Neural nets have been used in a variety of ways in medical domains [5-8], including prediction of length of hospital stay, so it was reasonable to assume we could use it to predict length of recovery time. The next task was to select the relevant inputs for determining some measure of recovery time. The selected inputs are shown in Table 3. As we discussed this with the HIV Clinical Research Group staff, needing to know diagnoses was again raised as being important to this determination. Of course, we already knew that data was not available; however, we did have our newly determined health status, which was developed to be a measure of the nature of the current illness. This lead to the decision to include the health status measure to represent diagnoses as one of the inputs for our neural net.

Table 3. Recovery Time Neural Net Inputs.

1. Days Since Previous Event
2. Days Until Next Event

3-5. Health Status (Previous, Current and Next Values)
6-8. Event Type (Previous, Current and Next Values)
9-11. Event Severity (Previous, Current and Next Values)
12-29. Normalized Blood Test Values (Previous, Current and Next Values for each of WBC, etc.)

For additional inputs to the neural net, we include the current event type, since the output is a measure of how long it takes to recover from the current event. The current laboratory test results are also included. Further, in order to put the current event in context, we choose to include the same data related to the previous event and the next event, as well as the time since the previous event and the time until the next event.

For training cases, we randomly selected six days from each of 50 patients. We then abstracted the data needed for the neural net inputs. We again had three clinicians rate the recovery time of the patients based solely on the information we would be providing to the neural net. We originally asked them to predict in number of days. When we saw a significant disparity in the predictions, we decided that we needed to decrease the granularity of the measure. Therefore, we use a scale of 0 to 5, where 0 to 4 represented estimated weeks to recovery, and 5 represented anything over 4 weeks. Again, where we found discrepancies, we went back to the clinicians for a consensus.

We use the NevProp3 neural net software [9] and its default 2/3-1/3 holdout, five sample cross validation. NevProp3 only allows for a single hidden layer. Within that context we experimented with various network structures. As shown in Table 4, the network with six hidden nodes performs the best, with an 85.3% correct prediction rate. The MeanSqErr is the mean of the squared differences between the predictions the model made and the target values designated by the clinicians, where 0 is best. R^2 is commonly interpreted as the fraction of variance explained by the model, where 0 means that the model predicts the mean of the target values and 1 means that the model predicts the correct target value. NevProp3 allows a model to be saved and then used to generate values on new data inputs. The six hidden node model was saved and used to generate a recovery time (WTR) field for each day of each patient's data as part of the data reduction preprocessing. This new knowledge, coupled with the health status, gives more overall meaning to the data in the absence of an explicit diagnosis.

Table 4. Results of Neural Net Training.

Hidden Nodes	MeanSqErr (Best=0)	R^2 (Best=1)	Predicted (Best=1)
4	0.216	0.886	0.713
5	0.181	0.905	0.813
6	0.140	0.926	0.853
7	0.174	0.909	0.810

The result of the data reduction and projection step is the reduced data set. This data set consists of 20 variables, the 18 listed in Table 1 plus HS and WTR, deemed to well represent the larger set of all data available.

Matching Goals to A Particular Data Mining Method

The purpose of this step is to select a method or methods to be used for searching for patterns in the data. This involves deciding which models may be appropriate and deciding which data mining method or methods match the goals of the KDD process. Model selection is usually based on what type of data is being mined, and mining method selection is based on what the end results needs to be, usually discovery or prediction.

```
< { (EV C ) (HS 3) (WTR 0) (WBC 0) (PLT -1) (HCT 0) (LMPH -3)
    (onD 0000000000) }
  { (EV E ) (HS 3) (WTR 2) (WBC 3) (PLT -1) (HCT 1) (LMPH 4)
    (onD 0000000000) }
  { (EV C ) (HS 3) (WTR 0) (WBC 1) (PLT 0) (HCT 0) (CD4P -3)
    (CD4A -1) (LMPH 0) (onD 1010000000) }
  { (EV C ) (HS 3) (WTR 1) (WBC -1) (PLT -1) (HCT 1) (LMPH 2)
    (onD 1010000000) }
  { (EV E ) (HS 3) (WTR 1) (WBC 2) (PLT -1) (HCT 1) (LMPH 4)
    (onD 0000000000) }
  { (EV C ) (HS 3) (WTR 1) (WBC 1) (PLT 0) (HCT 0) (CD4P -3)
    (CD4A -2) (LMPH 0) (onD 1010000000) } >
```

Figure 2. Typical pattern discovered by TEMPADIS.

We are trying to discover patterns in sequences of events across patients in a database. An example of the type of patterns that we have discovered with TEMPADIS is shown in Figure 2. This pattern contains six event-sets; each enclosed in curly braces. The first event-set is based on a clinic visit (EV C), the patients' health status is 3 on the scale of 1 to 5 described above (HS 3), and the recovery time measure is 0 on the scale of 0 to 5 described above (WTR 0).

Further, only four of the six lab tests were run on this visit. These are (WBC 0), (PLT –1), (HCT 0) and (LMPH –3). The only one that would be of significant concern is the lymphocytes being considerably low at –3. The other two lab tests, CD4P and CD4A, can be seen in the third and sixth event-sets. Finally, the ten drug categories are represented by the binary values 0 or 1. In the first event-set, (onD 0000000000), means that none of the drug categories was currently being taken by the patients. However, in the third event-set, drugs from category 1, Nucleoside Analogs, and category 3, Prophylactic Therapies, are indicated as currently being taken.

In our review of the literature, we found that there were only a few data mining methods relevant for our goal [10-14]. We chose Srikant and Agrawal's General Sequential Patterns (GSP) Algorithm [10,14] as the basis for the data mining method we would use.

The GSP Algorithm uses atomic events as the basis for building up sequences. In our domain an example of an atomic event, as seen in the example pattern above, would be (WBC 0), which is the occurrence of a White Blood Cell test result that is in the normal range for that patient. The database is searched for all atomic events that occur in the database, and then each atomic event is checked for support by the database. In our domain, support is the percentage of the patients in the database that had that event occur at least once. Only those events that meet the support threshold are "supported" by the database. We have used various support threshold levels from 6.6% (i.e., 1 of every 16 patients should support the pattern) to 50%. Those atomic events that survived are then combined as pairs, both as sequences and as concurrent occurrences. For example, if we had the atomic events (WBC 1) and (HCT 0) which were supported by the database, then the resulting combinations would be <{(WBC 1)(HCT 0)}>, which is a concurrent occurrence, and <{(WBC 1)} {(HCT 0}> and <{(HCT 0)} {(WBC 1)}>, which are sequences. All three are considered to be sequences of length two because they contain two atomic events. These sequences of length two are checked for support by the database. Only those with enough support contribute to what are called candidate sequences for the next iteration. Once the supported sequences are at least length two, the next generation of candidate sequences is created by joining together sequences which are supported at the previous length. For example, given supported sequences of length two:

<{(WBC 1)(HCT 0)}>
<{(HCT 0)} {(PLT -2)}>
<{(PLT -2)(CD4A -1)}>

Then we could generate two candidate sequences of length three:

<{(WBC 1)(HCT 0)} {(PLT -2)}>
<{(HCT 0)} {(PLT -2)(CD4A -1)}>

If both of those sequences were supported by the database, then we could generate the length four candidate sequence:

<{(WBC 1)(HCT 0)} {(PLT -2)(CD4A -1)}>

Note that the joined sequences must be exactly the same except for the first event of one and the last event of the other.

This continues until there are no candidate sequences that have support at the current level. We would say that the discovered pattern shown at the beginning of this section was of length 104, since it includes 104 atomic events, if you consider each drug type a separate event. This pattern would have been discovered on the 104th iteration of the algorithm.

The GSP Algorithm also provides the windowing concept. The minimum and maximum gap (i.e., the allowable time between events for them to be considered to have happened consecutively) can be specified. In our domain, the minimum time between consecutive events would be 0 days, since we are interested in events that may happen on consecutive days (e.g., hospitalizations). Currently, we are using 90 to 120 days as the maximum gap. This range allows for a patient, who is going through a period of relatively good health, to only see the doctor every 2 to 4 months for follow-up visits. Otherwise, his or her sequence of event-sets would be partitioned (i.e., split into two parts at the point at which time between visits is greater than the maximum gap). This maximum gap, of course, allows for more frequent visits by those who are not so healthy. Further, the time window within which events can happen and still be considered to part of the same event-set can also be specified. In our domain, a window of 7 to 10 days is necessary to allow patients to come to have lab work done a week in advance of their next clinic appointment.

Exploratory Analysis/Model and Hypothesis Selection

The purpose of this step is to evaluate the model and data mining method selections. This can result in modifications and refinements to the original selections. Further, upon seeing the exploratory results, hypotheses can be made about what are the realistic results of the particular KDD process.

We ran our exploratory analysis on a modified version of the original GSP Algorithm model. Though the basic algorithm is the same, the details of implementation are different. The GSP algorithm was designed to work on sequences of events that either occurred or did not, where the occurrence or lack thereof was significant to the patterns discovered. Also, none of the events had attributes. The differences in domains lead to several significant observations. In our domain, the occurrence of an event or lack thereof does not necessarily have any specific significance. The events themselves have attributes, especially when viewed from the event-set perspective. Finally, the shear numbers of events being dealt with computationally strains an algorithm that was designed to discover patterns at an individual event level. We concluded that our original modified-GSP implementation was insufficient [3]. However, those experiments led us to propose our Event Set Sequence approach and a further modification to the GSP Algorithm. We call the system that implements this approach TEMPADIS, or the TEMporal PAttern DIscovery System.

TEMPADIS

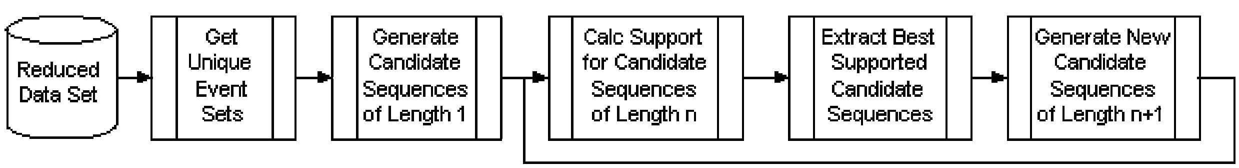

Figure 3. Algorithm for TEMPADIS.

Our concept of an event-set is based on the idea that the events in the database exist for one of three reasons: 1) some type of visit to a medical facility was made; 2) laboratory tests were performed; or 3) prescriptions were dispensed. Generally those events that happen on the same day or on days very close together are all related. For example, a patient who is having periodic check-ups may have lab tests done about a week prior to an appointment to ensure the test results will be available on the day they actually see the doctor. Further, they may not pick up a prescription until a day or two after seeing the doctor. Therefore, we have incorporated the time-windowing technique from the GSP Algorithm, that allows for those events to be considered as a single event-set. We use a set difference method that allows us to compare event-sets, looking at all 20 variables as a single unit. The algorithm for TEMPADIS is shown visually in Figure 3 and listed below:

1. Read database
2. Get unique event-sets from database
3. curSeqs = GenerateNewSeqs from unique event-sets
4. while curSeqs

4a. CalculateSupportInDatabase for curSeqs
4b. supportedSeqs = ExtractBestSupportedSeqs from curSeqs
4c. curSeqs = GenerateNewSeqs from supportedSeqs
endwhile

Whereas GSP retrieved the unique atomic events, in step 2 we create a list of all of the unique event-sets in the database. In step 3, we put these unique individual event-sets into a sequence format of length one. Step 4 is to continue the algorithm as long as there are sequences to consider.

In step 4a, we determine the support for each sequence under consideration in the database. This is where our algorithm differs significantly from the original GSP. In GSP, the current sequence under consideration was supported by a specific instance in the database on an all or nothing basis, i.e., it supported it or it did not. Our algorithm is necessarily fuzzier than that.

There are many parameters that can affect what patterns are supported by the database. Above we mentioned the support threshold as being one. Within the CalculateSupportInDatabase function of step 4a there are several more. In this function, there is a critical sub-function called DegreeOfMatch. The method for determining the set difference can be varied and the weights of the individual elements of the set can be varied. In the current version, we have not yet addressed the issue of missing data in any in-depth manor. For example, if a lab result is missing, in either the current sequence under consideration or in the particular database instance we are looking at for support, or both, then we give that a value of 50% support for that element of the event-set. However, the issue of missing drug data does not get addressed at all. If the drug is not present in the database, then there is no support for that element of the event-set, even if upon visual inspection of a patient's records we could reasonably assume they were on the drug at the time. These issues will be addressed in future work.

For the data that is present, we use a partial match system. For example, for WBC we might decrease the degree of match by 33% for each point difference on our scale of –4 to +4. In this example, if the current sequence under consideration has a value of -1, then for the value in the particular database instance of -1 we would give 100% match. For values of 0 or –2 we would give 67% match, and for values of 1 or –3 we would give 33% match. All other values (i.e., -4, 2, 3 and 4) would be given 0% match.

The net result is that DegreeOfMatch returns a value ranging from 0.0 to 1.0 for each event-set in the length of the sequence. TEMPADIS uses a weakest-link/average-link method for determining whether or not a sequence under consideration is supported by a given patient's data. For example, the weakest-link value might be set to 0.72, and the average-link value might be set to 0.80. This means that every event-set in the sequence must have at least 0.72 as its

DegreeOfMatch, but the entire sequence must average at least 0.80 before it is considered to support the candidate sequence.

The last thing to consider in step 4a is the fact that each patient in the database might have multiple instances of support for a sequence currently under consideration. Because of this, and the fact that we want the best supported sequences to be found, each instance must have its support value calculated. Then only the highest value is saved for use in calculating the total support of the database for that sequence. These highest values are summed for all patients in the database that met the average-link threshold. If the sum is greater than the number of patients times the average-link value, then the sequence is considered to be supported by the database.

From the previous explanation, one might imagine that TEMPADIS is very computationally intensive, and it is. Therefore, as one of the many search control strategies that we have implemented, step 4b limits the number of sequences that can be carried over to the next iteration. We use a pruning method that considers a minimum number of sequences under which no pruning will be done, a maximum number over which pruning must be done, and a pruning factor, all of which can be varied.

Finally, step 4c generates the new set of sequences for consideration on the next iteration. We incorporate intelligent selection of the event-sets with which to attempt to lengthen the patterns. The intelligent selection is based on the event-sets that were present in the database immediately prior to and immediately after the best supported patterns within each individual patient. This list was saved during step 4a. Each sequence can spawn many new sequences during this step, including many duplicates. However, we use a hash tree to track the newly generated sequences and it discards duplicates as they are generated.

Data Mining

After all of the above steps of the knowledge discovery process are completed, we are finally ready to begin using TEMPADIS in the data-mining step. The data has been cleaned, preprocessed and reduced. We have incorporated knowledge learned from the database back into the database to give us more intelligent data from which to discover. We have developed a technique which reduces the computational complexity, and now we put our Event Set Sequence approach to the test.

We have stated that we use multiple methods of search control to reduce the computational complexity. When we begin the data mining, we implement search control strategies that will only look at patterns that are of interest. We initialize our search strategy during step 2, Get unique event-sets from database. If we are particularly interested in patterns with hospitalizations, then we would only retrieve all unique event-sets that have a hospital stay as the visit type. If we are particular interested in patterns that have a specific trend in a specific variable then we screen for that trend during step 4c, GenerateNewSeqs.

Interpreting Mined Patterns

The purpose of this step is to look at what was found and make some sense of it. It may result in returning to a previous step and revising or fine tuning it. Once the algorithm has completed on a given set of parameters, the clinicians can examine the patterns for significance and meaning. The director of the HIV Clinical Research Group examined the example pattern, shown in Figure 2. Her conclusion was:

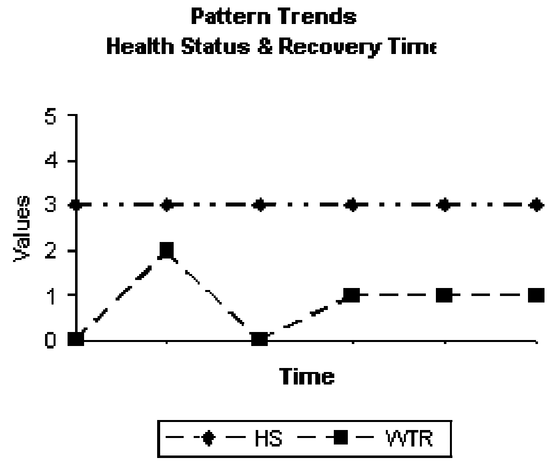

Figure 4. Pattern Trends in Health Status and Recovery Time.

> "These [look] like fairly advanced patients in the era of poor or no anti-retroviral suppression of their viral loads. Therefore, they would be subject to any number of viral infections such as CMV "flares" which would likely make their lymphocyte counts go up. The cause of CMV flares is unknown but may be from any number of causes such as mild "colds", etc."

She observes that the patients have "poor or no anti-retroviral suppression of their viral loads". This conclusion can be drawn from that fact that it was not until 1996 when Protease Inhibitors came into use for suppressing replication of the HIV, and it was only at that point when viral replication was successful repressed in large numbers of HIV patients. The pattern clearly shows no use of Protease Inhibitors (drug category 2) and also shows sporadic use of the other drug therapies. Her comments further reflect the relative flatness of all the variables (see Fig. 4 and Fig. 5) except the white blood cells and the lymphocytes, which jump around significantly. Once a pattern is deemed to be significant or interesting, we can look at the specific patients that supported the pattern and do the various types of analysis mentioned above on this sub-population.

Discussion

Once we have run through the KDD process, we can evaluate the results. Fayyad, Piatetsky-Shapiro, and Smyth [15] give us some criteria for evaluating discovered patterns. They tell us patterns should be understandable and novel, but that these concepts are subjective. They further state that the patterns should be "potentially useful", leading to some benefit to the user. They also note that a the concept of interestingness [16] is an overall measure of the pattern value, combining factors such as validity, novelty, usefulness and simplicity, but can be explicitly defined, or manifested implicitly by the system itself.

It is clear from the presentation that the pattern in Fig. 2 was found to be understandable. In fact, it seemed to represent a specifically recognizable condition. TEMPADIS is biased towards discovery of interesting patterns through the use of search control parameters that allow the user to specify what kinds of patterns would be interesting. As for the usefulness, that is the next step. Now that groups of patients can been identified by a pattern, that group can be investigated for the purposes that the user originally had for specifying parameters that would yield patterns of that type.

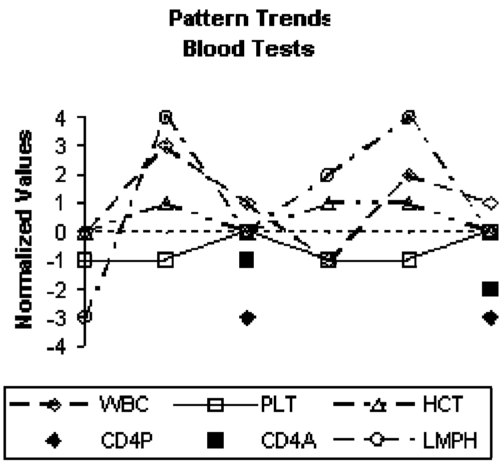

Figure 5. Pattern Trends in Blood Tests.

Other measures that we considered for evaluation purposes include significance in terms of number of patients represented, and length of time covered by the patterns. With the parameters used for the pattern in Fig. 2, the pattern would have only had to represent one out of every sixteen patients in order to be discovered by TEMPADIS. However, this pattern actually represents one out of every 9.6 patients. Again with the parameters used for that particular run, this pattern likely spans a period of from 1 to 12 months. Examination of the 10 specific patients supporting the pattern shows that it actually varies from 5 to 10 months. Given that the majority of our patient data spans 48 to 60 months, patterns of this length are non-trivial.

Our experience with TEMPADIS reaffirms that this type of problem easily becomes intractable. It is clear that this approach cannot be used to randomly sift through the database to discover whatever patterns might be out there. However, as stated earlier we are not interested in all the patterns that could be found. With careful use of search control, TEMPADIS can be used to discover meaningful patterns in areas of specific research interest.

References

1. Fayyad U, Piatetsky-Shapiro G, Smyth P: From data mining to knowledge discovery in databases. AI Magazine (Fall):37-53, AAAI, Menlo Park CA, 1996.
2. Ibid, p 39.
3. Ramirez JCG, Peterson LL, Peterson DM: A sequence building approach to pattern discovery in medical data. In: Cook, DJ (Ed): Proc Eleventh Int Florida Artificial Intelligence Research Symp Conf (FLAIRS-98), AAAI Press, Menlo Park CA, pp 188-192, 1998.

4. Quinlin JR: C4.5: programs for machine learning. Morgan Kaufmann, 1993.

5. Dombi GW, Nandi P, Saxe JM, Ledgerwood AM, Lucas CE: Prediction of rib fracture injury outcomes by an artificial neural network. J of Trauma: Injury, Infection, and Critical Care 39(5):915-21, 1995.

6. Frye KE, Izenberg SD, Williams MD, Luterman A: Simulated biologic intelligence used to predict length of stay and survival of burns. J of Burn Care and Rehab 17(6):540-6, 1996.

7. Izenberg SD, Williams MD and Luterman A: Prediction of trauma mortality using a neural network. American Surgeon 63(3):275-81, 1997.

8. Mobley BA, Leasure R, Davidson L: Artificial neural network predictions of lengths of stay on a post-coronary care unit. Heart and Lung 24(3):251-6, 1995.

9. Goodman PH: NevProp neural network software, version 3. University of Nevada, Reno, 1996. (ftp://ftp.scs.unr.edu/pub/goodman/nevpropdir/index.htm)

10. Agrawal R, Srikant R: Mining sequential patterns. In: Proc Eleventh Int Conf on Data Engineering (ICDE-95), pp 3-14, 1995.

11. Mannila H, Toivonen H, Verkamo AI: Discovering frequent episodes in sequences. In: Proc First Int Conf on Knowledge Discovery in Databases (KDD-95), AAAI Press, Menlo Park CA, pp 210-15, 1995.

12. Mannila H, Toivonen H: Discovering generalized episodes using minimal occurrences. In: Proc Second Int Conf on Knowledge Discovery in Databases (KDD-96), AAAI Press, Menlo Park CA, pp 146-51, 1996.

13. Padmanabhan B, Tuzhilin A: Pattern discovery in temporal databases: a temporal logic approach. In: Proc Second Int Conf on Knowledge Discovery in Databases (KDD-96), AAAI Press, Menlo Park CA, pp 351-4, 1996.

14. Srikant R, Agrawal R: Mining sequential patterns: generalizations and performance improvements. In: Proc Fifth Int Conf on Extending Database Technology (EDBT-96), Springer-Verlag, pp 3-17, 1996.

15. Fayyad U, Piatetsky-Shapiro G, Smyth P, p 41.

16. Siberschatz A, Tuzhilin A: On Subjective Measures of Interestingness in Knowledge Discovery. In: Proc 1st Intl Conf on Knowledge Discovery and Data Mining (KDD-95), AAAI Press, Menlo Park, CA, pp 275-81, 1995.

Jorge C. G. Ramirez is currently a Ph.D. candidate and Assistant Instructor in the Computer Science and Engineering Department at the University of Texas at Arlington, and is also a Research Fellow in the HIV Clinical Research Group at the University of Texas Southwestern Medical Center at Dallas. His research interests include software engineering, machine learning, knowledge discovery in databases, and intelligent systems in medicine. Mr. Ramirez received his B.S. from the Georgia Institute of Technology in 1982 and his M.S. from the University of Louisville in 1985. His current home page is http://www-cse.uta.edu/~ramirez/.

Diane Cook is currently an Associate Professor in the Computer Science and Engineering Department at the University of Texas at Arlington. Her research interests include artificial intelligence, machine learning, robotics, and machine planning. Dr. Cook received her B.S. from Wheaton College in 1985, and her M.S. and Ph.D. from the University of Illinois in 1987 and 1990, respectively. Her current home page is http://www-cse.uta.edu/~cook/.

Lynn Peterson is a Professor in the Computer Science and Engineering Department at the University of Texas at Arlington, and is Associate Dean of Engineering for Academic Affairs. She holds a Ph.D. in Mathematical Sciences (Medical Computer Science) from the University of Texas Southwestern Medical Center at Dallas. Her research interests include knowledge representation, natural language processing, artificial intelligence applications, intelligent computer-based instructional systems, and medical computer science. Her current home page is www-cse.uta.edu/~peterson/.

Dolores M. Peterson is currently an Associate Professor of Internal Medicine at the University of Texas Southwestern Medical Center at Dallas. She graduated from the UT Medical School in San Antonio in 1987 and completed her residency in Internal Medicine at Baylor Medical School in Houston in 1990. Her primary research area is HIV Disease and she heads the HIV Clinical Research Group. She has conducted many clinical trials concerning anti-retroviral treatment of HIV/AIDS and complications of HIV/AIDS and has an extensive publication record. Recent studies include an NIH trial directed at the immunology of HIV/AIDS when highly active anti-retroviral treatment is begun as well as an NIH study for collection of data on malignancies in HIV patients. The clinical trials she is conducting which are currently enrolling are listed on the GIM HIV Clinical Research Group web site at www.swmed.edu/home_pages/hivcrg/.