Spatio-Temporal Data Mining with Expected Distribution Domain Generalization Graphs

Howard J. Hamilton, Liqiang Geng, Leah Findlater, and Dee Jay Randall
Department of Computer Science, University of Regina, Regina, SK, Canada S4S 0A2
{hamilton, gengl}@cs.uregina.ca

Abstract

We describe a method for spatio-temporal data mining based on expected distribution domain generalization (ExGen) graphs. Using familiar calendar and geographical concepts, such as workdays, weeks, climatic regions, and countries, spatio-temporal data can be aggregated into summaries in many ways. We automatically search for a summary with a distribution that is anomalous, i.e., far from user expectations. We repeatedly ranking possible summaries according to current expectations, and then allow the user to adjust these expectations.

1. INTRODUCTION

Recently, spatio-temporal data mining has been identified as a distinct research area concerned with knowledge discovery from datasets containing explicit or implicit temporal, spatial or spatio-temporal information [24]. People have rich background knowledge about time and space, including vivid knowledge of multiple ways that time and space values can be aggregated. A ***calendar attribute*** has a domain consisting of date and time values, such as birth dates or check out times [22], and a ***geospatial attribute*** has a domain consisting of Earth-based location values, such as geographic coordinates or city names. Strong effects on patterns in data may occur because calendar values reflect strong physical constraints (e.g., Earth's rotation and revolution around the Sun) and strong cultural constraints (e.g., month-end paydays, Christmas shopping). Similarly, geospatial values also reflect strong physical constraints (e.g., proximity, weather) and strong cultural constraints (e.g., political regions, urban versus rural).

In this paper, we describe a summarization method suited to mining spatio-temporal data. It is applicable to other types of data, but it is illustrated using calendar and geospatial attributes, because they best justify it and demonstrate its utility. By ***summarization***, we refer to the formation of interesting, compact descriptions of data. Summarization was listed by Fayyad et al. as one of the six primary data mining tasks [10]. Early work on attribute-oriented induction [6], function finding [25], multivariate visualization techniques, and derivation of summary rules provided diverse approaches to summarization Online analytical processing (OLAP), data cubes with rollup and drilldown operators, and the *rollup* and *cube by* operators in SQL all address the task of summarization.

We have found three chief weaknesses in the previously described approaches to summarization. It is common to create numerous summaries, all valid, for the same data. Unfortunately, it is tedious and time consuming for the data analyst to have to examine these summaries one by one to assess them. Secondly, incorporating domain knowledge into the summarization process is not facilitated by some of these approaches. Attribute-oriented induction does allow background knowledge to be incorporated in the form of a concept hierarchy for each attribute, but it does not provide a means of specifying multiple ways of aggregating the values for an attribute during a single data mining task. Lastly, these methods provide limited scope to handle the changes in the user's knowledge that naturally occur during knowledge discovery. Consider an example. Suppose that it is discovered and reported to the user that purchasers in Canadian cities buy fewer basketball-related goods than those in US cities, after analysing information from North American sporting goods stores. Then, it will be subsequently less interesting to discover that purchasers in Saskatchewan (a province in Canada) buy fewer basketball-related goods than those in other parts of North America. In the application of our DGG-Discover software to a wide variety of commercial and institutional databases over the past five years, we have found that these weaknesses are most manifest for temporal and spatial attributes because of the detailed domain knowledge that people have about these domains, as

Proceedings of the 10th International Symposium on Temporal Representation and Reasoning and
Fourth International Conference on Temporal Logic (TIME-ICTL'03) 1530-1311/03 $17.00 © 2003 IEEE

described at the beginning of this section.

Our summarization method is based on a combination of expected distributions and domain generalization graphs. A ***domain generalization graph*** (DGG) [14, 16, 17] is a graphical structure that can be used both as a navigational aid by the user and as a guide to heuristic data mining procedures. Each path in the graph corresponds to a generalization consistent with the specified generalization relations. An ***expected distribution domain generalization (or ExGen) graph*** is a DGG where each node has been augmented with a description of the expected distribution of the values in the corresponding domain. Initial and changing domain knowledge is described by ExGen graphs. For a calendar attribute, we created a standard ExGen graph that describes all well-known temporal relationships, including the number of days in a week, month, and year, as well as the seasons and leap years. To customize the calendar ExGen graph for a particular application, additional information about workdays, financial quarters, academic terms, etc. is added. During the knowledge discovery process, the expectations at a particular node, which reflect a user's knowledge about the corresponding domain, can be updated. As well, expectations can be propagated to other nodes in the graph; for example, if the user revises the expectation about the fraction of movies watched in the evening, then the expectation about the fraction of movies watched from 8:00 pm to 9:00 pm can be automatically adjusted.

For a calendar attribute, our approach has five steps. First, a domain generalization graph for a calendar attribute is created by explicitly identifying the domains appropriate to the relevant levels of temporal granularity and the mappings between the values in these domains. Second, a probability distribution is associated with each node in the graph. Third, the data are aggregated in all possible ways consistent with this graph. Aggregation is performed by transforming values in one domain to another, according to the directed arcs in the domain generalization graph. Each aggregation is called a ***summary***. Fourth, the summaries are ranked according to their distance from the expected distribution for the appropriate domain, using a diversity-based interestingness measure [15, 16]. Fifth, the highest ranked summaries are displayed. Expectations are then adjusted and steps repeated as necessary.

Our work can be contrasted with recent research on the connection between multiple temporal granularities and data mining. Granularity factors that affect data mining were described by Andrusiewicz and Orlowska [1]. Bettini et al. provide conventions similar to those given here for naming the multiple temporal granularities, although they do not provide a data structure similar to DGGs for representing the relationships between these granularities or for recording or manipulating expectations [5]. They define a *calendar algebra* to represent granularities of calendar data and the relationships between those granularities. Specified *calendar operations* are used to create new granularities, either by recursively grouping data from an existing granularity or by filtering data from a granularity. They apply their system in the context of data mining by looking for frequent event sequences that have a specified minimum confidence at some level of temporal granularity. Bertino et al. build on the work of Bettini et al. to specify the syntax and semantics of expressions involving data with multiple temporal granularities [4]. Combi et al. use a much simpler structure than our calendar DGG with an ordered granularity from SUP (top) to year, month, day, hour, minute, second, INF (bottom) [8,12]. However, they cannot handle expectations or phenomenon such as weeks.

In data mining, work on contrast sets has identified pairs of values that lead to significantly different outcomes in the context of particular combinations of values [3], but this approach does not allow known generalization relations among attributes or user expectations to be incorporated. Work on temporal data mining has emphasized searching for recurring patterns in time series [2]; our method is not restricted to time series. Adding temporal semantics to association rules has also attracted attention [6,19,20,21]. Rainsford and Roddick provide a method based on structured relationships among temporal relations, rather than our structure among domains [21]. Li et al. build on the Apriori algorithm for mining association rules to include temporal semantics [20].

The remainder of this paper is organized as follows. In the following section, we review domain generalization graphs and present a particular graph for a calendar attribute. In Section 3, we describe ExGen graphs and explain how expected distributions are attached and propagated. In Section 4, we briefly describe our methodology and illustrate its application to two data sets. Finally, in Section 5, we present our conclusions.

2. GENERALIZING CALENDAR DATA WITH DOMAIN GENERALIZATION GRAPHS

In this section, we describe domain generalization graphs and explain how they can be used to represent domain knowledge relevant to calendar attributes. The approach is also relevant to knowledge about geospatial attributes, but we emphasize temporal data, in keeping with the symposium theme of temporal representation and reasoning.

Informally, a DGG can be thought of as a graph showing possible generalizations as paths through a

graph. Formally, a domain generalization graph is defined in terms of a generalization relation (adapted from [16,17]). Given a set $X = \{x_1, x_2, ..., x_n\}$ representing the base-level domain of some attribute and a set $P = \{P_1, P_2, ..., P_m\}$ of partitions of the set S, we define a nonempty binary relation $\preceq$ (called a ***generalization relation***) on P, where we say $P_i \preceq P_j$ if for every section $S_a \in P_i$, there exists a section $S_b \in P_j$, such that $S_a \subseteq S_b$. The generalization relation $\preceq$ is a partial order relation. If $P_i \preceq P_j$, for each section $S_b \in P_j$, there exists a set of sections $\{S_{a_1}, ..., S_{a_k}\} \subseteq P_i$, denoted $Spec(S_b, P_i)$, such that $S_b = \bigcup_{i=1}^{k} S_{a_i}$.

A ***domain generalization graph*** (*DGG*) $G = \langle P, E \rangle$ is constructed based on a generalization relation $\langle P, \preceq \rangle$ as follows. The nodes of the graph are the elements of P. There is a directed arc from P_i to P_j iff $P_i \neq P_j$, $P_i \preceq P_j$, and there is no $P_k \in P$ such that $P_i \preceq P_k$ and $P_k \preceq P_j$. Each node corresponds to a domain of values. Each arc corresponds to a generalization relation, which is a mapping from the values in the domain of the initial (or parent) node to that of the final node (or child) of the arc. The ***bottom*** (or source) node of the graph corresponds to the original domain of values X and the ***top*** (or sink) node T corresponds to the most general domain of values, which contains only the value ANY.

Figure 1 shows part of a DGG for a calendar attribute (simplified from [22]). The node labelled *YYYYMMDDhhmm* represents the most specific domain considered, i.e., the finest granularity of our calendar domain is one minute. Higher-level nodes represent generalizations of this domain. To handle data with calendar values specified to finer granularity, e.g., seconds, more specific nodes could be added to the DGG.

The calendar DGG can be used to guide the generalization of calendar data into higher-level concepts. For example, we generalize from *YYYYMMDD* to *YYYYMM* by removing the *DD* information from the calendar attribute. When a new representation is required in the calendar domain, this DGG can be extended by adding new nodes and arcs and by defining new generalization relations associated with the arcs.

Four types of generalization relations are associated with the arcs in a calendar DGG: granularity, subset, lookup, and algorithmic. For ***granularity generalization***, we assume that *YYYYMMDDhhmm* can be represented as five subattributes (*YYYY*, *MM*, *DD*, *hh*, *mm*). We generalize by suppressing subattributes from least significant (*mm*) to most (*YYYY*). All four domains this creates are shown in Figure 1. For example, granularity generalization could be used to generalize from *YYYYMMDDhhmm* to *YYYYMMDD*. ***Subset generalization***, which includes granularity generalization as a special case, discards any combination of subattributes. The remaining subattributes need not be adjacent. For example, we could generalize from *YYYYMMDDhhmm* to *MMhh*. Figure 1 shows only a few of the domains that can be created by subset generalization. ***Lookup generalization*** uses a lookup table to generalize from lower-level concepts, such as *DayOfWeek*, to higher level concepts, such as W*eekdayOrWeekend (WDWE)*. ***Algorithmic generalization*** uses an algorithm to generalize. It allows convenient calculation of regular relationships, such as those required to determine the number of days in a year and the beginnings of seasons. Although algorithmic generalization subsumes the three previous types, we find the distinction useful and only refer to algorithmic generalization when no other is applicable.

3. EXGEN GRAPHS

An ***expected distribution domain generalization* (*ExGen*) *graph*** is a DGG that has a probability distribution associated with every node [11]. Each distribution represents the expected probability of occurrence of the values in the domain corresponding to the node. For a node (i.e., partition) $P_j = \{S_1, \ldots, S_k\}$, we have $0 \leq \Pr(S_i) \leq 1$ and $\sum_{i=1}^{k} \Pr(S_i) = 1$, where $\Pr(S_i)$ denotes the probability of occurrence of a value $S_i \in P_j$, i.e., the *i*th section in node P_j. Each probability distribution represents the user's expectation for the frequency of occurrence of the values in the domain corresponding to the node. For example, if the domain is the names of countries of the world and expectations are based on population, the distribution could be specified by giving each country's name associated with the ratio of that country's population to the world population. Such a distribution is most often called a ***prior*** in statistics.

The simplest approach is to assume uniform distribution for all domains. Unfortunately, this approach may suggest inconsistent distributions. As a simple example, uniform distribution over the *WeekdayName* domain (1/7 for each day) is inconsistent with uniform distribution over the *WDWE* domain (1/2 for the value Weekday and 1/2 for the value Weekend). Two days, Saturday and Sunday, with a total expectation of 2/7 are generalized to Weekend, with a total expectation of 1/2, which is inconsistent.

To define consistency, assume node Q is a parent of node R in an ExGen graph, and therefore for each section $S_b \in R$, there exists a set of (more specialized) sections

$Spec(S_b, Q) = \{ S_{a_1}, ..., S_{a_k} \} \subseteq Q$, such that $S_b = \bigcup_{i=1}^{k} S_{a_i}$.

If for all $S_b \in R$, $\Pr(S_b) = \sum_{i=1}^{k} \Pr(S_{a_i})$, we say that Q and R are ***consistent***. In an ExGen graph, we say that node R is ***bottom-consistent***, i.e., consistent with the bottom node X, if for all $S_i \in R$, $\Pr(S_i) = \sum_{x \in S_i} \Pr(x)$. We say an ExGen graph G is ***consistent*** if all pairs of adjacent nodes in G are consistent.

Theorem 1 [11]: An ExGen graph G is consistent iff every node in G is bottom-consistent.

Proof sketch: Proof follows by induction based on the distance from the bottom node and the expectations for each section S_b at node R equalling the sum of the expectations of the more specific values at node Q that correspond to this section, namely $Spec(S_b, Q)$.

To avoid inconsistencies and simplify the process of specifying expectations, a distribution can be specified for the bottom node, and then propagated upward to all nodes (***bottom-up propagation***). Or a distribution can be associated with a single node in the ExGen graph and then using an assumption of a uniform (or other) distribution among the values in each section, it can be propagated to the bottom node of the graph, and bottom-up from there.

Theorem 2 [11]: If an ExGen graph G is constructed by bottom-up propagation from a DGG D and a distribution E for the values in X, the bottom node of the graph, then graph G is consistent.

Proof sketch: Proof follows by induction based on the distance from the bottom node, as with Theorem 1.

For example, if the logins to a system are expected to be uniformly distributed among all days in a three week period, then propagating upward gives a uniform distribution for the seven members of *WeekdayName*, and propagating further upwards gives a {2/7,5/7} = {0.29, 0.71} distribution for node *WDWE* = {Weekday,

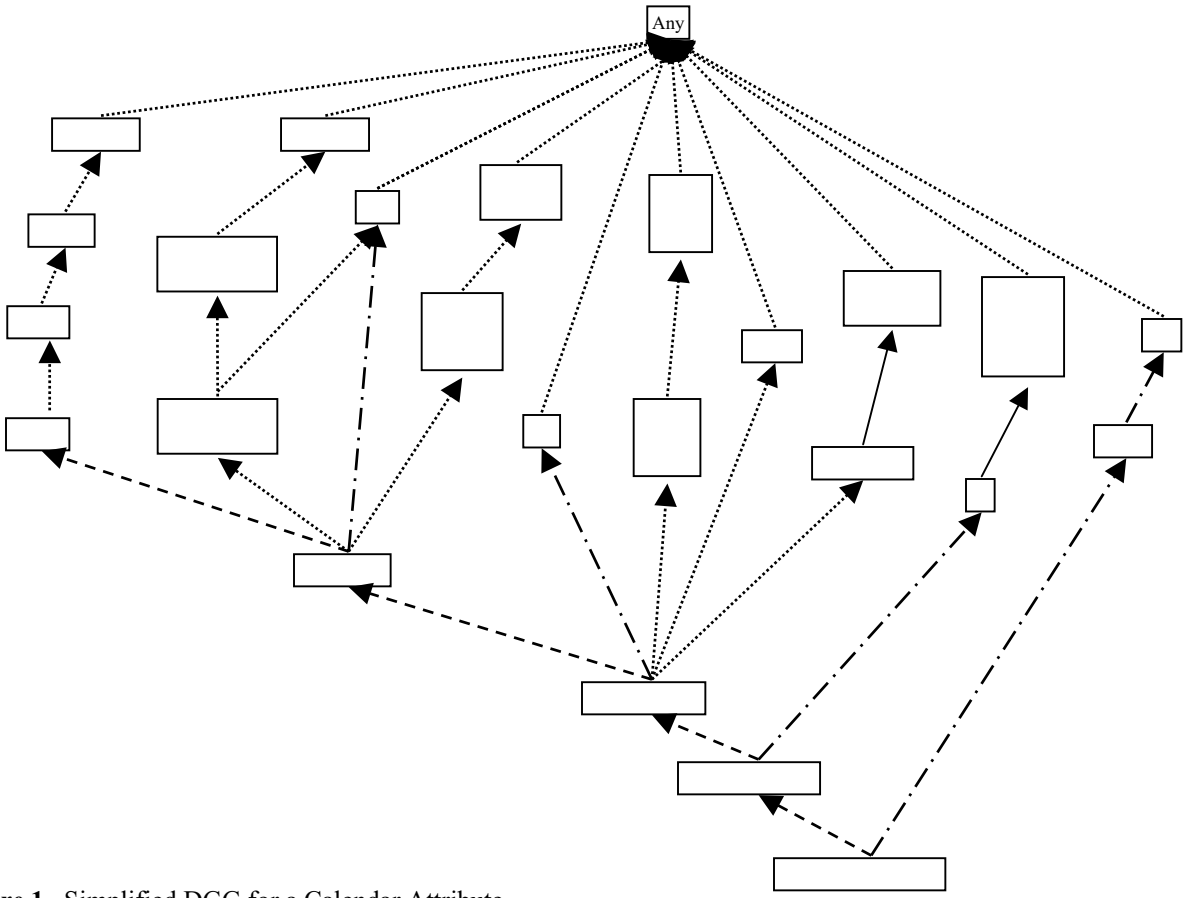


Figure 1. Simplified DGG for a Calendar Attribute

Weekend}. (We use two significant digits in this paper, but double precision in our implementation.)

If distributions are separately specified for two nodes and then propagated downward, they can suggest inconsistent distributions for a particular node in the ExGen graph. This inconsistency can happen when values are propagated down from multiple children to a single parent, consistent values cannot be calculated without additional information. We have proposed a strategy based on optimization to resolve such inconsistencies in [11], but we do not discuss this problem further here.

When several attributes have ExGen graphs, aggregation is performed to all nodes in the cross product of the ExGen graphs. By the ***cross product of the ExGen graphs***, we mean every combination of nodes from the ExGen graphs where one node is taken from each ExGen graph. The expectation of a combination of values (or ***generalized tuple***) $t = (d_1, d_2, \ldots, d_m)$ is by default the product of the expectations of the values of its component attributes; i.e., $e(t) = e(d_1)\, e(d_2) \ldots e(d_m)$. (If an expectation has been specified for a subset of the attributes, as a joint-probability distribution, then value is derived from this distribution instead of the product of the expectations.)

During the data mining process, an output summary can be produced for every combination of nodes in the ExGen graphs where one node is taken from each ExGen graph. A summary is produced as a file of comma-separated values, which can be readily displayed and processed with Microsoft Excel and other standard tools. Given a set of summaries, an interest measure assigns a numeric score to each. These scores can be used to rank the results and determine which generalized relations are consistent with an expectation and which ones conflict with it. An interest measure is computed by comparing an observed distribution to an expected distribution.

4. METHOD AND SAMPLE APPLICATIONS

We now describe the data mining technique we implemented in our DGG-Discover 5.0 software. For clarity, the method is first briefly explained and then its application to two datasets is described. The first dataset concerns weather in the province of Saskatchewan, Canada, and the second concerns logins to a shared computer system called Hercules at the University of Regina.

Station	Station Name	Lat	Long	Date	High Temp	Total Precip
4012400	Estevan A	49.217	102.967	12/27/1944	-12.8	0
4063560	Island Falls	55.533	102.350	12/27/1944	-22.2	0
4016560	Regina A	50.433	104.667	12/27/1944	-14.4	0
4018160	Tugaske	50.883	106.300	12/27/1944	-20.6	0
4048520	Waseca	53.133	109.400	12/27/1944	-18.9	0
4019080	Yorkton	51.267	102.467	12/27/1944	-18.9	0

Table 1. Sample Weather Data.

With DGG-Discover 5.0, the following steps are followed:

1. Create the domain generalization graphs, as described in Section 2, corresponding to user knowledge of the domains.
2. Associate a probability distribution with one node in each graph, and propagate this distribution in a consistent manner, as described in Section 3.
3. Generalize the data to create a set of summaries, each one consistent with the original data [14, 16].
4. Rank the summaries based on their distance from expectations, and examine the highly ranked ones [15,16].
5. Adjust expectations.
6. Repeat Steps 4 and 5 until no adjustments can be made, or the user terminates the process.We now describe how these steps were applied to the weather data.

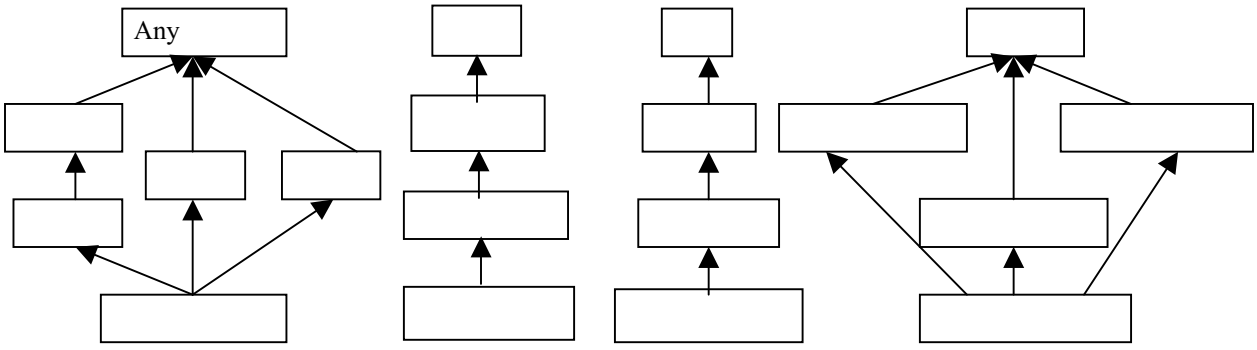
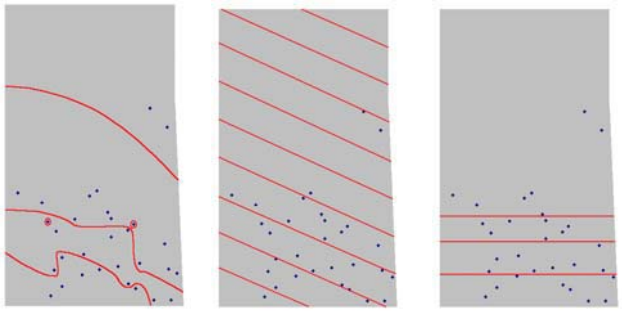


Figure 2. DGGs for *Date*, *Temp*, *Precip*, and *Station*

Our goal was to assess whether this discovery methodology could guide exploration of the data. We used the daily high temperature (in 0.1 degree Celcius) and daily total precipitation (in mm, with snow converted to equivalent water) for all weather stations for all days from January 1, 1900 to December 31, 1949. The number of daily weather observations (tuples) was 211,534. Example data is given in Table 1, where *StationName* and *Lat* (latitude) and *Long* (Longitude) depend on the *Station* attribute. This data was previously described in an invited paper [13].

Creating Domain Generalization Graphs: The attributes we used in our experiment are *Station, Date*, *Daily High Temperature* (abbreviated *Temp*), and *TotalPrecip* (abbreviated *Precip*). For *Date*, we used the simplified version of Figure 1 shown in Figure 2(a). This DGG indicates that a particular date (*YYYYMMDD*) can be generalized to a year, a month, or a season. As well, years can be generalized to decades. For the *Temp*, *Precip*, and *Station* attributes, we used the DGGs shown in Figure 2(b), 2(c), and 2(d), respectively. The three paths in the Station DGG correspond to the three maps shown in Figure 3.



(a) CRegion (b) DRegion (c) LRegion

Figure 3. Maps Corresponding to the Interior DGG nodes for the Station Attribute

Figure 3(a) shows the stations clustered using the *k*-means algorithm with $k = 6$ (other values of *k* gave less plausible maps). Figure 3(b) shows ten regions defined based on the adjusted distance *d* to any point (Lat, Long) from the southwest corner of the province, which is at (49°N, 110°E), using the formula

$$d = (\text{Lat} - 49) + 0.35\,(110 - \text{Long})$$

This assumes that similar weather occurs along a slanted line across the province. The 0.35 is an arbitrary constant defined based on a map of the province showing ecological regions [9]. Since the northeast corner of the province is at (60°N, 102°E), the values for *d* ranged from 0 to 13.8. To create the DistanceRegion (abbreviated DRegion) node, we divided this range into 10 equal-sized intervals. Figure 3(c) shows the stations grouped from south to north into four regions (South, LowMid, HighMid, and North) to create the LatitudeRegion (abbreviated LRegion) node.

Associating expectations: We assumed initial distribution for some nodes in the DGGs For the *Date* attribute, we assumed a uniform distribution at the *Year* (*YYYY*) node, i.e., we assumed an equal number of observations from each year 1900-1949. For *Station*, we assumed a uniform distribution at each weather station (*StationSpecific* node). For *Temp*, we assumed a uniform distribution at the *TempRange* node, which means that cold days, cool days, warm days, and hot days each have a probability of 0.25. Similarly, for *Precip*, we assumed a uniform distribution at the *PrecRange* node, which means that the numbers of days are assumed to be equal for these ranges: no precipitation, very low (0, 50mm], low (50, 100], medium (100, 400], and high (> 400 mm). A crucial property of our method is that it is self-correcting: if inappropriate initial distributions are assumed, the user is quickly informed of them, and through adjustment and automatic propagation, they are easily improved.

Generalize the Data: Next the weather data are aggregated in all possible ways consistent with the domain generalization graphs. Since the *Date*, *Temp*, *Precip*, and *Station* DGGs have 6, 4, 4, and 5 nodes, respectively, the number of aggregations or summaries produced is 6(4)(4)(5) = 480. Each summary is evaluated with an interestingness measure and also optionally stored as a Comma-Separated-Value (CSV) file.

Rank the Summaries: The summaries are ranked according to their distance from the expected distribution using a selected interestingness measure.

A list of the *k* highest ranking summaries is displayed; Table 2 shows the top 10 summaries for this example. Also, the highest ranked summary, which corresponds to the (Any, Any, Any, PrecSplit) node combination, is automatically displayed as an Excel graph by our interface. This two-line summary shows that, when *Station*, *Date*, and *Temp* are ignored (set to Any), the percentage of daily observations in the data with rain or other precipitation is 21%, and the number without is 79%. Since the original expectation was 80%/20%, the actual distribution is far from the expected one, and according to the variance measure, farthest from the expectation of any node combination.

Adjust Expectations: At this point, the user has learned something about the domain, and the expectations can be adjusted according to the acquired knowledge. To continue this example, we assume that the distribution is simply accepted for the PrecipSplit node and propagated to the PrecipRange and PrecipSpecific nodes in the *Precip* ExGen. This assumption follows in a straightforward fashion from the results, and can be readily automated. The effect on further data mining corresponds to saying: "I accept that only 21% of the days have precipitation; now, don't tell me about that again or about any logical consequence of that."

Station	Date	Temp	Precip	Variance
Any	Any	Any	PrecSplit	0.688432
Any	Any	TempSplit	PrecSplit	0.116666
Any	Any	Any	PrecRange	0.110478
LRegion	Any	Any	PrecSplit	0.026357
Any	Any	TempRange	PrecSplit	0.025587
Any	Any	TempSplit	PrecRange	0.025189
Any	Season	Any	PrecSplit	0.024728
CRegion	Any	Any	PrecSplit	0.020763
DRegion	Any	Any	PrecSplit	0.020292
Any	Decade	Any	PrecSplit	0.018175

Table 2. Top-Ranked Summaries After Run 1

Station	Date	Temp	Precip	Variance
Any	Decade	Any	Any	0.010509
Any	Any	TempSplit	Any	0.008428
Any	Season	TempSplit	Any	0.006702
LRegion	Any	Any	Any	0.006344
CRegion	Any	Any	Any	0.005890
Any	Any	TempRange	Any	0.003774
Any	Any	Any	PrecRange	0.002911
Any	Decade	Any	PrecSplit	0.002876
Any	Decade	TempSplit	Any	0.002542
DRegion	Any	Any	Any	0.002389

Table 3. Top-Ranked Summaries After Run 2

Continue Data Mining: Using the revised expectations, the ten highest ranked summaries are shown in Table 3. Most summaries with PrecipSplit or PrecipRange have disappeared from the top ten list because of the change in expectations. The (Any, Any, Any, PrecRange) and (Any, Decade, Any, PrecSplit) summaries remain in the top ten. However, the rank of the (Any, Any, Any, PrecRange) summary changed from 3 to 7 and its interestingness measure decreased from 0.110478 to 0.002911. Although the (Any, Decade, Any, PrecSplit) summary has a higher rank in the new table (8 instead of 10), its interestingness measure decreased from 0.018175 to 0.002876. This suggests that through propagation from the PrecSplit node, the distribution for other nodes in the Precip ExGen has been adjusted appropriately.

The highest ranked summary in Table 3 tells us that the expectation for decade node is far from the observed one. Among the five decades from 1900 to 1949, the percentages of observations from the decades in order are: 7%, 13%, 21%, 25%, and 34%. Again, the user can simply accept this, or explore deeper to understand why. The actual reason was that only a few weather stations existed in 1900 and others were gradually added. This relationship is best addressed by creating a joint distribution (discussed further below) between Station and Date (at the Year or YYYYMMDD node). For this example, we will assume that the user simply accepts the observed distribution as the expected distribution for Decade. This distribution was propagated downward to Year.

Tables 4 to 7 list the ten most interesting summaries for runs 3 to 6, respectively.

When the top-ranked summary has more than one domain value that is not "Any", the summary corresponds to a ***joint probability distribution*** (or ***joint expectation***). For example, after Run 4, as shown in Table 5, the top-ranked summary is (Any, Season, TempSplit, Any), which means that the proportion of low and high temperature days varies with the season. To accept this distribution, a joint expectation between Season and TempSplit is created. All subsequent runs will consult this table whenever an expectation for the combination of Season and TempSplit needs to be calculated. After propagation of the joint expectation and calculation of the interestingness of the summaries, as shown in Table 6, we can see that not only has (Any, Season, TempSplit, Any) disappeared from the top ten summaries, but also closely related nodes (Any, Season, TempSplit, PrecSplit) and (Any, Season, TempRange, Any) have become less interesting and consequently disappeared as well. It appears that variance due to date is mostly captured by the discovered knowledge about the distribution among the decades and the distribution among the season-temperature combinations.

The results that we obtain from the system are an ordered list of summaries that have high interestingness values. Within each summary is a list of expected and observed probability distributions. If the variance of a record in the summary is greater than a threshold, we will highlight this record and provide it to the user. If the observed probability is higher than the expectation, we say that this record is more likely than what the user expected and highlight it in red (light pink). Otherwise, we say it is less likely and highlight it in blue. For example, if the "summer, hot" record in the (Any, Season, TempRange, Any) summary has observed and expected probabilities of 0.2 and 0.1, respectively, we highlight it in red, which means that "hot, summer" records occurred more frequently than expected.

The knowledge discovery process continued in the same manner for some time. To illustrate the type of knowledge being found, in Table 8, we list one relationship from each of the first 6 runs. Although we originally expected to learn about the weather, we also learned a substantial amount about the particular dataset, including that summer readings were taken more faithfully than winter ones. All relationships reported were statistically significant at the 0.05 significance level.

The algorithm used to produce the English-language summaries is given in simplified form in Figure 4. For each of the top k nodes, it examines the generalized records in the corresponding summary. If the record is significantly different than expected, it is marked for output. Records are ordered for output by variance.

Station	Date	Temp	Precip	Variance
Any	Any	TempSplit	Any	0.008428
Any	Season	TempSplit	Any	0.006702
LRegion	Any	Any	Any	0.006344
CRegion	Any	Any	Any	0.005890
Any	Any	TempRange	Any	0.003774
Any	Any	Any	PrecRange	0.002911
DRegion	Any	Any	Any	0.002389
Any	Season	TempSplit	PrecSplit	0.002113
Any	Season	TempRange	Any	0.002080
Any	Any	TempSplit	PrecSplit	0.001928

Table 4. Top Ranked Summaries After Run 3.

Station	Date	Temp	Precip	Variance
Any	Season	TempSplit	Any	0.006401
LRegion	Any	Any	Any	0.006344
CRegion	Any	Any	Any	0.005890
Any	Any	Any	PrecRange	0.002911
CRegion	Any	Any	PrecSplit	0.001770
DRegion	Any	Any	Any	0.002389
Any	Any	TempRange	Any	0.002369
Any	Season	TempSplit	PrecSplit	0.002017
Any	Season	TempRange	Any	0.002009
LRegion	Any	Any	PrecSplit	0.001727

Table 5. Top Ranked Summaries After Run 4

Station	Date	Temp	Precip	Variance
LRegion	Any	Any	Any	0.006344
CRegion	Any	Any	Any	0.005890
Any	Any	Any	PrecRange	0.002911
DRegion	Any	Any	Any	0.002389
Any	Any	TempRange	Any	0.002369
CRegion	Any	Any	PrecSplit	0.001770
LRegion	Any	Any	PrecSplit	0.001727
LRegion	Any	TempSplit	Any	0.001507
CRegion	Any	TempSplit	Any	0.001463
CRegion	Any	Any	PrecRange	0.000773

Table 6. Top Ranked Summaries After Run 5

Station	Date	Temp	Precip	Variance
CRegion	Any	Any	Any	0.003998
Any	Any	Any	PrecRange	0.002910
Any	Any	TempRange	Any	0.002369
DRegion	Any	Any	Any	0.001359
CRegion	Any	Any	PrecSplit	0.001204
CRegion	Any	TempSplit	Any	0.001019
Any	Any	TempSplit	PrecRange	0.000670
Any	Any	TempRange	PrecSplit	0.000588
CRegion	Any	Any	PrecRange	0.000570
Any	Season	TempRange	Any	0.000516

Table 7. Top Ranked Summaries After Run 6

Run #	Highest Ranked Node	English-Language Description of the Top-Ranked Difference in Top Ranked Summary
1	Any-Any-Any-PrecSplit	More readings (79%) have precipitation = NONE than expected (20%).
2	Any-Decade-Any-Any	More readings (33%) have decade = 1940s than expected (20%).
3	Any-Any-TempSplit-any	Fewer readings (44%) have temperature = WARMER than expected (50%).
4	Any-Season-TempSplit-any	More readings (24%) have season = SUMMER and temperature = WARMER than expected (11%).
5	LRegion-Any-Any-Any	Fewer readings (17%) have latitude-region = SOUTH than expected (27%).
6	CRegion-Any-Any-Any	More readings (51%) have clustered-region = 5 than expected (39%).

Table 8. English-Language Highlights from Six Runs.

Algorithm PrintEnglishSummary.

Input: a list of summary nodes
k, the number of the nodes to print
Output: the records that are interesting

```
Rank nodes by the interestingness measure
For each of the top k nodes
   For each record R in the node
      Obs1 is the observed number of R
      Exp1 is the expected number of R
      Obs2 = TotalVotes – Obs1
      Exp2 = TotalVotes – Exp1

      // Degree of freedom is 1, alpha = 0.05
      ChiSquare = (Exp1-Obs1)^2 /Exp1 +
                  (Exp2-Obs2)^2/Exp2
      If ChiSquare >= 3.84
         Mark R as an output record
   Rank output records by their variances
   For each output record R, in ranked order
      If Obs1 < Exp1
         Print R "Fewer than expected"
      Else
         Print R "More than expected"
```

Figure 4. Algorithm PrintEnglishSummary

Login Data: Space does not permit the presentation of another full example. Nonetheless, to demonstrate the utility of our approach, we will mention the results of applying our DGG-Discover software to 523,253 lines of output from the Unix "`last`" program, which produces a single line for each user session, showing user id, login and logout times, and the duration of the session. The input data were collected over a period of somewhat more than two years, from June 1998 to June 2000. We focus on the login times, which can easily be mapped to the *YYYYMMDDhhmm* node in the ExGen graph, and user-ids, which can be grouped into categories such as CS undergrad, CS grad, staff, etc. The problem is to find interesting summaries of the data at various levels of granularity. In [22], a simpler version of the same problem with far less data was considered.

Using uniform expectations for each group of users and each node in the calendar DGG, the summaries listed in Table 6 were ranked highest. Since (*Group*, *Any*) is ranked highest, the expectation that people from all groups would log in with approximately equal frequency was furthest from matching the data, according to the variance measure. Full details on this example are given in [23].

The three highest ranked nodes in Table 9 are related. Investigation revealed that the (*Group*, *Any*) relation is unusual because `csugrd` and `unknown` have far more logins than expected while every other group has far fewer. The (*Any*, *WDWE*) relation is unusual because `weekday` logins are higher than expected based on the number of days. As well, these two factors combine to make (*Group*, *WDWE*) unusual.

USER	LOGINTIME	Variance
Group	Any	0.0341
Any	WDWE	0.0177
Group	WDWE	0.0124
Group	FiscalYear	0.0063
Group	AcademicYear	0.0052
Group	YYYY	0.0052
Any	AcademicYearAndTerm	0.0020
Any	AcademicYear	0.0015

Table 9. Top Ranked Nodes for Login Data

USER	LOGINTIME	Variance
Any	WDWE	0.0177
Any	AcademicYearAndTerm	0.0020
Any	AcademicYear	0.0015
Any	YYYY	0.0015
Any	FiscalYearAndQuarter	0.0013
Any	WeekdayName	0.0011
Any	FiscalYear	0.0011
Group	FiscalYear	0.0008

Table 10. Top Ranked Nodes after Run 2

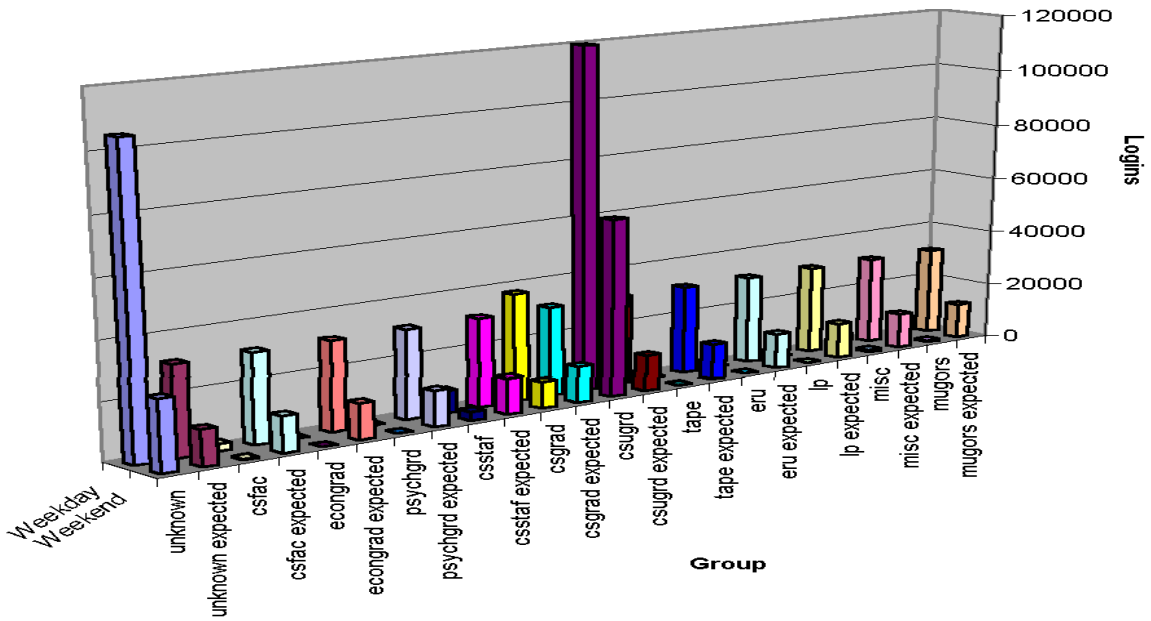


Figure 5. Graph for (Group, WeekdayOrWeekend)

This relationship is shown in Figure 5 for (*Group*, *WDWE*). On the X axis (across), the first value is the observed number of logins for the first group, the second value is the expected number of logins for this group, the third value is the observed number for the second group, the fourth value is the expected number for the second group, and so forth. The `csugrd` group continues off the chart to just under 300000 logins on weekdays.

After the observed distribution was accepted as the expected distribution for the groups of users, the ranking of the nodes was as shown in Table 10. Some nodes, most notably (*Group*, *Any*) and (*Group*, *WDWE*), have lower rankings because of the added knowledge about the distribution in logins among the groups. The top ranked summary is (*Any*, *WDWE*), which tells us that the number of logins on a weekday is higher than on a weekend day. (*Any*, *AcademicYearAndTerm*) tells us that fewer logins occur during some particular terms (e.g., 1998-2) than during other terms, where there are three terms per year.

When variance is used as the measure of interest, our technique provides an easy way for a user to construct a hierarchical statistical model based on his/her knowledge of the domain. By studying the summaries corresponding to the highest ranked nodes, the user may gradually recognize the factors that contribute to the observed variance. As expectations are adjusted, the variance may be reduced closer and closer to zero. Unlike traditional hierarchical statistical models, our approach allows multiple paths through the hierarchy.

5. CONCLUSION

We have outlined an approach to summarization, a type of data mining that aggregates data in a variety of ways. Our approach is based on Expected Distribution Domain Generalization Graphs, and it is well suited for domains where calendar and geospatial attributes play a crucial role due to the complexity of background knowledge about these types of attributes. We specified the components of an expected distribution domain generalization graph suitable for a calendar attribute. Because of the complexity of the calendar DGG, it is useful to specify distributions of values at various nodes and explore the consequences of these distributions on the interestingness ratings of various summaries of the same data. When applied to weather data and login data, using successive models of the user's expectations in each case, our approach conveniently identified several summaries that illustrated interesting facets of the data.

REFERENCES

[1] A. Andrusiewicz and M. E. Orlowska, On Granularity Factors That Affect Data Mining, *Eighth Int'l Database Workshop, Data Mining, Data Warehousing and Client/Server Databases*, Hong Kong, 1997.

[2] C. Antunes, A. Oliveira, Temporal Data Mining: An Overview, *KDD 2001 Workshop on Temporal Data Mining*, San Francisco, August 2001.

[3] Bay, S.D., and Pazzani, M.J., Detecting Group Differences: Mining Contrast Sets. *Data Mining and Knowledge Discovery*, 5 (3):213-246, 2001.

[4] E. Bertino, E. Ferrari, G. Guerrini, I. Merlo. Navigating Through Multiple Temporal Granularity Objects. In *Proc. 8th International Symposium on Temporal Representation and Reasoning (TIME'01)*, Cividale del Friuli, Italy, July 2001.

[5] C. Bettini, S. Jajodia, and X.S. Wang. *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Springer, Berlin, 2000.

[6] Y. Cai, N. Cercone, and J. Han. Attribute-oriented Induction in Relational Databases. In G. Piatetsky-Shapiro and W.J. Frawley (eds), *Knowledge Discovery in Databases*, AAAI Press, 1991, 213-228.

[7] X. Chen, I. Petrounias, and H. Heathfield. Discovering Temporal Association Rules in Temporal Databases, *Proc. Third European Conference on Principles of Data Mining and Knowledge Discovery (PKDD'99)*, 295-300, Prague, Sept., 1999.

[8] C. Combi, F. Pinciroli, and G. Pozzi. Managing Time Granularity of Narrative Clinical Information: The Temporal Data Model TIME-NESIS. In *Proc. Third Int'l Workshop on Temporal Representation and Reasoning (TIME'96)*, 88-93, Key West, FL, May 1996.

[9] http://interactive.usask.ca/skinteractive/modules/environment/ecoregions.

[10] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From Data Mining to Knowledge Discovery: An Overview. In U.M. Fayyad, G. Piatetsky-Shapiro, R. Smyth, and R. Uthurusamy (eds), *Advances in Knowledge Discovery and Data Mining*, AAAI Press, 1996, 1-34.

[11] L. Geng and H.J. Hamilton, *Expectation Propagation in ExGen Graphs for Summarization:* Extended Report, Technical Report CS-2003-3, Department of Computer Science, University of Regina, Regina, SK, Canada, May, 2003.

[12] I. Goralwalla, Y. Leontiev, M. T. Özsu, D. Szafron and C. Combi, Temporal Granularity: Completing the Puzzle, *Journal of Intelligent Information Systems*, 16 (1):41-63, January 2001.

[13] H.J. Hamilton and L. Findlater, Looking Backward, Forward, and All Around: Temporal, Spatial, and Spatio-Temporal Data Mining. In *Proceedings Fifteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2002)*, AAAI Press, Key West, FL, May, 2002, pp. 481-485. Invited paper.

[14] H.J. Hamilton, R.J. Hilderman, and N. Cercone. Attribute-oriented Induction Using Domain Generalization Graphs. In *Proc. Eighth IEEE International Conference on Tools with Artificial Intelligence (ICTAI'96)*, 246-253, Toulouse, France, November 1996.

[15] R.J. Hilderman and H.J. Hamilton, Heuristic Measures of Interestingness. In *Proc. PKDD'99*, 232-241, Prague, September 1999.

[16] R.J. Hilderman and H.J. Hamilton, *Knowledge Discovery and Measures of Interest*, Kluwer Acadamic, 2001.

[17] R. J. Hilderman, H. J. Hamilton, and N. Cercone, Data Mining in Large Databases using Domain Generalization Graphs, *Journal of Intelligent Information Systems*, 13:195-234, 1999.
[18] K. Hornsby, S. Kaufmann, *Data Visualization Techniques for Knowledge Discovery Based on Domain Generalization,* M.Sc. Thesis, Department of Computer Science, University of Regina, April 2002.
[19] C.H. Lee, C.R. Lin and M.S. Chen, On Mining General Temporal Association Rules in a Publication Database, *Proc. First IEEE International Conference on Data Mining*, San Jose, CA, Nov./Dec. 2001.
[20] Y. Li, P. Ning, X.S. Wang and S. Jajodia, Discovering Calendar-based Temporal Association Rules, in *Proc. TIME'01*, 111-118 Cividale del Friuli Italy, June 2001.
[21] C. P. Rainsford and J. F. Roddick, Adding Temporal Semantics to Association Rules, in *Proc. PKDD'99*, 504-509, Prague, September, 1999.
[22] D. J. Randall, H. J. Hamilton, and R. J. Hilderman, Generalization for Calendar Attributes Using Domain Generalization Graphs, *Fifth Int'l Workshop on Temporal Representation and Reasoning (TIME'98)*, 177-184, Sanibel Island, FL, May 1998.
[23] D. J. Randall, *Temporal Generalization in Databases Using Domain Generalization Graphs*, M.Sc. Thesis, Department of Computer Science, University of Regina, April 2002.
[24] J. F. Roddick and K. Hornsby (Eds.), *Temporal, Spatial, and Spatio-Temporal Data Mining*, Springer, Berlin, 2001.
[25] J. Zytkow, From Contingency Tables to Various Forms of Knowledge in Databases. In U.M. Fayyad, G. Piatetsky-Shapiro, R. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, AAAI Press, 1996, 329-349.