**Mining Long Shareable Patterns in Trajectories of Moving Objects**

Győző Gidofalvi and Torben Bach Pedersen

**Identification of trips**

But why do I even bother? What is it for me?

For clarity the time-of-day domain is projected down to the 2D image. Lines represent trips performed by a single object. The number of time the trip was performed is represented by the width of the line (also written in parenthesis in the legend).

**STEP 1:** Filter infrequent items!

**DEF:** An item (ST-region) is frequent if the number of transactions (trips) that contain that item is at least some threshold, i.e., larger than Min supp (for example 4), and the number of unique objects associated with those transactions is also larger than Min length (for example 2).

**SQL statements to filter infrequent items**

```
INSERT INTO T (item, count) SELECT item, count(*) FROM T 
WHERE item IN (SELECT item FROM FT_i)
```

**STEP 2:** Filter short transactions!

**DEF:** A transaction is short if the number of items in it is less than Min length (for example 4)

**SQL statement to filter short transactions**

```
SELECT item, count(*) FROM T GROUP BY item
HAVING COUNT(item) >= MinLength
```

**STEP 3 & 4:** Project DB and find the single most frequent itemset!

**DEF:** An item-projected DB, T_i, contains all the items from the transactions (trips) that contain that item and its items all have maximal support in T_i.

There is a single most frequent itemset in item-projected DB, and its items all have maximal support in T_i.

**SQL statement to construct item-projected DB**

```
INSERT INTO T_i (oid, tid, item) 
SELECT T.oid, T.tid, T.item FROM T, F 
WHERE T.item = F.item
```

**STEP 5:** Delete unnecessary items from predecessor DB!

An item that has the same support in the projected DB as in the predecessor DB can be deleted from the latter.

**SQL statements to delete unnecessary items**

```
DELETE FROM FT_i WHERE item IN (SELECT item FROM F)
```

My experiments show, that:

1. the SQL implementation of my algorithm is effective
2. my algorithm behaves as expected when I vary its parameters

We all learned something today,...

I learned that such an algorithm has many uses in moving object database management, Location-Based Services, and Location-Based Advertising.

And I am goin’ HOME!