Looking Backward, Forward, and All Around: Temporal, Spatial, and Spatio-Temporal Data Mining

Howard J. Hamilton and Leah Findlater
Department of Computer Science
University of Regina, Regina, SK, Canada S4S 0A2
{hamilton, findlatl}@cs.uregina.ca

Abstract

We describe current research in temporal, spatial, and spatio-temporal data mining. In these types of data mining, a model of time, space, or space-time plays a nontrivial role. As an example of current research, we describe our MegaMiner prototype software. The DGG-Discover 5.2 module of MegaMiner is based on expected distribution domain generalization graphs (EDDGGs), which allow detailed domain knowledge about temporal and spatial generalization relationships to be specified, and then applied during the data mining process. As well, user expectations about the data can be specified and updated during the mining process. We illustrate the current state of the MegaMiner software by applying it to a previously unseen data set, describing the weather of the province of Saskatchewan for the period 1900 to 1949. We were able to find temporal and spatial relationships, but not spatio-temporal ones.

Introduction

We describe current research in temporal, spatial, and spatio-temporal data mining (*TSST mining*). In these types of data mining, a model of time, space, or space-time plays a nontrivial role. Rather than treating time as an attribute of a standard type, such as nominal, ordinal, integer, or real, *temporal data mining* employs a specific model of time that describes its special properties, such as the way it forms intervals, the periodic relationships of the calendar, the continuously moving point called "now", etc. Similarly, *spatial data mining* employs a specific model of space that describes its three-dimensional nature, its containment properties, etc. *Spatio-temporal data mining* employs a model relating to both time and space.

As an example of current research, we describe our MegaMiner prototype software. The DGG-Discover 5.2 module of MegaMiner is based on expected distribution domain generalization graphs (EDDGGs), which allow detailed domain knowledge about temporal, spatial, and spatio-temporal generalization relationships to be specified, and applied during the data mining process. An EDDGG is a graph-based structure for supporting automated data mining where the nodes describe domains of values at some level of granularity, the arcs describe generalization functions that translate values between domains, and expected distributions are associated with each node. For example, for a date-time attribute, the nodes represent concepts, such as days, the arcs represent aggregation relations, such as that from days to weeks, and the expected distribution for the day node represents the expected distribution for a measure attribute among the days of the week. Our method allows data to be automatically aggregated into summaries in many ways. By using the relative entropy (Kullback-Leibler) distance measure, summaries are ranked according to their distance from user expectations.

The remainder of this paper is organized as follows. First, we describe the major approaches to temporal, spatial, and spatio-temporal data mining. In each case, we emphasize the model that distinguishes the approach from general data mining techniques. Next, we describe an approach to data mining based on expected-distribution domain generalization graphs. Then we give a full example of the data mining process, highlighting the search for temporal, spatial, and spatio-temporal relationships. This example is based on applying our MegaMiner software to a previously unseen data set, describing the weather of the province of Saskatchewan for the period 1900 to 1949. Finally, we present our conclusions and suggestions for future research.

TSST Mining

Let us briefly review current research in temporal data mining, spatial data mining, and temporal-spatial data mining. In temporal data mining, most research has focussed on describing and comparing time series. For the model of time, difference equations are used for discrete values and differential equations for continuous values. A difference equation specifies how to predict the next value of an attribute based on the current values of the attributes.

To detect frequently occurring patterns in time series, portions of time series must be compared. If a pattern with a constant period is sought, [Han et al. 1999] provide a method for categorical values that finds partial patterns, such as: Jim reads the newspaper every morning between 7:00 and 7:30 am, but the rest of his day does not have much regularity. If trend components are present in the series, [Yu, Ng, and Huang 2001] recommend that the series be decomposed into three components, a seasonal component, trend, and noise.

If the series does not have a constant period or if different series must be compared, dynamic time warping [Berndt and Clifford 1994] allows elastic shifting of the X-axis (time) to detect similar shapes in the Y-axis (attribute value). [Keogh et al. 2000] introduced a piecewise algorithm for approximating a time series by dividing it into

equal-length segments and recording the mean value of the data points that fall within each segment and then applying dynamic time warping. [Chu et al. 2001] provide a different method based on comparing the sequences at successively doubled number of segments.

[Allen 1983] contributed an elegant model of time, based on 13 relations between intervals, for representing information about partially ordered events with durations. Given a set of partial relations between events, his algorithm fully propagates all constraints between intervals. By using reference intervals, such as major periods in a person's life, base level events can be related to intervals at successively higher levels of granularity. Allen did not mention the problem of automatically choosing suitable reference intervals, a relevant data mining problem.

[Chen, Petrounias, and Heathfield 1999] looked at discovering temporal association rules in temporal databases. [Rainsford and Roddick 1999] provide a method for adding temporal semantics to association rules based on structured relationships among temporal relations. [Li et al. 2001] look at discovering calendar-based temporal association rules. They also build on the Apriori algorithm for mining association rules to include temporal semantics. [Höppner 2001] uses a sliding window in combination with Allen's intervals to identify frequent patterns. During pre-processing, qualitative descriptions are used to divide the time series into small segments.

[Randall et al. 1998] first described the problem of performing data mining based on a nontrivial calendar structure, which was represented by a DGG. [Bettini et al. 2000] provide a *calendar algebra* to represent granularities of calendar data and the relationships between those granularities, although they do not provide a data structure similar to DGGs for representing the relationships between these granularities. [Bertino et al. 2001] build on the work of Bettini et al. to specify the syntax and semantics of expressions involving data with multiple temporal granularities. [Goralwalla et al. 2001] use an ordered granularity hierarchy from SUP (top) to year, month, day, hour, minute, second, INF (bottom).

Major approaches to spatial data mining are based on clusters, concept hierarchies, and spatial relations. [Ng and Han 1994] used clustering to discover relationships and characteristics that existed implicitly in spatial databases. Their CLARANS model of a spatially significant relationship is a cluster of spatially close points. [Koperski et al. 1998] perform generalization-based spatial mining by ascending hierarchies related to either spatial or non-spatial attributes. [Egenhofer and Franzosa 1991] developed an ontology of spatial relationships based on boundaries and interiors and some data mining systems build on this model.

Problems facing spatio-temporal data mining are described by [Erwig et al. 1999]. Since points move and regions move and change their shape (grow, shrink), a spatio-temporal database should contain information about moving objects. Mining such a database would be based on queries such as "Find all pairs of airplanes that came closer to each other than 500 meters during their flights." [Bittner 2002] suggests that a spatio-temporal granularity hierarchy be represented as a cross-product of temporal and spatial granularity hierarchies. [Roddick et al. 2001] list existing research on spatio-temporal data mining, based mainly on finding similarities in images from different times.

Expected Distribution Domain Generalization Graphs

An ***Expected Distribution Domain Generalization Graph*** (EDDGGs) is a graph-based structure where the nodes describe domains of values at some level of granularity, the arcs describe generalization functions that translate values between domains, and expected distributions are associated with each node. For example, for a date-time attribute, the nodes represent concepts, such as days, the arcs represent aggregation relations, such as that from days to weeks, and the expected distribution for the day node represents the expected distribution for a measure attribute among the days of the week.

Our data mining approach has five steps. First, a domain generalization graph is created (or adapted) for every relevant attribute by explicitly identifying the domains appropriate to the levels of 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 the Kullback-Leibler distance function. Fifth, the highest ranked summaries are displayed. Expected distributions are then adjusted and steps repeated as necessary. This method allows user expectations to be incorporated dynamically during the mining process.

Informally, a DGG can be thought of as a graph showing possible generalizations as paths through a graph. A formal definition is given in [Hilderman et al. 1999]. 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 node to that of the final node of the arc. The ***bottom*** node in the graph corresponds to the most specific domain of values and the ***top*** node corresponds to a domain called *Any*, containing all values.

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.

The simplest approach is to assume uniform distribution for all domains. Unfortunately, this approach may suggest inconsistent distributions. For example, uniform distribution over the domain of *WeekdayName* (0.14 for each day) is inconsistent with uniform distribution over the domain of *WeekdayOrWeekend* (0.5 for each). Two days, Saturday and Sunday, with a total expectation of 0.29 are generalized to Weekend, with a total expectation of 0.5, which is inconsistent.

To avoid inconsistencies and simplify the process of specifying expectations, a distribution is associated with a node in the EDDGG and from there it is propagated either upward or downward through the graph. ***Upward propagation*** translates a distribution from a node in the EDDGG up to at least one of its child node(s) and possibly higher as well. The distribution at the higher level is the original distribution proportionately weighted according to the relevant generalization relation. Either a uniform or a non-uniform distribution can be propagated upward. ***Bottom-up propagation*** propagates a distribution from the bottom node of the EDDGG to all other nodes. ***Downward propagation*** propagates a distribution from a node to at least one of its parent nodes and possibly lower as well. ***Top-down propagation*** propagates a distribution from the top node of the EDDGG to all other nodes, based on the assumption of uniformity at *Any*.

Bottom-up propagation always creates consistent distributions among all nodes because the distribution at each node is consistent with the distribution of the base values at the bottom node. However, the top-down approach can suggest inconsistent distributions for a particular node in the EDDGG based on multiple paths down to it. Propagating a distribution from a single child down to a single parent or multiple parents gives an unambiguous result, if the distribution of child values among parent values is known or assumed to be uniform. But when values are propagated down from multiple children to a single parent, it can be impossible to calculate consistent values without recalculating as much as the whole graph.

Our EDDGG implementation allows the user to specify expected distributions by the four methods described below. The generalized relations are compared to these distributions to discover anomalous distributions. An *explicit uniform distribution* gives the single expected probability for all values at a node. An *explicit histogram* specifies individual expected probabilities for all values at a node. The *data driven* approach applies bottom-up propagation to a (typically non-uniform) distribution. The *data dictionary* approach obtains an expected distribution of values for any node in the EDDGG from a data dictionary, such as a database relation. With any of these techniques, the distributions can be propagated both upward and downward. The *data driven* approach applies bottom-up propagation to a (typically non-uniform) distribution. An output summary is produced as comma-separated values, which are readily displayed and processed with Microsoft Excel and other standard tools.

The generalized relations are compared to these distributions to discover anomalous distributions. We use the Kullback-Leibler distance function to compare distributions. The formula is:

$$d = \sum_{k=1}^{n} p_k \log_2 \left(\frac{p_k}{q_k} \right)$$

where n is the number of observations, p_k is the observed probability distribution, q_k is the expected probability distribution.

When several attributes have EDDGGs, aggregation is performed to all nodes in the cross product of the EDDGGs. In this case, the expectation of a tuple is the product of the expectations of the values of its component attributes, unless an expectation (joint probability distribution) has been specified for some subset of the relevant DGG nodes.

Example Application

We conducted a series of experiments using data describing the weather in the province of Saskatchewan (hereafter SK) in Canada for the period 1900 to 1949. Each data record gives the highest temperature (0.1 degree Celcius) and the total precipitation (in mm, with snow converted to water) for every day of every year, for every weather station in the province. Other fields concerning low temperatures and snowfall were not used in our analysis. The number of tuples was 211,534.

We began the study with no knowledge of the data set, although as Saskatchewan residents, we had some opinions about the weather! Before beginning our study, we listed some "obvious" relationships that data mining should find:

W1. Weather in SK is hot in summer, cold in winter, cool in spring and fall.
W2. Weather in the south is warmer than in the north.
W3. Weather in the southwest is warmest.
W4. Precipitation falls mainly in the spring.
W5. On most days, it does not rain or snow.
W6. Most weather systems travel from west to east.

Our hope was that a straightforward application of our software could automatically find these or similar relationships, allow these relationships to be accepted as part of the domain theory, and then find some novel relationships.

We adapted a DGG previously created [Randall, Hamilton, and Hilderman 1998] for temporal attributes to form the simplified Date DGG shown in Figure 1. 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.

Additional DGGs are shown in Figure 2 for the Temperature, Precipitation, and Station. For the two nodes

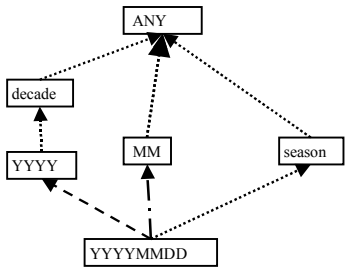

Figure 1. DGG for Date Attribute

in the Temperature DGG, we used the ranges shown in Table 1. Precipitation was handled similarly, with the PrecipSplit node having values of NoPrecip and SomePrecip. For Station, we grouped them from south to north into four regions (South, LowMid, HighMid, and North) to create the Region node. We also calculated an adjusted distance d to any point (Lat, Long) from the

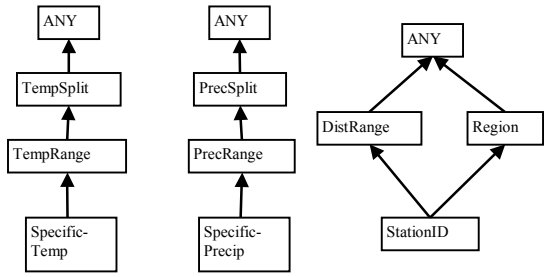

Figure 2. Temperature, Precipitation, and Station DGGs

DGG Node			
TempRange	**Nominal value**	**Min Value**	**Max Value**
	Cold	-42C (-44F)	0C (32F)
	Cool	0C (32F)	15C (61F)
	Warm	15C (61F)	25C (77F)
	Hot	25C (77F)	52C (126F)
TempSplit	**Nominal Value**	**Values from TempRange**	
	Colder	Cold, Cool	
	Hotter	Warm, Hot	

Table 1. High Temperature Ranges

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 [Eco 2002]. 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 DistRange node, we divided this range into 10 equal-sized intervals.

We began by investigating the number of values, which is the simplest measure attribute. We used the flexibility of the EDDGG framework to specify our expectations at whatever level of granularity seemed most appropriate. First, we specified the minimum and maximum dates at the *YYYYMMDD* node with an expected distribution of uniform (indicated as U), and this expectation was propagated to all nodes above it in the DGG. This propagation automatically recorded the expectation that February would have a smaller count, that some years and decades would have a slightly smaller count due to the exact number of leap years, etc. By adapting a standardized calendar DGG, we immediately obtained this benefit. For Temperature, we assumed a uniform distribution across the TempRange, and propagated this assumption upward, and uniformly downward. Similarly, for Precipitation, we assumed a uniform distribution across PrecipSplit, which means that an equal number of days with and without precipitation are expected, and propagated this distribution downwards. Lastly, we assumed a uniform distribution at the StationId node, i.e., that all weather stations are expected to have the same number of observations, and propagated this upward.

Run 1. (YYYYMMDD=U, TempRange=U, PrecipSplit=U, StationID=U). The results from the first run immediately identified false expectations. The highest ranked node was MM-TempRange-PrecipRange-StationId, and four other nodes with MM-…-StationId and Year-…-StationId were next, indicating that a great variation occurred in the number of reports from various stations among the years and months. Further examination of the data showed that stations only gradually started reporting across the 50 year period, and that occasionally stations would stop reporting for a while, or completely. We were easily able to determine the date of first and last report for each station. Using these dates, we specified an expected distribution, hereafter called A, which was uniformly spread across all stations between their starting and ending dates. This joint probability distribution was specified at the *YYYYMMDD-StationId* node of the generalization space. We manually propagated distribution A to the *YYYYMMDD* node in the Station DGG as distribution A_Y, and to the *StationId* node in the Station DGG, as distribution A_S.

Run 2. (YYYYMMDD= A_Y, TempRange=U, PrecipSplit=U, StationID= A_S): The second run ranked MM-TempRange-PrecipRange-StationId node highest. Since the next few nodes also included PrecipRange, we looked at the distribution (hereafter called B) for ANY-ANY-PrecipRange-ANY, which is shown in Figure 4. We made B the expected distribution at the PrecipRange node, and deleted our previous expectation at PrecipSplit. Distribution B was propagated upward and uniformly downward. Distribution B corresponds to relationship W5.

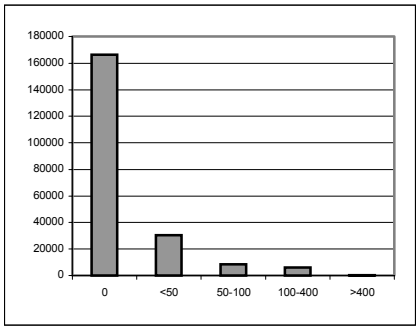

Figure 4. PrecipRange Distribution (B).

Run 3. (YYYYMMDD= A_Y, TempRange=U, PrecipRange=B, StationID= A_S): The third run ranked the MM-TempRange-ANY-Station node as highest. We accepted the distribution (C) at the TempRange node. Distribution C was propagated upward and uniformly downward.

Overall, we found that the highest ranked node was not providing a clear indication of what expectation to adjust next. The problem seemed to be with the Kullback-Leibler measure, which although widely accepted for comparing distributions, does not seem like an appropriate heuristic for guiding node selection in MegaMiner.

We switched to checking individual nodes in the EDDGG that corresponded to our hypotheses. When we examined the results at the Season-TempRange node, we confirmed W1, as shown in Figure 5. When we examined the results at the TempRange-Region node, we confirmed W2, as shown in Figure 6. At TempRange-DistRange, we found that the most southwestern region was **not** the warmest, but thereafter the temperatures decreased. Further examination, showed that only one station was in the first region and it had large gaps in its reporting periods. From Season-PrecipRange, we found that most rainy days were in the summer not the spring, and the average at Season-SpecificPrecip confirmed that we were wrong about W4, as

shown in Table 2. We were unable to check W6, because no adequate spatio-temporal model was defined. To do this, we would have searched for significant precipitation at western versus eastern stations on successive days, but the software did not support this conveniently.

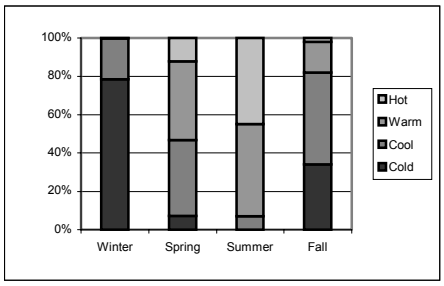

Figure 5. Season vs. Counts for TempRange

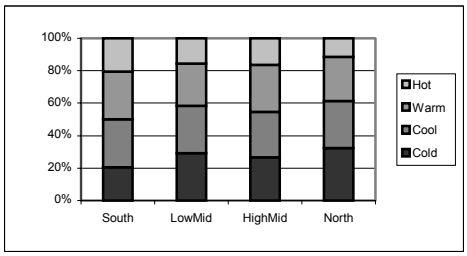

Figure 6. Region vs. Counts for TempRange.

Season	Average Temp	Average Precip
Winter	-8.9	5.7
Spring	14.4	13.1
Summer	23.7	16.6
Fall	3.8	6.9

Table 2. Weather by Season.

Conclusion

Temporal, spatial, and spatio-temporal data mining provide many challenges. Data mining has been applied using temporal models for time series, intervals, temporal association rules, and calendar data. Spatial data mining is less developed but a variety of clustering techniques have been applied and containment models are beginning to be applied. Spatio-temporal data mining is in its infancy. Some of the problems have been clearly identified, but few general techniques have been designed. In particular, the facilities in our MegaMiner software are inadequate.

Four possible improvements to MegaMiner were identified during this research. First, the use of other interestingness measures [Hilderman and Hamilton 2001] than Kullback-Leibler distance should be explored in the context of providing additional guidance to the user when selecting among summaries. Secondly, facilities for propagating joint probability distributions for combinations of attributes are needed. Thirdly, additional functionality for specifying spatial DGGs would be helpful. Finally, support for spatio-temporal data mining is required.

References

Allen, J. F. 1983. Maintaining Knowledge about Temporal Intervals, *CACM* 26(11):510-521.

Antunes, C., and Oliveira, A. 2001. Temporal Data Mining: An Overview, *KDD 2001 Workshop on Temporal Data Mining*, San Francisco.

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

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

Bittner, T., Granularity in Reference to Spatio-Temporal Locations and Relations, *Proc. FLAIRS'2002*, this volume.

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

Chu, S., Keogh, E., Hart, D., and Pazzani, M. 2002. Iterative Deepening Dynamic Time Warping for Time Series, *SIAM KDD* 2002.

Eco 2002. http://interactive.usask.ca/skinteractive/modules/environment/ecoregions.

Egenhofer, M. J. and Franzosa, R. D. 1991. Point-set topological spatial Relations, *Int. J. Geographical Information Systems*, 5(20): 161-174.

Erwig, M., Guting, R. H., Schneider, M., and Vazirgiannis, M. 1999. Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases, *GeoInformatica*, 3(3).

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

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

Han, J., Dong, G., and Yin, Y. 1999. Efficient Mining of Partial Periodic Patterns in Time Series Database, In *IEEE Conf. on Data Engineering (ICDE'99)*, 106-115.

Hilderman, R. J., Hamilton, H. J., and Cercone, N. 1999. Data Mining in Large Databases using Domain Generalization Graphs, *Journal of Intelligent Information Systems*, 13:195-234.

Hilderman, R. J., and Hamilton, H. J. 2001. *Knowledge Discovery and Interest Measures*, Kluwer, 2001.

Höppner, F. 2001. Discovery of Temporal Patterns: Learning Rules about the Qualitative Behaviour of Time Series, *Principles of Data Mining and Knowledge Discovery (PKDD'2001)*.

Koperski, K., Han, J., and Adhikary, J. 1998. Mining Knowledge in Geographical Data, *CACM*.

Li, Y., Ning, P., Wang, X. S., and Jajodia, S. 2001, Discovering Calendar-based Temporal Association Rules, in *Eighth Int'l Symposium on Temporal Representation and Reasoning (TIME'01)*, 111-118 Cividale del Friuli Italy.

Ng, R. T., and Han, J. 1994. Efficient and Effective Clustering Methods for Spatial Data Mining. In *Proceedings of the 20th VLDB Conference*, 144--155, Santiago, Chile.

Rainsford, C. P., and Roddick, J. F. 1999. Adding Temporal Semantics to Association Rules, in *Proc. PKDD'99*, 504-509, Prague.

Randall, D. J., Hamilton, H. J., and Hilderman, R. J. 1998. Generalization for Calendar Attributes Using Domain Generalization Graphs, *Fifth Int'l Workshop on Temporal Representation and Reasoning (TIME'98)*, 177-184, Sanibel Island, FL.

Roddick, J., Hornsby, K., and Spiliopoulou, M. 2001. An updated bibliography of temporal, spatial and spatio-temporal data mining research, *Temporal, Spatial, and Spatio-temporal Data Mining*, Springer, 147-163.