Pattern Mining in Sparse Temporal Domains, an Interpolation Approach Work in Progress

Christian Pölitz
University of Bonn
poelitz@iai.uni-bonn.de

Abstract

Weblog systems, mobile phone companies or GPS devices collect large amounts of personalized data including temporal, positional and textual information. Patterns extracted from such data can give insight in the behavior and mood of people. These patterns are often imprecise due to sparseness in the data. We propose an interpolation technique that augments local patterns with elements that seem to be locally unimportant but with global information they are interesting.

id	topics		
1	A	B	C
2	A		C
3	A	B	
4		B	C
5	A		C
6	A	B	C
	$[s_1, e_1]$	$[s_2, e_2]$	$[s_3, e_3]$
	time intervals		

Table 1: A possible development of topics in a weblog.

1 Introduction

Large amounts of data are gathered in social web applications, on mobile phone calls, GPS signals from cars or animals, to name only a few. The companies that possess these data are highly interested in benefiting from the contained information. One way to extract such information is to find patterns. By patterns we mean regularities like frequently mentioned topics in weblogs, paths that certain animals take for foraging or main traffic routes.

All the data that we investigate include temporal information like the time a weblog is written, the time stamp of a telephone call or the time of GPS signals. This leads to two characteristics for the patterns we are interested in. First the patterns must be temporally ordered, for instance which topics follow a certain frequent topic in weblogs or frequently consecutively visited places. These patterns are called sequential patterns and several algorithmical approaches exist to solve this task. An overview of different sequential pattern mining methods and some special cases are given by Zhao and Bhowmick in [Q. Zhao, 2003].

The second characteristic is that the patterns have a temporal extend and transition times. This means each pattern has a clear beginning time and an end time. All contained subpatterns have also a beginning and an end. From this we indirectly know the time interval of the pattern. The transition time is the time interval between two consecutive elements of the pattern and can be calculated from the temporal extend of the subpatterns.

Possible patterns can be the evolution of topics in blog entries. Such information is useful for many different companies. They can directly search for such topic patterns that deal with their products or services. Other patterns could be movements of groups of people in certain areas. These patterns can be used for targeted advertisements by poster campaigns. To know where certain people move could influence the kind of commercial being shown on posters.

One big problem by extracting patterns from the data is sparseness. Sparseness means that the data contains large temporal gaps and/or very few data points with an assignment to one specific identifier like blog entry author, mobile phone device or GPS device. Such weaknesses in the data can badly influence the search for local patterns. As a result we may get patterns with very large temporal intervals with no information of the intermediate time. We try to deal with this problem by using global information to interpolate between consecutive pattern elements with a big temporal difference. The global information can be of arbitrary source, additionally or data immanent.

In weblogs for instance, ”bloggers” might write entries quite rarely or there are large breaks between entries. This can for example be due to individual behavior or holidays. For these times it might be impossible to retrieve local patterns. We try to overcome this problem by using global information of all bloggers in the corresponding times. We simply use the assumption that the hot topics in the time in which we lack local information would also track the attention of the corresponding bloggers.

On table 1 we see a possible distribution of topics A, B and C that are discussed by six different bloggers in certain time intervals. We see that two of three persons who write about topic A in time interval $[s_1, e_1]$ do write later on in time interval $[s_3, e_3]$ about topic C. This leads to the rule if someone writes about topic A, they also write about topic C in the given times with a confidence of four fifth.

For the intermediate time between the end of topic A e_1 and the beginning of topic C s_3 there is topic B. Only two persons write about topic B in the mean time after writing about topic A and before topic C. The data supports this only by one third. Applying strictly a pattern mining method we could loose this intermediate topic due to the low support.

From the data itself there is the possible pattern of writing about topic A in time interval $[s_1, e_1]$ leads to writing about topic B in $[s_2, e_2]$. Further there is the pattern of people writing about topic B in $[s_2, e_2]$ will later write

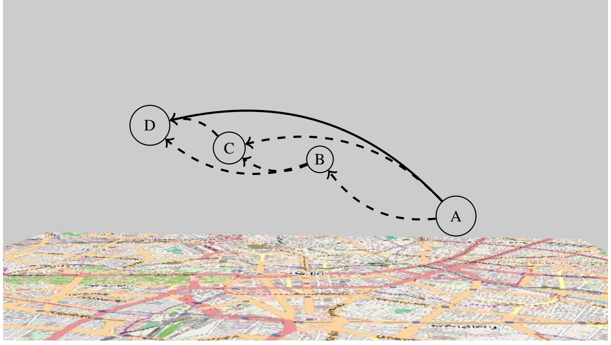

Figure 1: Possible movement pattern in geographical space and in time.

about topic C in $[s_3, e_3]$. Both patterns have a confidence of three fifth. Since topic B was mentioned four times in the time between A and C we augment the previous pattern by: if people write about topic A and C in the corresponding times they will also write about topic B in the intermediate time $[s_2, e_2]$.

From this example we gain the following knowledge. A possible interpolation of a pattern (or augmentation) should only contain elements (topics) with at least a certain amount of appearance. Additionally the interpolated topics should by at least mentioned by some of the persons that form the previous (original) pattern.

An other example that concentrates on movement patterns of GPS data is shown on figure 1. Four places are visited in different times indicated by the height of the nodes. The GPS signals have unequal dwell times which is shown by different sizes of the nodes. The solid directed line shows a possible pattern among the places extracted from the GPS data. This pattern leads to the rule that if someone was at place A at the specific time they will be later at place D at another specific time.

Due to unregularities and annoyances there is a large temporal and spatial gap between place A and D. From the whole data set we know that in the intermediate time the places C and B are frequently visited and some of the people that support the previous patterns have been at this times at these places, but not enough to make this pattern interesting. From this information we augment the pattern by the places C and B. The resulting pattern is shown on figure 1 as dashed directed lines.

2 Pattern Extraction Method

The task to solve the above stated problems is a clear pattern mining task. An introduction in pattern mining and a survey on several algorithms is described by Goethals in [Goethals, 2003].

First we describe definitions about the patterns we want to find. We include constrains that make sure we retrieve reasonable patterns from data from a temporal domain containing additional information like texts from blog entries or GPS coordinates of cars.

Generally we define a pattern as $P = i_1[s_1, e_1], i_2[s_2, e_2], \cdots, i_n[s_n, e_n]$. i_j are the basis elements of the patterns, analogue to the items in frequent item set mining. The elements are for instance topics in weblogs, mobile radio cells or clusters of GPS signals. Each element has a starting time s_j and an ending time e_j. This means at this time interval the topics are frequently used in weblogs, many phone calls were made in radius of a cell or lots of GPS signals are received. Analogue to sequential pattern mining we assume the elements to be temporally ordered, i.e. $s_k \leq s_l$ for $k < l$.

We define the elements i_j as disjoint subsets of the analyzed data set $D = \{o_1, \cdots, o_n\}$. Each subset contains only data points with a similarity larger than a given threshold. The similarity must be defined w.r.t. to the domain of the data and the application. For instance, if the data are blog entries and we are interested in patterns of topics, the similarity could be the semantical alikeness of the words in the entries to predefined or learned topics. In this case each subset would contain all the blog entries that contain words that indicate a certain topic.

In case the data are mobile phone calls, the similarity should be the mobile radio cell the phone is connected to (We assume that the mobile phones are only connected to one cell at a time). This means all mobile phone calls in a certain cell are highly similar and are contained in the same subset.

For GPS data of cars the similarity could be the geographical distance to certain streets or to concentrations of many other cars (clusters). A grid (even or uneven) division of the traveled area is also possible. Hence all GPS data of cars within a specific grid cell are similar.

These are only a few possible implementations of a similarity. In general we define the elements as described in equation 1 with a similarity measure $sim : D \times D \rightarrow \Re$ and a user defined (domain and application dependent) threshold parameter τ.

$$i_j = \{o_{1_j}, \cdots, o_{n_j} | o_{i_j} \in D, sim(o_{k_j}, o_{l_j}) \geq \tau\} \qquad (1)$$

Among the elements, patterns are extracted w.r.t. a frequency measure. The measure depends on the uniqueness of the data points. For that we assume the data points to be assigned to an identifier id_j. Possible identifiers are the persons that wrote blog entries, mobile phones or GPS devices of cars. We generally assume the following structure of data points $o_j = (X, t, id)$ with X a vector information, t a time stamp and id the identifier. The information X can be the textual content of a blog entry, the cell information of a mobile radio cell or the position of a GPS device at time t.

A one element pattern $i_j[s_j, e_j]$ is frequent when it contains more than a predefined threshold n_{min} many unique identifiers. Additionally all contained data points have a time stamp $t \in [s_j, e_j]$ and are similar w.r.t. the above mentioned similarity measure sim.

A continuation of an $n-1$ elements pattern $P = i_1[s_1, e_1], \cdots, i_{n-1}[s_{n-1}, e_{n-1}]$ are the elements $i_{n_j}[s_{n_j}, e_{n_j}]$ with the following properties:

- $i_{n_j}[s_{n_j}, e_{n_j}]$ contains at least n_{min} data points with identifiers that are also contained in $i_{n-1}[s_{n-1}, e_{n-1}]$
- all data points in $i_{n_j}[s_{n_j}, e_{n_j}]$ with identifier id_j are temporally after the data point in $i_{n-1}[s_{n-1}, e_{n-1}]$ that have the identifier id_j too

Since the time span of the elements can be very large we partition the temporal information in intervals. These intervals can be periods with many blog entries to a certain topic, times with lots of phone calls or congestion of cars. A calendrical partitioning for instance in days or weeks is also possible.

To find such partitions we define an additional temporal similarity on the time information of the data points. All data points $o_k \in i_j$ must have a similarity $sim_t : D \times$

$D \rightarrow \Re$ larger than a predefined threshold τ_t (see equation 2). Together with the previous statements we define the elements i_j in equation 2.

$$i_j = \{o_{1_j}, \cdots, o_{n_j} | o_{i_j} \in D, sim(o_{k_j}, o_{l_j}) \geq \tau, \\ sim_t(o_{k_j}, o_{l_j}) \geq \tau_t\} \tag{2}$$

We include this statement in our pattern definition. For the one element pattern $i_j[s_j, e_j]$ we have the additional restriction that the contained data points are similar w.r.t. the temporal similarity measure sim_t. The continuations $i_{n_j}[s_{n_j}, e_{n_j}]$ have the additional property:

- all data points in $i_{n_j}[s_{n_j}, e_{n_j}]$ with identifier id_j that are in $i_{n-1}[s_{n-1}, e_{n-1}]$ too must be similar w.r.t. to the temporal similarity sim_t

The made definitions can be easily integrated into a sequential pattern mining method like PrefixSpan. In [Pei *et al.*, 2001] Pei et al. introduce the sequential pattern mining algorithm PrefixSpan that generates sequential patterns by growing prefixes of patterns. Such a method extracts all sequential patterns that are supported by at least n_{min} identifiers. For blog entries this means that at least n_{min} different persons must have written articles to the topics in the found patterns in the specified times. In case of GPS or telephone data the places in the patterns must be visited in the corresponding time intervals by at least n_{min} different GPS or telephone devices.

3 Pattern Extraction of Sparse Data

In the case of sparse data the pattern extraction becomes more difficult as we described above. In this case sparseness means that only very few data points share the same identifier while on the other hand there are many different identifiers. The data points with the same identifiers additionally have large differences in their temporal attributes.

To cope with these shortcomings we suggest an interpolation technique that uses all data points to find descriptions of the time between data points with the same identifier. According to a pattern extraction method we infer intermediate sequence elements of the patterns. This is done by allowing to add to a pattern sequence $P = i_1[s_1, e_1], \cdots, i_{n-1}[s_{n-1}, e_{n-1}]$ not only elements having enough data points with identifiers also contained in element $i_{n-1}[s_{n-1}, e_{n-1}]$ but having enough other data points with a very large similarities.

For possible continuations $i_{n_j}[s_{n_j}, e_{n_j}]$ of a pattern $P = i_1[s_1, e_1], \cdots, i_{n-1}[s_{n-1}, e_{n-1}]$ we state that in case the temporal difference $|e_{n-1} - s_{n_j}|$ exceeds a predefined threshold Δt and there exists no other $i_{n_{j'}}[s_{n_{j'}}, e_{n_{j'}}]$ that is temporally between e_{n-1} and s_{n_j} with the properties for a continuation, we insert between $i_{n-1}[s_{n-1}, e_{n-1}]$ and $i_{n_j}[s_{n_j}, e_{n_j}]$ interpolated subpatterns $P^* = i^*_1[s^*_1, e^*_1], \cdots, i^*_n[s^*_n, e^*_n]$ that result in an augmentation of the pattern to: $i_{n-1}[s_{n-1}, e_{n-1}], P^*, i_{n_j}[s_{n_j}, e_{n_j}]$.

By Δt we try to generate reasonable patterns with no more than this threshold of time between consecutive pattern elements. Depending on how much the temporal difference of two such pattern elements extends the threshold we allow to augements the pattern by less supported elements. For that we introduce a new parameter n^*_{min} (see **??**.

$$n^*_{min} = \frac{\Delta t \cdot n_{min}}{s_{n-1} - e_{n_j}} \tag{3}$$

The first element $i^*_1[s^*_1, e^*_1]$ of the pattern P^* contains more than a predefined threshold n_{min} many unique identifiers of which at lest n^*_{min} many are also in $i_{n_j}[s_{n_j}, e_{n_j}]$. Additionally all contained data points are temporally between e_{n-1} and s_{n_j}, further they are similar w.r.t. the above mentioned similarity measures sim and sim_t.

The continuation of an $n-1$ elements subpattern $P^* = i^*_1[s^*_1, e^*_1], \cdots, i^*_{n-1}[s^*_{n-1}, e^*_{n-1}]$ are the elements $i^*_{n_j}[s^*_{n_j}, e^*_{n_j}]$ with the following properties:

- $i^*_{n_j}[s^*_{n_j}, e^*_{n_j}]$ contains at least $n^*_{min} < n_{min}$ data points with identifiers that are also contained in $i_{n-1}[s_{n-1}, e_{n-1}]$ and in $i_{n_j}[s_{n_j}, e_{n_j}]$
- all together $i^*_{n_j}[s^*_{n_j}, e^*_{n_j}]$ contains at least n_{min} data points with different identifiers
- all data points in $i^*_{n_j}[s^*_{n_j}, e^*_{n_j}]$ with identifier id_j are temporally after the data points in $i_{n-1}[s_{n-1}, e_{n-1}]$ that have the identifier id_j too and later than the ones in $i_{n_j}[s_{n_j}, e_{n_j}]$
- all data points in $i^*_{n_j}[s^*_{n_j}, e^*_{n_j}]$ must be similar w.r.t. the similarity measure sim
- all data points in $i_{n_j}[s_{n_j}, e_{n_j}]$ with identifier id_j that are in $i_{n-1}[s_{n-1}, e_{n-1}]$ too must be similar w.r.t. to the temporal similarity sim_t

These definitions can be easily integrated in our previously stated pattern extraction algorithm. For a continuation of a pattern with large temporal difference as described above we simply apply again a sequential pattern mining method like PrefixSpan, but only to data points that have timestamps in the corresponding time interval.

A schematic example of the interpolation and augmentation of patterns of weblogs is shown in table 2 and for GPS movement patterns on figure 2.

As quality for the interpolated elements we use a similarity measure sim_{inter} based on the differences $d(i^*_j, i^*_{j+1})$ between consecutive elements i^*_j, i^*_{j+1}. We generally assume that such elements do not differ too much. This assumption is similar to many interpolation techniques that require smooth interpolating functions. For movement data the difference could be the geographical distance and for weblog semantical difference between topics. Finally the similarity of the elements is the corresponding difference divided by their temporal difference $s^*_{j+1}-, e^*_j$ (see 4).

$$sim_{inter}(i^*_j, i^*_{j+1}) = \frac{d(i^*_j, i^*_{j+1})}{s^*_{j+1} - e^*_j} \tag{4}$$

4 Future and Ongoing Work

Currently we are applying our proposed interpolation method on several data sets. Experiments on GPS signals show promising results. In this case it is easier to interpret the results since they can be shown on maps and the patterns are accustomed movements. The results on a large data set of GPS signals from cars are shown on figure 3. The patterns are extracted among concentrations of cars in Milan in the morning.

There is a pattern of 5 cars starting at place A from 4:06 am to 6:52 am reaching place E from 6:10 am to 7:09 am. The corresponding data points that form this pattern have a large temporal and a large spatial difference. W.r.t. the found patterns it seems very likely that cars moving from A to E went there by B, C and/or D. Applying our method for

pattern	hot topics among weblog entries							
P	$\cdots$	i_{n-1}						i_n
P_1^*			i_1^*	i_2^*	$\cdots$	i_{n-1}^*	i_n^*	
P_2^*			i_1^*	i_2^*	$\cdots$	i_{n-1}^*		
P_3^*			i_1^*	i_2^*	$\cdots$			
				$\cdots$				
$P * P_1^*$	$\cdots$	i_{n-1}	i_1^*	i_2^*	$\cdots$	i_{n-1}^*	i_n^*	i_n
$P * P_2^*$	$\cdots$	i_{n-1}	i_1^*	i_2^*	$\cdots$	i_{n-1}^*		i_n
$P * P_3^*$	$\cdots$	i_{n-1}	i_1^*	i_2^*	$\cdots$			i_n
				$\cdots$				
		$[s_{n-1}, e_{n-1}]$	$[s_1^*, e_1^*]$	$[s_2^*, e_2^*]$	$\cdots$	$[s_{n-1}^*, e_{n-1}^*]$	$[s_n^*, e_n^*]$	$[s_n, e_n]$
	time intervals							

Table 2: A schematic representation of interpolated patterns of topics in weblogs.

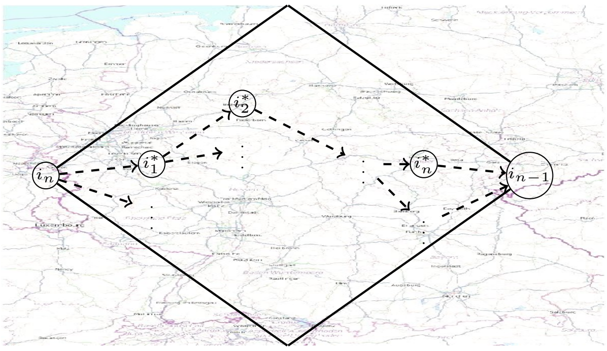

Figure 2: A schematic representation of a possible interpolation of two consecutive pattern elements with large temporal/spatial distance.

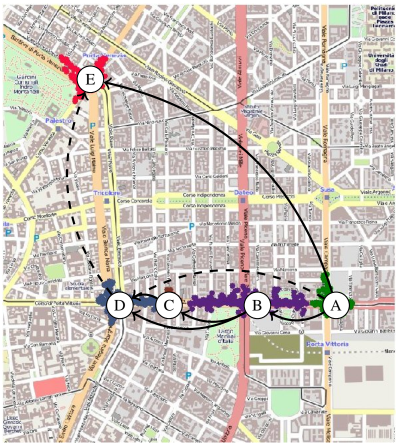

Figure 3: Augemented patterns among concentrations of cars.

pattern extraction on sparse data with $n^*_{min} = 2$ we retrieve additional patterns as shown by the dashed lines on figure 3. There are 2 of the 5 cars from the original pattern that went from A to D to eventually arrive at E. Beside these 2 cars there were several others cars in the intermediate time from 5:05 am to 5:54 am. We can now augment the pattern of moving from A to E by place D.

On weblog data an interpolation is harder to validated. We plan to artificially include sparseness in data. We want to analyze how good the augmented patterns are compared to corresponding patterns found without artificial sparseness. To achieve this we use retrieved patterns from weblog that have no sparse elements w.r.t. the statements above. Furthermore we insert sparseness into the data on data points that support the previously found patterns. On different degreess of sparseness we want to analyze how good the augmented patterns can reconstruct the original ones and how much they differ.

In conjunction with that, we plan to further investigate the influence of the parameters n_{min} and n^*_{min}. We hope to find heuristic descriptions of possible settings of these parameters concerning the domain of the data, additional (statistical) information and the application.

An other important aspect is the computation time of the pattern extraction and the interpolation. We generally assume the data to be ordered in time. By this the patterns can be found in a faster way and less main memory is used since we must only consider data with timestamps larger than the last element of the current pattern. Although this means in worst case we have to scan the whole data ($O(n)$) generally we assume local patterns to be small enough and short in time to be easily placed in main memory. For the interpolation we even expect subpattern with very few elements. That is due to the fact that only a smaller number of data points with timestamps between the corresponding elements of the original pattern might exists compared to the whole data set. Different data structures and data sources will be investigated in this context.

The propose method seems very promising to find patterns among large amounts of data from a temporal domain with additional sparseness. We are planning to show this on many different data sets in future works.

5 Aknowledgement

The work has been done within the research project ViAMoD Visual Spatiotemporal Pattern Analysis of Movement and Event Data, which is funded by DFG Deutsche Forschungsgemeinschaft (German Research Foundation) within the Priority Research Programme Scalable Visual Analytics (SPP 1335). We also thank the anonymous reviewers for their helpful suggestions.

References

[Goethals, 2003] B. Goethals. Survey on frequent pattern mining. Technical report, 2003.

[Pei *et al.*, 2001] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei chun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. pages 215–224, 2001.

[Q. Zhao, 2003] S.S. Bhowmick Q. Zhao. Sequential pattern mining: a survey. Technical report, Technical Report Center for Advanced Information Systems, School of Computer Engineering, Nanyang Technological University, Singapore, 2003.