Discovery of Temporal Patterns in Sparse Course-of-Disease Data

Jorge C. G. Ramirez[1,2], Lynn L. Peterson[1], Diane J. Cook[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
5323 Harry Hines Blvd, Dallas, TX 75235-9103
Tel: 214-648-3246; Fax: 214-648-5415

Knowledge discovery in databases containing course-of-disease data for chronic illness is the thrust of the work detailed in this chapter. As seems to be typical of such databases, the data that is recorded is sparse and was collected with no concerted effort to maintain data quality. Despite this, we hypothesize that, given a database of clinical data for patients who had the same catastrophic or chronic illness, we can discover that subsets of these patients had a similar experience during the course of that disease. We use our experience in the development of the knowledge discovery system, Temporal Pattern Discovery System or TEMPADIS, to help understand the overall process of knowledge discovery in a medical database environment.

Keywords: temporal, course-of-disease, KDD, data cleaning, medical databases

1 Understanding the Problem Domain

1.1 Determination of objectives

The objective of the work is as 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 the data which show that groups of patients had similar experiences during the course of the disease. This work is motivated by the knowledge that analysis of the course of such diseases can enhance provision of care, prognosis, monitoring, outcomes research, cost/benefit analysis, and quality assurance. Figure 1 diagrams the project plan, the steps used in the development of TEMPADIS.

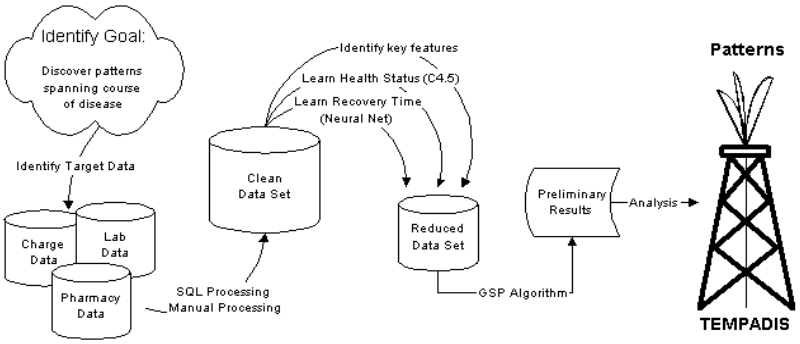

Fig. 1. Overview of development of TEMPADIS

2 Understanding the Data

2.1 Description of the data

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.

The types of data available for our analysis include binary, numeric, symbolic and text data. The data fields that are recorded for each patient vary greatly between collection instances, and also vary between patients even for similar medical events.

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. The significance of the temporal aspect, however, varies with the nature of the data. For example, for clinic or emergency room visits, or lab test results, the occurrence of an event and the order of occurrence relative to other events are important. For diagnoses or drug therapies, the duration of an event is just as important as the relative order of occurrence.

In summary, our task is to discover useful patterns in temporal, non-standard form, variable data-field, medical data.

2.2 Initial exploration of the data

The objective of our initial exploration of the data is to select a data set and to focus on a subset of variables. It is important to have enough data to contain significant patterns, and also to focus on as small a set of variables as will produce useful results in a reasonable amount of time.

We examine the database to characterize how long patients had been monitored and how many times they visited a medical facility. 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 had been seen for only a

month or less. On the other hand there are several hundred that had been monitored for seven years or more. We need 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 had 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 randomly select groups of patients from these 1,100 patients for the results shown in this chapter. The number of patients used for any given task is discussed along with that task.

From the mass of available data, it is important to focus on a subset of the variables that are related to the knowledge to be discovered in the KDD process. To help focus, we sought a dozen key variables, and 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 the data preparation phase of the KDD process, 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 (clinic visits, emergency room (ER) visits, and hospital stays). The second 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 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.

An examination of the laboratory test data reveals that although there are literally thousands of different medical laboratory tests, some tests are recorded significantly more often than others throughout the database. When looking at the pharmacy data we find that 71 different drugs had been dispensed that were specific to HIV and/or HIV-related illnesses. Selection of subsets of these data is addressed in the discussion of data preparation.

2.3 Verification of the quality of the data

When examining the database from a data quality perspective, we find that the database contains duplicate laboratory data for several different time periods, i.e., that particular data was in fact loaded into the database twice during those time periods. We also discover that there are 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 pharmacy records.

Data for a single patient consists of entries for each day in which some medical event took place, e.g., visiting a medical facility, filling a prescription, obtaining lab results. By nature that data is sparse, i.e., thinly scattered or distributed, with respect to the attributes recorded and with respect to time. It is therefore apparent to us that the nature of the data (e.g., its sparseness, the obvious errors) is such that a major part of the effort involved in this knowledge discovery process has to come in the next phase, the preparation of the data.

3 Preparation of the data

3.1 Data cleaning

The purpose of the data cleaning step is to remove noise from the data or to develop a method for accounting for that noise, to look at strategies for coping with the sparseness of the data, and to handle any other known changes that need to be made. In our case, some identified problems can be cleaned up as a group by processing the database using straightforward SQL statements. This can be done to correct obvious errors like the same misspelling and/or miscoding of a drug in the pharmacy data that appears multiple times. Removal of duplicate records requires a manual search of the data. 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 can be cleaned up with SQL statements are easy, we can correct them for the entire database. However, handling the manual problems in the entire database would be 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. A significant note to make at this point is that it required approximately 3 man-months to clean up those 400 patients' data. Our data cleaning was therefore limited to the correction obvious errors, and errors due to missing or incorrectly recorded data were not fixed.

3.2 Data selection

The purpose of the data selection step is to find useful features to represent the data, depending heavily 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 is to use only a minimum number of key variables that well represented the clinical relevance of the data.

First, we reexamine the lab test results. We focus on variables that are most likely to be recorded on any given encounter and are important indicators of the health status of an HIV-infected patient. To make this determination we consulted the clinicians of the HIV Clinical Research Group.

The six variables selected are White Blood Cell (WBC), Hematocrit (HCT), Platelets (PLT), CD4 Absolute (CD4A), CD4 Percent (CD4P), and Lymphocytes (LMPH). These selections were made based on the fact that the first five of those variables are values that the clinicians look at first when examining a patient's lab results. The sixth, LMPH, is included because CD4P is calculated from WBC, LMPH and CD4A, so if CD4P is missing but the others were present, which is true especially in the early years of data collection, CD4P can be calculated.

Choosing a subset of the pharmacy data is a more challenging task. Since we want to find patterns that represent similar experiences during the course of disease, the most obvious solution is 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/toxoplasmosis, Anti-mycobacterials, Anti-wasting syndrome, Anti-fungals and Chemotherapies. The clinicians agree that this tracking of whether or not a patient is on a drug in a particular category on a given day is sufficient information for our purposes. The data selection process therefore yields 18 variables, or events, as shown in Table 1.

Figure 2 shows an example of a portion of a patient file after abstracting the basis for these 18 variables (i.e., in the figure, the drug data has not yet been translated to the ten drug category variables; drugs are still listed by individual drug code). Note that in this particular example the column in between Event Type and WBC, where Event Severity should be, is blank for each day, meaning that all of these samples are normal clinic visits or hospital days.

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

Nature of the Event	Drug Categories
1. Event Type	9. Nucleoside Analogs
2. Event Severity	10. Protease Inhibitors
	11. Prophylaxis Therapies
Blood Tests	12. Intravenous 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

```
833 C  1.5 31.6  245  4.6    7 10  0
839 C  0.0  0.0    0  0.0    0  0  0
861 C  1.1 26.1  167  0.0    0 16  0
862    0.0  0.0    0  0.0    0  0  2 24: 30 38: 50
867 H  4.3 19.2  144  0.0    0 11  3  0:  3 22:  1 35:  2
868 H  2.2 26.2  144  0.0    0  5  3  0:  3 22:  1 35:  2
869    0.0  0.0    0  0.0    0  0  1 35: 60
874 C  1.3 32.4    0  0.0    0 17  0
889 C  1.1 30.4  154  0.0    0 36  0
890    0.0  0.0    0  0.0    0  0  3 22: 30 38: 50 39:480
923    0.0  0.0    0  0.0    0  0  1 39:480
933 H  3.6 20.4  182  0.0    0 11  3  0:  2 22:  1 39: 12
934 H  3.7 29.7  181  0.0    0  6  3  0:  3 22:  1 39: 16
935 H  1.6 27.9  186  0.0    0 11  3  0:  3 38:  2 39: 16
936 H  4.0 29.7  259  0.0    0  6  1  0:  3
937 H  2.7 24.1  246  0.0    0  9  1  0:  3
```

Fig. 2. Portion of abstracted patient record. From left to right the columns are: Day, Charge Event Type (C = Clinic Visit, H = Hospital Room, Blank = drug dispensed only), WBC, HCT, PLT, CD4P, CD4A, LMPH, Number of Drugs Dispensed, Drug Code: Number Dispensed (repeated as necessary)

3.3 Constructing and merging the data

From a clinical point of view, there is still important information, which should be considered, that is not contained explicitly in these 18 variables. First, knowing that the "normal" value of lab results depends to some extent on the patient, we had previously developed a methodology for normalizing all the values for each patient so that comparisons could more appropriately be made with other patient's data (Ramirez et al. 1998). Each variable is treated separately for each patient. Each is normalized to a range of integers from –4 to +4. In this range, 0 is considered normal, and both -4 and +4 are indicative of severe illness (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.

Second, the set of 18 variables does not explicitly contain a variable representing diagnosis. Diagnosis is the data element in the database that was collected in the least automated way. It 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 the recorded diagnosis data is rather incomplete. We therefore choose not to use the available diagnosis data for our discovery purposes, but instead develop another way to obtain equivalent information.

Based loosely on the way that the Centers for Disease Control classify the stages of HIV Disease, a new variable, health status, is added to the 18 previously selected variables to approximate the missing diagnosis data. This approach was recommended by the clinicians in the HIV Clinical Research Group. The five health status categories are shown in Table 2.

We use a machine learning technique called decision tree induction to learn rules for determining the health status value for any given patient on any given day. It is important to note that even though we use 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 of the patient, we use 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. 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 construct a set of data by randomly selecting four

days from each of 100 patients and listing all drugs being taken on those days. Three clinicians 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.

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 then use C4.5 (Quinlin 1993) to induce the decision tree. The inputs are the standard files required by C4.5. The names file contains the classes (which are 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 contains the 400 samples with the attribute values and the class. We run C4.5 in iterative mode using the default parameters, except that we specify a 95% confidence level on the pruned tree. The resulting tree is converted to rules that are used to determine health status values. The health status field can now be inserted in each day of each patient's data as part of the data reduction preprocessing.

Now that we have a measure of health status, which gives us an idea of the current state of the patient, we need a measure that gives 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 in a general way, but it gives no specific 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 between 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, again, this is not the type of information that appears in the database. Once again, we turn to a machine learning technique to inject information not already explicitly present in the database to enhance the discovery process.

Since neural nets have been used in a variety of ways in medical domains (Dombi et al. 1995; Frye et al. 1996; Izenberg et al. 1997; Mobley et al. 1995), including prediction of length of hospital stay, it is reasonable to assume we could use one to predict length of recovery time. The next task is to select the relevant inputs for determining some measure of recovery time. The inputs selected are shown in Table 3. An obvious first choice is our newly determined health status, which was developed to be a measure of the nature of the current illness.

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 select six days from each of 50 patients. We then abstract the data needed for the neural net inputs. Again three clinicians rated 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 (Goodman 1996), and its default 2/3-1/3 holdout, five sample cross validation. NevProp3 only allows for a single hidden layer. Within this 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. R2 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. This new knowledge, coupled with the health status, is then incorporated into the database for the purpose of giving 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 preparation step is the reduced data set. This data set consists of 20 variables deemed to well represent the larger set of all data available. Figure 3 shows the results of the transformation of the data from Fig. 2. It is this set of data with 20 variables that is used in the data mining step.

```
833 C  3 0 -3 -1  0 -4 -4 –3  1 0 2 0 0 0 0 0 0 0
839 C  3 1 -9 -9 -9 -9 -9 –9  0 0 1 0 0 0 0 0 0 0
861 C  3 1 -4 -3  0 -9 -9 –1  0 0 2 0 0 0 0 0 0 0
867 H  4 4  0 -4 -1 -9 -9 –2  0 0 2 0 0 0 1 1 0 0
868    4 1 -2 -3 -1 -9 -9 –4  0 0 2 0 0 0 1 1 0 0
874 C  4 3 -4 -1 -9 -9 -9  0  0 0 2 0 0 0 1 1 0 0
889 C  4 2 -4 -2 -1 -9 -9  2  0 0 2 0 0 0 1 1 0 0
933 H  4 4  0 -4  0 -9 -9 –2  0 0 1 0 0 0 0 2 0 0
934 H  4 2  0 -2  0 -9 -9 –4  0 0 1 0 0 0 0 2 0 0
935 H  4 2 -3 -3  0 -9 -9 –2  0 0 1 0 0 0 0 2 0 0
936 H  4 2  0 -2  1 -9 -9 –4  0 0 1 0 0 0 0 2 0 0
937 H  4 2 -2 -4  0 -9 -9 –3  0 0 1 0 0 0 0 2 0 0
```

Fig. 3. Portion of patient record after data preparation step. From left to right columns are: Day, Charge Event, Health Status, Recovery Time, six columns for normalized laboratory test results (-9 = not present) and ten columns for drug category data.

4 Data Mining

4.1 Selection of the data mining method

The next task is selection of 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.

We are trying to discover patterns in sequences of events across patients in a database. Patterns are represented in a form as shown in Fig. 4. 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.

```
< { (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) } >
```

Fig. 4. Typical pattern discovered by TEMPADIS

In our review of the literature, we found that there are only a few data mining methods relevant for our goal (Agrawal and Srikant 1995; Mannila et al. 1995; Mannila and Toivonen 1996; Padmanabhan and Tuzhilin 1996; Srikant and Agrawal 1996). We chose Srikant and Agrawal's General Sequential Patterns (GSP) Algorithm (Agrawal and Srikant 1995; Srikant and Agrawal 1996) 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 are using various support threshold levels from 5% (i.e., 1 of every 20 patients should support the pattern) to 33%. Those atomic events that survive are then combined as pairs, both as sequences and as concurrent occurrences. For example, if we have the atomic events (WBC 1) and (HCT 0) which are 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 that 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 can generate two candidate sequences of length three:

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

If both of those sequences are supported by the database, then we can 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 in Fig. 4 is of length 104, since it includes 104 atomic events, if you consider each drug type a separate event. This pattern would be 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.

4.2 Building and assessing the model

The evaluation of the model and data mining method selections 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. Our evaluation and the changes made as a result are explained in this section.

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 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 (Ramirez et al. 1998). 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.

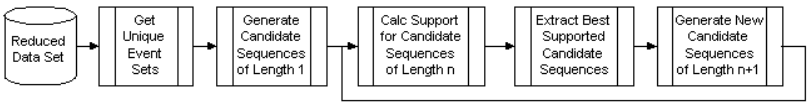

Fig. 5. Algorithm for TEMPADIS

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 Fig. 5 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. 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. 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.

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 the value in the particular database instance of -1 results in 100% match. The values of 0 or –2 result in a 67% match, and the values of 1 or –3 result in a 33% match. All other values (i.e., -4, 2, 3 and 4) result in a 0% match.

DegreeOfMatch then 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 actual threshold values used were determined by empirical results, such that the results yielded a higher percentage of interesting patterns and still found patterns at all.

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. The algorithm is exponential in the length of the discovered patterns, and within that it is linear in the number of patterns of that length and linear in the number of patients. 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.

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 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.

5 Evaluation of the discovered knowledge

5.1 Assessment of the results vs. the objectives.

At this point, we look at what was found and try to interpret it. This may again result in returning to a previous step to revise or fine-tune 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 Fig. 4, which was discovered by TEMPADIS. Her conclusion was:

> 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 shows only sporadic use of the other drug therapies. Her comments further reflect the relative flatness of all the variables 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.

Simply discovering patterns in the database is not an issue, since TEMPADIS has discovered many patterns in our domain database. Discovering interesting patterns is the issue. In order to facilitate this type of discovery, we have begun implementation of a query module based on a set of questions the clinicians posed. These questions describe the types of patterns they would find interesting. Using those questions as a guide, we implement search control techniques to avoid discovering patterns that we well know should be in the database, but are not particularly interesting except for the fact that they validate the TEMPADIS’s capability to discover valid patterns. However, these search control techniques still do not always bring about the desired results. It is true that even when we attempt to constrain the search, the results include many patterns that are not particularly interesting.

```
< { (EV C ) (HS 3) (WTR 0) (WBC -1) (PLT -1) (HCT 0)
    (CD4P -4) (CD4A -4) (LMPH 0) (onD 0010000000) }
  { (EV C ) (HS 3) (WTR 1) (WBC -2) (PLT 0) (HCT -1)
    (CD4P -4) (CD4A -4) (LMPH 0) (onD 1010000000) }
  { (EV C ) (HS 3) (WTR 1) (WBC 0) (PLT 0) (HCT 0)
    (CD4P 0) (CD4A 0) (LMPH 1) (onD 1010000000) }
  { (EV C ) (HS 3) (WTR 1) (WBC 0) (PLT 0) (HCT 0)
    (CD4P -2) (CD4A 0) (LMPH 1) (onD 1000000000) }
  { (EV C ) (HS 3) (WTR 1) (WBC -1) (PLT -1) (HCT 0)
    (LMPH 0) (onD 0000000)}
  { (EV C ) (HS 3) (WTR 1) (WBC -1) (PLT -1) (HCT 0)
    (CD4P 0) (CD4A 0) (LMPH 0) (onD 1000000000) } >
```

Fig. 6. Discovered pattern with search control set for non-decreasing CD4A.

On the other hand, among the mass of discovered patterns are some that show that our original goal is met. TEMPADIS is able to discover useful concepts in our domain database. In Fig. 6, we see a pattern discovered when the search control was set for non-decreasing CD4A. This pattern is interesting for the fact that it is one of the few discovered that shows a group of patients that consistently maintained their drug therapy, and the result was the improvement in CD4A from –4 to 0 sustained over a period of time.

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

Fig. 7. Discovered pattern with search set to unchanged CD4A.

Figure 7 shows a pattern discovered while searching for unchanged CD4A. The intention of this query was to discover what other events were doing during a time period of stable CD4. Though the pattern seems unremarkable, it represents a group of patients that were not on any drug

therapy and shows that the patients' condition seems to gradually be deteriorating due to a general trend of increased recovery time, despite the stable CD4.

5.2 Reviewing the entire knowledge discovery process

Having completed the KDD process for our database with our goal, we can evaluate the results. Fayyad, Piatetsky-Shapiro, and Smyth (Fayyad et al. 1996) provide some criteria for evaluating discovered patterns. They say that 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 the concept of interestingness (Silbershatz and Tuzhilin 1996) 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 patterns in Figs. 4, 6 and 7 are understandable. In fact, they seem to represent specifically recognizable conditions. 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 be identified by a pattern, those groups can be investigated for the purposes that the user originally had for specifying parameters that would yield patterns of that type.

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 search 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.

6 Using the discovered knowledge

6.1 Implementation and monitoring plans

Implementation depends on actions out of the control of this research group: acquisition of hardware, acquisition of database software, finding time for research nurses to supervise and insure quality of data entry, among others.

6.2 Overview of the entire project for future use and improvements

Our overview of the project provides some lessons learned which can be passed on for future use by this group and others. We simply need to recognize the nature of medical data. It may have been stored for reasons (e.g., billing) that had nothing to do with the use in data mining to which it is currently being put. While it may serve its original function well, it certainly does not lend itself well to anything but simple analyses of a statistical nature. In contrast to medical imaging data or data collected with the *a priori* intent to do statistical analysis, this is typical of data sets from realistic domains collected for other purposes which are rarely "neat and tidy". Furthermore, this data was collected by a large number of people without concerted efforts to maintain consistency. A strong message that our experience sends, therefore, is that one should not minimize the need for data preparation, and the time and resources that this preparation will take.

One of the features of TEMPADIS that should be highlighted in this summary is the increased power that was gained using inexact discovery methods. Exact match will work for many domains, but should be questioned in the medical domain.

Our experience also shows the need to focus the search in the data mining step, to know what is being sought from the data and not simply to let data mining methods look in a database for patterns. Combinatorial explosion occurs easily with undirected search methods. This emphasizes the need to work closely with a domain expert who can focus the search based on the meaning of the data and the purpose of the search. As usual, the partnership between the computer scientist and the medical expert is key to a successful effort in medical knowledge discovery.

References

Agrawal R. and Srikant R. 1995. Mining sequential patterns. In: *Proc 11th Int Conf on Data Engineering*, 3-14

Dombi G.W., Nandi P., Saxe J.M., Ledgerwood A.M., and Lucas C.E. 1995. Prediction of rib fracture injury outcomes by an artificial neural network. *J of Trauma: Injury, Infection, and Critical Care*, 39(5):915-21

Fayyad U., Piatetsky-Shapiro G., Smyth P. 1996. From data mining to knowledge discovery in databases. *AI Magazine*, (Fall):37-53, AAAI, Menlo Park CA

Frye K.E., Izenberg S.D., Williams M.D., Luterman A. 1996. Simulated biologic intelligence used to predict length of stay and survival of burns. *J of Burn Care and Rehab*, 17(6):540-6

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

Izenberg S.D., Williams M.D. and Luterman A. 1997. Prediction of trauma mortality using a neural network. *American Surgeon* 63(3):275-81

Mannila H., Toivonen H., Verkamo A.I. 1995. Discovering frequent episodes in sequences. In: *Proc 1st Int Conf on Knowledge Discovery in Databases*, AAAI Press, 210-15

Mannila H. and Toivonen H. 1996. Discovering generalized episodes using minimal occurrences. In: *Proc 2nd Int Conf on Knowledge Discovery in Databases*, AAAI Press, 146-51

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

Padmanabhan B., Tuzhilin A. Pattern discovery in temporal databases: a temporal logic approach. In: *Proc 2nd Int Conf on Knowledge Discovery in Databases*, AAAI Press, 351-4

Quinlin J.R. 1993. C4.5: programs for machine learning. Morgan Kaufmann

Ramirez J.C.G., Peterson L.L., and Peterson D.M. 1998. A sequence building approach to pattern discovery in medical data. In: Cook, DJ (Ed): *Proc 11th Int Florida Artificial Intelligence Research Symp Conf*, AAAI Press, 188-192

Silberschatz A. and Tuzhilin A. 1995. On Subjective Measures of Interestingness in Knowledge Discovery. In: *Proc 1st Intl Conf on Knowledge Discovery and Data Mining*, AAAI Press, 275-81

Srikant R. and Agrawal R. 1996. Mining sequential patterns: generalizations and performance improvements. In: *Proc 5th Int Conf on Extending Database Technology*, Springer-Verlag, 3-17