T-Pattern Miner*

Short user manual

Mirco Nanni and Fabio Pinelli

Pisa KDD Laboratory

ISTI - CNR, Area della Ricerca di Pisa

Via Giuseppe Moruzzi, 1 - 56124 Pisa, Italy

November 6, 2007

*This research is partly funded by the EU contract GeoPKDD IST-FP6-014915.

Contents

1 What is T-Pattern Miner

T-pattern Miner is a frequent sequential pattern algorithm designed for trajectory data that describe the movements of a set of Moving Objects, e.g., GPS traces of mobile devices. The extracted patterns are represented as sequences of spatial regions temporally annotated with typical transition times. The extraction engine of T-pattern Miner is based on the PrefixSpan algorithm. The extraction of annotated frequent patterns is performed w.r.t. a minimum support threshold (minimum frequency of a pattern) and a time threshold (time tolerance used in matching transition times). The regions used in the extracted patterns are computed by means of heuristics for locating Regions Of Interest (in particular: dense regions w.r.t. a density threshold) of bounded size over a grid.

2 Syntax

General syntax:

```
TPatternMiner <input_file> <Min_sup> <tau> [OPTIONS]
```

`input_file` is a text file containing the input trajectories, formatted according to the specifications presented below.

`Min_sup` is the minimum support threshold, i.e., a standard parameter used in frequent pattern mining algorithms. A pattern is *frequent* if it occurs at least `Min_sup` times in the dataset, and, by default, this value is also used to extract Regions of Interest on the grid: a cell of the grid is dense if at least `Min_sup` trajectories have crossed such cell. The default behavior can be changed by means of the `-density` option described below.

Finally, `tau` is a *temporal tolerance* threshold used to find typical transition times: two transition times will be considered equivalent if their difference is not greater than `tau`.

The optional parameters:

-density NN : the default value of this parameter is equal to the minimum support value. This threshold is used during the extraction of the Regions of Interest, for establishing if a cell is dense or not (the cell is dense $\Leftrightarrow$ density of the cell$\geq$NN).

-epsilon NN : this is the spatial approximation around each point of the trajectories. The default value is equal to 0.

-no_interpolate : typically, we linearly interpolate the position of moving objects between two consecutive points of a trajectory. Setting this options the software uses only the input points for computing the density of the cells crossed by the trajectory. The default value is OFF.

-rescale_density Setting this options the density threshold proportionally decreases w.r.t the number of trajectories belonging to the projected database. The default value for this parameter is OFF.

-time_gap NN : minimum temporal gap required between a region and the following one in the same pattern. The default value is 0.

-space_gap NN : minimum spatial distance between two consecutive regions in the same pattern. The default value is 0.

-side NN : size of the (square) cells used to compute Regions of Interest. Default is 1/100 of the diagonal of the spatial bounding box of the dataset.

-max_reg_size NN : each side of the generated Regions of Interest must be large at most NN cells. The default value is ∞.

-difference_tolerance NN : two consecutive regions in a pattern can overlap at most of a fraction NN. The defaul is 0.

-semi_static : the Regions of Interest used in the patterns will be computed once for all the beginning of the computation – instead of being refine through the computation. The default value is OFF.

-static FILENAME : the Regions of Interest used in the patterns will be read from file and kept unchanged along the whole computation.

There are also available some debug parameters:

-max_n_trajs NN : sets the maximum number of trajectories of the data input file that the algorithm has to take as input.

-max_n_points NN : sets the maximum number of points that algorithm takes for each trajectory. With this parameter the dataset is cut vertically.

-skip_first_n_trajs NN : this parameter allows to skip the first NN trajectories in the dataset.

-verbose : with this option active, the execution of the algorithm returns the files with all regions found for each extracted prefix. The default value is OFF and the algorithm returns only the file with all the regions and the frequent patterns extracted.

3 Input format

Each line of the input file contains the whole trajectory of a moving object, i.e.:

<ID> $< k = n.snapshots > < t_1 >< x_1 >< y_1 > \ldots < t_k >< x_k >< y_k >$

In particular, times $t1, \ldots ..., t_k$ have to be ordered (ascending order).

Example 1 *The same line is broken for display purposes.*

```
0 4 0.0 9933.0 8551.46 2.67 9944.38 8437.65 5.33 9963.98
8324.4 8.0 9961.1 8209.65
```

4 Output format

The T-Pattern miner generates two files: `MiSTA.output` and `regions.output`.

The first file (`MiSTA.output`) contains the list of extracted frequent patterns ordered by length with relative temporal annotations, except the patterns of length 1 that have not temporal annotation (they contain only 1 region, so there is no transition at all). For each pattern, there is a line that describes the pattern with its relative support and absolute support. Then, such line is followed by the temporal annotations (transition times) associated with it, one set of intervals per line, described by their lower and upper bounds, with the corresponding frequency (also called density, in this context).

Example 2 *This example shows a segment of file MiSTA.otuput, related to pattern "(0)→(9)" and its transition times:*

```
(0) (9) :  0.5 [abs:126]
[542.87, 544.6] Density:  76
[545.4, 547.72] Density:  76
```

The second file (`regions.output`) contains the description of the regions extracted for each prefix: the ID of the region, followed by its lower-left and upper-right coordinates in a 2D space.

Example 3 *This example shows the content of regions.output, assuming it contains 3 regions:*

```
18 59 38 59 38
19 58 38 59 39
20 56 38 56 40
```

References

[1] F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. Trajectory Pattern Mining. In *Proc. of ACM International Conference on Knowledge Discovery and Data Mining (KDD 2007)*, 2007.

[2] F. Giannotti, M. Nanni, and D. Pedreschi. Efficient mining of sequences with temporal annotations. In *Proc. SIAM Conference on Data Mining*, pages 346–357. SIAM, 2006.

[3] F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. Mining Sequences with Temporal Annotations. In *Proc. of ACM SAC 2006 - DM Track*, 2006.