IX Brazilian Symposium on GeoInformatics

Trajectory Data Warehouses: Proposal of Design and Application to Exploit Data

Fernando José Braz[1,2]

[1]Department of Informatics – University of Region of Joinville
Joinville, Brazil.
[2]Department of Computer Science – Ca' Foscari University
Venice, Italy.

November 25, 2007

Outline

Introduction

- The movement of a spatio-temporal object *o* - *trajectory* - can be represented by a finite set of *observations*.
- The trajectory can be represented by position (*X* and *Y*) and time data.
    - Each point of the trajectory is represented by a tuple *(id, x, y, t)* corresponding to an object *id* in a location *(x, y)* at time *t*.
    - A set of observations represents data about several moving objects positions.
- On a data warehouse model, the data are summarized and aggregated in a multidimensional way in order to facilitate access and data analysis.
- A *Trajectory Data Warehouse* must be capable to receive, compute and store data and measures in order to improve the analysis of a set of trajectories of moving objects.

Introduction

- The movement of a spatio-temporal object *o* - *trajectory* - can be represented by a finite set of *observations*.
- The trajectory can be represented by position (*X* and *Y*) and time data.
  - Each point of the trajectory is represented by a tuple *(id, x, y, t)* corresponding to an object *id* in a location *(x, y)* at time *t*.
  - A set of observations represents data about several moving objects positions.
- On a data warehouse model, the data are summarized and aggregated in a multidimensional way in order to facilitate access and data analysis.
- A *Trajectory Data Warehouse* must be capable to receive, compute and store data and measures in order to improve the analysis of a set of trajectories of moving objects.

Introduction

- The movement of a spatio-temporal object *o* - *trajectory* - can be represented by a finite set of *observations*.
- The trajectory can be represented by position (*X* and *Y*) and time data.
  - Each point of the trajectory is represented by a tuple *(id, x, y, t)* corresponding to an object *id* in a location *(x, y)* at time *t*.
  - A set of observations represents data about several moving objects positions.
- On a data warehouse model, the data are summarized and aggregated in a multidimensional way in order to facilitate access and data analysis.
- A *Trajectory Data Warehouse* must be capable to receive, compute and store data and measures in order to improve the analysis of a set of trajectories of moving objects.

Introduction

- The movement of a spatio-temporal object *o* - *trajectory* - can be represented by a finite set of *observations*.
- The trajectory can be represented by position (*X* and *Y*) and time data.
  - Each point of the trajectory is represented by a tuple *(id, x, y, t)* corresponding to an object *id* in a location *(x, y)* at time *t*.
  - A set of observations represents data about several moving objects positions.
- On a data warehouse model, the data are summarized and aggregated in a multidimensional way in order to facilitate access and data analysis.
- A *Trajectory Data Warehouse* must be capable to receive, compute and store data and measures in order to improve the analysis of a set of trajectories of moving objects.

Introduction

- The movement of a spatio-temporal object *o* - *trajectory* - can be represented by a finite set of *observations*.
- The trajectory can be represented by position (*X* and *Y*) and time data.
  - Each point of the trajectory is represented by a tuple *(id, x, y, t)* corresponding to an object *id* in a location *(x, y)* at time *t*.
  - A set of observations represents data about several moving objects positions.
- On a data warehouse model, the data are summarized and aggregated in a multidimensional way in order to facilitate access and data analysis.
- A *Trajectory Data Warehouse* must be capable to receive, compute and store data and measures in order to improve the analysis of a set of trajectories of moving objects.

Introduction

- The movement of a spatio-temporal object *o* - *trajectory* - can be represented by a finite set of *observations*.
- The trajectory can be represented by position (*X* and *Y*) and time data.
  - Each point of the trajectory is represented by a tuple *(id, x, y, t)* corresponding to an object *id* in a location *(x, y)* at time *t*.
  - A set of observations represents data about several moving objects positions.
- On a data warehouse model, the data are summarized and aggregated in a multidimensional way in order to facilitate access and data analysis.
- A *Trajectory Data Warehouse* must be capable to receive, compute and store data and measures in order to improve the analysis of a set of trajectories of moving objects.

Problems

- To define a model to store the trajectories data using the current data warehouse systems.
- To develop mechanisms in order to receive, compute and store the trajectories data.
  - Produced at different rates
  - Arrive in streams in an unbounded way
- To compute the super-aggregate values, in a data stream environment, preserving the usage of the limited available resources and improving the accuracy of the results.
- To develop a tool to allow the definition and implementation of a *Trajectory Data Warehouse* considering the capability of receiving and computation of a data stream environment.

Problems

- To define a model to store the trajectories data using the current data warehouse systems.
- To develop mechanisms in order to receive, compute and store the trajectories data.
  - Produced at different rates
  - Arrive in streams in an unbounded way
- To compute the super-aggregate values, in a data stream environment, preserving the usage of the limited available resources and improving the accuracy of the results.
- To develop a tool to allow the definition and implementation of a *Trajectory Data Warehouse* considering the capability of receiving and computation of a data stream environment.

Problems

- To define a model to store the trajectories data using the current data warehouse systems.
- To develop mechanisms in order to receive, compute and store the trajectories data.
  - Produced at different rates
  - Arrive in streams in an unbounded way
- To compute the super-aggregate values, in a data stream environment, preserving the usage of the limited available resources and improving the accuracy of the results.
- To develop a tool to allow the definition and implementation of a *Trajectory Data Warehouse* considering the capability of receiving and computation of a data stream environment.

Problems

- To define a model to store the trajectories data using the current data warehouse systems.
- To develop mechanisms in order to receive, compute and store the trajectories data.
  - Produced at different rates
  - Arrive in streams in an unbounded way
- To compute the super-aggregate values, in a data stream environment, preserving the usage of the limited available resources and improving the accuracy of the results.
- To develop a tool to allow the definition and implementation of a *Trajectory Data Warehouse* considering the capability of receiving and computation of a data stream environment.

Problems

- To define a model to store the trajectories data using the current data warehouse systems.
- To develop mechanisms in order to receive, compute and store the trajectories data.
  - Produced at different rates
  - Arrive in streams in an unbounded way
- To compute the super-aggregate values, in a data stream environment, preserving the usage of the limited available resources and improving the accuracy of the results.
- To develop a tool to allow the definition and implementation of a *Trajectory Data Warehouse* considering the capability of receiving and computation of a data stream environment.

Problems

- To define a model to store the trajectories data using the current data warehouse systems.
- To develop mechanisms in order to receive, compute and store the trajectories data.
  - Produced at different rates
  - Arrive in streams in an unbounded way
- To compute the super-aggregate values, in a data stream environment, preserving the usage of the limited available resources and improving the accuracy of the results.
- To develop a tool to allow the definition and implementation of a *Trajectory Data Warehouse* considering the capability of receiving and computation of a data stream environment.

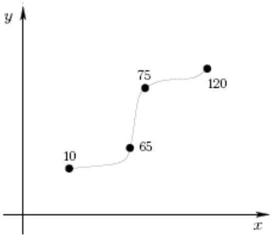
Outline

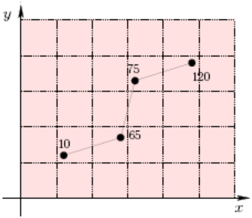
Environment

We are considering a trajectory as the movement of a object (along the time) in 2D discretized space divided into a regular grid (*cells*).

The data arrive on an unpredictable and continuous way.

We are considering the model proposed by [1].





Outline

Loading

- Begins at the base cells of the base cuboid, with suitable *sub-aggregate* measures, from which starting to compute *super-aggregated* functions.
- The reconstruction of the trajectories is done by the usage of the *linear interpolation*.
- The interpolation infers additional spatio-temporal locations of intermediate points, these points occur between two known trajectory observations.
- Before the interpolation some base cells could be traversed by the trajectories but, since no observation falls in them, they not appear in the *fact* table.
- The interpolation computes the additional points for each cell traversed by a trajectory.
  - Computed considering the intersections between the trajectory and the border of the spatio-temporal cells.

Loading

- Begins at the base cells of the base cuboid, with suitable *sub-aggregate* measures, from which starting to compute *super-aggregated* functions.
- The reconstruction of the trajectories is done by the usage of the *linear interpolation*.
- The interpolation infers additional spatio-temporal locations of intermediate points, these points occur between two known trajectory observations.
- Before the interpolation some base cells could be traversed by the trajectories but, since no observation falls in them, they not appear in the *fact* table.
- The interpolation computes the additional points for each cell traversed by a trajectory.
    - Computed considering the intersections between the trajectory and the border of the spatio-temporal cells.

Loading

- Begins at the base cells of the base cuboid, with suitable *sub-aggregate* measures, from which starting to compute *super-aggregated* functions.
- The reconstruction of the trajectories is done by the usage of the *linear interpolation*.
- The interpolation infers additional spatio-temporal locations of intermediate points, these points occur between two known trajectory observations.
- Before the interpolation some base cells could be traversed by the trajectories but, since no observation falls in them, they not appear in the *fact* table.
- The interpolation computes the additional points for each cell traversed by a trajectory.
  - Computed considering the intersections between the trajectory and the border of the spatio-temporal cells.

Loading

- Begins at the base cells of the base cuboid, with suitable *sub-aggregate* measures, from which starting to compute *super-aggregated* functions.
- The reconstruction of the trajectories is done by the usage of the *linear interpolation*.
- The interpolation infers additional spatio-temporal locations of intermediate points, these points occur between two known trajectory observations.
- Before the interpolation some base cells could be traversed by the trajectories but, since no observation falls in them, they not appear in the *fact* table.
- The interpolation computes the additional points for each cell traversed by a trajectory.
  - Computed considering the intersections between the trajectory and the border of the spatio-temporal cells.

Loading

- Begins at the base cells of the base cuboid, with suitable *sub-aggregate* measures, from which starting to compute *super-aggregated* functions.
- The reconstruction of the trajectories is done by the usage of the *linear interpolation*.
- The interpolation infers additional spatio-temporal locations of intermediate points, these points occur between two known trajectory observations.
- Before the interpolation some base cells could be traversed by the trajectories but, since no observation falls in them, they not appear in the *fact* table.
- The interpolation computes the additional points for each cell traversed by a trajectory.
  - Computed considering the intersections between the trajectory and the border of the spatio-temporal cells.

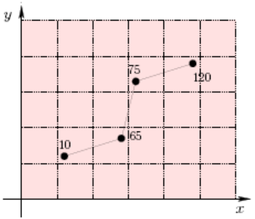
Loading

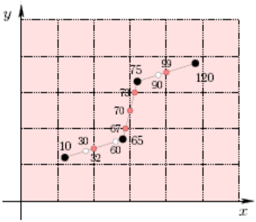
- Begins at the base cells of the base cuboid, with suitable *sub-aggregate* measures, from which starting to compute *super-aggregated* functions.
- The reconstruction of the trajectories is done by the usage of the *linear interpolation*.
- The interpolation infers additional spatio-temporal locations of intermediate points, these points occur between two known trajectory observations.
- Before the interpolation some base cells could be traversed by the trajectories but, since no observation falls in them, they not appear in the *fact* table.
- The interpolation computes the additional points for each cell traversed by a trajectory.
  - Computed considering the intersections between the trajectory and the border of the spatio-temporal cells.

Loading

The interpolated points, associated with temporal labels 30, 60, and 90, correspond to cross points of a temporal border of some 3D cell.

The points labeled with 32, 67, 70, 73, and 99, have been instead introduced to match the spatial dimensions.





Loading

Table: Cells representation - for each observation

Time label	X	Y	T	M_1	...	M_k
10	[30,60)	[30,60)	[0,30)	...	...	...
65	[60,90)	[30,60)	[60,90)	...	...	...
75	[90,120)	[90,120)	[60,90)	...	...	...
120	[120,150)	[90,120)	[60,120)	...	...	...

Table: Sequence of segments composing the interpolated trajectory.

$(\mathbf{t}_i, \mathbf{t}_{i+1})$	X	Y	T	M_1	...	M_k
(10,30)	[30,60)	[30,60)	[0,30)	...	...	...
(30,32)	[30,60)	[30,60)	[30,60)	...	...	...
(32,60)	[30,60)	[30,60)	[30,60)	...	...	...
(60,65)	[60,90)	[30,60)	[60,90)	...	...	...
(65,67)	[60,90)	[30,60)	[60,90)	...	...	...
(67,70)	[60,90)	[90,120)	[60,90)	...	...	...
(70,73)	[60,90)	[90,120)	[60,90)	...	...	...
(73,75)	[60,90)	[120,150)	[60,90)	...	...	...
(75,90)	[90,120)	[90,120)	[60,90)	...	...	...
(90,99)	[120,150)	[120,150)	[90,120)	...	...	...
(99,120)	[150,180)	[120,150)	[90,120)	...	...	...

Outline

Aggregation

The aggregate functions can be classified based on the space complexity for computing a super-aggregate starting from a set of sub-aggregates previously computed ([4]):

- *Distributive*: The super-aggregates can be computed from the sub-aggregates (*sum, min, max, count*).
- *Algebraic*: The super-aggregates can be computed from the sub-aggregates together with a finite set of auxiliary measures (*average, variance, standard deviation*).
- *Holistic*: The super-aggregates cannot be computed from the sub-aggregates, not even using any finite number of auxiliary measures (*median, most frequent, rank*).

Aggregation

The aggregate functions can be classified based on the space complexity for computing a super-aggregate starting from a set of sub-aggregates previously computed ([4]):

- *Distributive*: The super-aggregates can be computed from the sub-aggregates (*sum, min, max, count*).
- *Algebraic*: The super-aggregates can be computed from the sub-aggregates together with a finite set of auxiliary measures (*average, variance, standard deviation*).
- *Holistic*: The super-aggregates cannot be computed from the sub-aggregates, not even using any finite number of auxiliary measures (*median, most frequent, rank*).

Aggregation

The aggregate functions can be classified based on the space complexity for computing a super-aggregate starting from a set of sub-aggregates previously computed ([4]):

- *Distributive*: The super-aggregates can be computed from the sub-aggregates (*sum, min, max, count*).
- *Algebraic*: The super-aggregates can be computed from the sub-aggregates together with a finite set of auxiliary measures (*average, variance, standard deviation*).
- *Holistic*: The super-aggregates cannot be computed from the sub-aggregates, not even using any finite number of auxiliary measures (*median, most frequent, rank*).

Aggregation

Table: Numeric measures

m1	*numobs*	*Number of observations in the cell*
m2	*trajinit*	*Number of trajectories starting in the cell*
m3	*presence*	*Number of trajectories in the cell*
m4	*distance*	*Total distance covered by trajectories in the cell*
m5	*speed*	*Average speed of trajectories in the cell*
m6	v_{max}	*Maximum speed of trajectories in the cell*

- The computation of the super-aggregates for the measures *m1, m2, m4* and *m6* uses *distributive* aggregate functions.
- The super-aggregate for the measure *m5* is *algebraic*, it is necessary to compute an auxiliary measure in order to compute the aggregate function.
- The aggregate function for *m3* is *holistic*, then is necessary to compute the measure in an approximated way.

Aggregation

Table: Numeric measures

m1	*numobs*	*Number of observations in the cell*
m2	*trajinit*	*Number of trajectories starting in the cell*
m3	*presence*	*Number of trajectories in the cell*
m4	*distance*	*Total distance covered by trajectories in the cell*
m5	*speed*	*Average speed of trajectories in the cell*
m6	v_{max}	*Maximum speed of trajectories in the cell*

- The computation of the super-aggregates for the measures *m1, m2, m4* and *m6* uses *distributive* aggregate functions.
- The super-aggregate for the measure *m5* is *algebraic*, it is necessary to compute an auxiliary measure in order to compute the aggregate function.
- The aggregate function for *m3* is *holistic*, then is necessary to compute the measure in an approximated way.

Aggregation

Table: Numeric measures

m1	*numobs*	*Number of observations in the cell*
m2	*trajinit*	*Number of trajectories starting in the cell*
m3	*presence*	*Number of trajectories in the cell*
m4	*distance*	*Total distance covered by trajectories in the cell*
m5	*speed*	*Average speed of trajectories in the cell*
m6	v_{max}	*Maximum speed of trajectories in the cell*

- The computation of the super-aggregates for the measures *m1, m2, m4* and *m6* uses *distributive* aggregate functions.
- The super-aggregate for the measure *m5* is *algebraic*, it is necessary to compute an auxiliary measure in order to compute the aggregate function.
- The aggregate function for *m3* is *holistic*, then is necessary to compute the measure in an approximated way.

Aggregation - *Presence Function*

The *Presence* function represents the count of the *distinct* trajectories crossing a given cell. *Holistic* function.

- A few/constant memory size for maintaining the information associated with each base cell for the Data Warehouse.
- $Presence_{Distributive}$: The exact (or approximate) count of all the *distinct* trajectories. Super-aggregate (roll-up operation) by summing up all the measures associated with cell (*duplicate counting*).
- $Presence_{Algebraic}$: Each base cell stores an *M*-tuple of measures.
  - The exact (or approximate) count of all the *distinct* trajectories touching the cell.
  - Measures to compute the super-aggregate, in order to correct the errors introduces by function $Presence_{Distributive}$ (*duplicate counting*).

Aggregation - *Presence Function*

The *Presence* function represents the count of the *distinct* trajectories crossing a given cell. *Holistic* function.

- A few/constant memory size for maintaining the information associated with each base cell for the Data Warehouse.
- $Presence_{Distributive}$: The exact (or approximate) count of all the *distinct* trajectories. Super-aggregate (roll-up operation) by summing up all the measures associated with cell (*duplicate counting*).
- $Presence_{Algebraic}$: Each base cell stores an *M*-tuple of measures.
  - The exact (or approximate) count of all the *distinct* trajectories touching the cell.
  - Measures to compute the super-aggregate, in order to correct the errors introduces by function $Presence_{Distributive}$ (*duplicate counting*).

Aggregation - *Presence Function*

The *Presence* function represents the count of the *distinct* trajectories crossing a given cell. *Holistic* function.

- A few/constant memory size for maintaining the information associated with each base cell for the Data Warehouse.
- $Presence_{Distributive}$: The exact (or approximate) count of all the *distinct* trajectories. Super-aggregate (roll-up operation) by summing up all the measures associated with cell (*duplicate counting*).
- $Presence_{Algebraic}$: Each base cell stores an *M*-tuple of measures.
  - The exact (or approximate) count of all the *distinct* trajectories touching the cell.
  - Measures to compute the super-aggregate, in order to correct the errors introduces by function $Presence_{Distributive}$ (*duplicate counting*).

Aggregation - *Presence Function*

The *Presence* function represents the count of the *distinct* trajectories crossing a given cell. *Holistic* function.

- A few/constant memory size for maintaining the information associated with each base cell for the Data Warehouse.
- *Presence$_{Distributive}$*: The exact (or approximate) count of all the *distinct* trajectories. Super-aggregate (roll-up operation) by summing up all the measures associated with cell (*duplicate counting*).
- *Presence$_{Algebraic}$*: Each base cell stores an *M*-tuple of measures.
  - The exact (or approximate) count of all the *distinct* trajectories touching the cell.
  - Measures to compute the super-aggregate, in order to correct the errors introduces by function *Presence$_{Distributive}$* (*duplicate counting*).

Aggregation - *Presence Function*

The *Presence* function represents the count of the *distinct* trajectories crossing a given cell. *Holistic* function.

- A few/constant memory size for maintaining the information associated with each base cell for the Data Warehouse.
- $Presence_{Distributive}$: The exact (or approximate) count of all the *distinct* trajectories. Super-aggregate (roll-up operation) by summing up all the measures associated with cell (*duplicate counting*).
- $Presence_{Algebraic}$: Each base cell stores an *M*-tuple of measures.
  - The exact (or approximate) count of all the *distinct* trajectories touching the cell.
  - Measures to compute the super-aggregate, in order to correct the errors introduces by function $Presence_{Distributive}$ (*duplicate counting*).

$Presence_{Algebraic}$

Let $C_{x,y,t}$ be a generic base cell of our cuboid, where x, y, and t identify intervals of the form $[l, u)$, in which we have subdivided the spatial and temporal dimensions.
The associated measures are thus $C_{x,y,t}.presence$, $C_{x,y,t}.crossX$, $C_{x,y,t}.crossY$, and $C_{x,y,t}.crossT$.

- $C_{x,y,t}.presence$ is the count of all the *distinct* trajectories crossing the cell.
- $C_{x,y,t}.crossX$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x+1,y,t}$.
- $C_{x,y,t}.crossY$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x,y+1,t}$.
- $C_{x,y,t}.crossT$ is the number of *distinct* trajectories crossing the *temporal* border between $C_{x,y,t}$ and $C_{x,y,t+1}$

$Presence_{Algebraic}$

Let $C_{x,y,t}$ be a generic base cell of our cuboid, where x, y, and t identify intervals of the form $[l, u)$, in which we have subdivided the spatial and temporal dimensions.
The associated measures are thus $C_{x,y,t}.presence$, $C_{x,y,t}.crossX$, $C_{x,y,t}.crossY$, and $C_{x,y,t}.crossT$.

- $C_{x,y,t}.presence$ is the count of all the *distinct* trajectories crossing the cell.
- $C_{x,y,t}.crossX$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x+1,y,t}$.
- $C_{x,y,t}.crossY$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x,y+1,t}$.
- $C_{x,y,t}.crossT$ is the number of *distinct* trajectories crossing the *temporal* border between $C_{x,y,t}$ and $C_{x,y,t+1}$

$Presence_{Algebraic}$

Let $C_{x,y,t}$ be a generic base cell of our cuboid, where x, y, and t identify intervals of the form $[l, u)$, in which we have subdivided the spatial and temporal dimensions.
The associated measures are thus $C_{x,y,t}.presence$, $C_{x,y,t}.crossX$, $C_{x,y,t}.crossY$, and $C_{x,y,t}.crossT$.

- $C_{x,y,t}.presence$ is the count of all the *distinct* trajectories crossing the cell.
- $C_{x,y,t}.crossX$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x+1,y,t}$.
- $C_{x,y,t}.crossY$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x,y+1,t}$.
- $C_{x,y,t}.crossT$ is the number of *distinct* trajectories crossing the *temporal* border between $C_{x,y,t}$ and $C_{x,y,t+1}$

$Presence_{Algebraic}$

Let $C_{x,y,t}$ be a generic base cell of our cuboid, where x, y, and t identify intervals of the form $[l, u)$, in which we have subdivided the spatial and temporal dimensions.
The associated measures are thus $C_{x,y,t}.presence$, $C_{x,y,t}.crossX$, $C_{x,y,t}.crossY$, and $C_{x,y,t}.crossT$.

- $C_{x,y,t}.presence$ is the count of all the *distinct* trajectories crossing the cell.
- $C_{x,y,t}.crossX$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x+1,y,t}$.
- $C_{x,y,t}.crossY$ is the number of *distinct* trajectories crossing the *spatial* border between $C_{x,y,t}$ and $C_{x,y+1,t}$.
- $C_{x,y,t}.crossT$ is the number of *distinct* trajectories crossing the *temporal* border between $C_{x,y,t}$ and $C_{x,y,t+1}$

$Presence_{Algebraic}$

In order to compute the super-aggregate corresponding to two adjacent cells with respect to a given dimension, namely $C_{x',y',t'} = C_{x,y,t} \cup C_{x+1,y,t}$, we can compute it as follows:

- $Presence_{Algebraic}(C_{x,y,t} \cup C_{x+1,y,t}) = C_{x,y,t}.presence + C_{x+1,y,t}.presence - C_{x,y,t}.crossX$

Prototype

- We have developed our application using the synthetic datasets generated by the traffic simulator described in [2].
- To solve the problems of *loading* and *aggregation*.
- *TDW* was implemented in a traditional data warehouse tool (*MS SQL SERVER 2005*).
  - Modeled in agreement with the *star* model [5]
  - A fact table and three dimension tables (*X* and *Y spatial* dimensions and *T* temporal dimension).

Prototype

- We have developed our application using the synthetic datasets generated by the traffic simulator described in [2].
- To solve the problems of *loading* and *aggregation*.
- *TDW* was implemented in a traditional data warehouse tool (*MS SQL SERVER 2005*).
  - Modeled in agreement with the *star* model [5]
  - A fact table and three dimension tables (*X* and *Y spatial* dimensions and *T* temporal dimension).

Prototype

- We have developed our application using the synthetic datasets generated by the traffic simulator described in [2].
- To solve the problems of *loading* and *aggregation*.
- *TDW* was implemented in a traditional data warehouse tool (*MS SQL SERVER 2005*).
  - Modeled in agreement with the *star* model [5]
  - A fact table and three dimension tables (*X* and *Y spatial* dimensions and *T* temporal dimension).

Prototype

- We have developed our application using the synthetic datasets generated by the traffic simulator described in [2].
- To solve the problems of *loading* and *aggregation*.
- *TDW* was implemented in a traditional data warehouse tool (*MS SQL SERVER 2005*).
  - Modeled in agreement with the *star* model [5]
  - A fact table and three dimension tables (*X* and *Y spatial* dimensions and *T* temporal dimension).

Prototype

- We have developed our application using the synthetic datasets generated by the traffic simulator described in [2].
- To solve the problems of *loading* and *aggregation*.
- *TDW* was implemented in a traditional data warehouse tool (*MS SQL SERVER 2005*).
    - Modeled in agreement with the *star* model [5]
    - A fact table and three dimension tables (*X* and *Y spatial* dimensions and *T* temporal dimension).

Prototype - Fact Table

Table: Fact Table

tid	*time foreign key*
xid	*X spatial foreign key*
yid	*Y spatial foreign key*
numobs	*Number of observations in the cell*
trajinit	*Number of trajectories starting in the cell*
vmax	*Maximum speed of trajectories in the cell*
distance	*Total distance covered by trajectories in the cell*
time	*Total time spend by the trajectories in the cell*
presence	*Number of trajectories in the cell - distributive*
xborder	*Number of trajectories crossing the x cell border*
yborder	*Number of trajectories crossing the y cell border*
tborder	*Number of trajectories crossing the t cell border*
speed	*Average speed of trajectories in the cell*

- Each tuple represents a summarization of the measures that are delimited by the borders of the *cell*.
- The *base cell* are delimited by the *tid, xid* and *yid* values.
- Measures *presence, xborder, yborder* and *tborder* to compute the *holistic presence* measure.

Prototype - Fact Table

Table: Fact Table

tid	*time foreign key*
xid	*X spatial foreign key*
yid	*Y spatial foreign key*
numobs	*Number of observations in the cell*
trajinit	*Number of trajectories starting in the cell*
vmax	*Maximum speed of trajectories in the cell*
distance	*Total distance covered by trajectories in the cell*
time	*Total time spend by the trajectories in the cell*
presence	*Number of trajectories in the cell - distributive*
xborder	*Number of trajectories crossing the x cell border*
yborder	*Number of trajectories crossing the y cell border*
tborder	*Number of trajectories crossing the t cell border*
speed	*Average speed of trajectories in the cell*

- Each tuple represents a summarization of the measures that are delimited by the borders of the *cell*.
- The *base cell* are delimited by the *tid, xid* and *yid* values.
- Measures *presence, xborder, yborder* and *tborder* to compute the *holistic presence* measure.

Prototype - Fact Table

Table: Fact Table

tid	*time foreign key*
xid	*X spatial foreign key*
yid	*Y spatial foreign key*
numobs	*Number of observations in the cell*
trajinit	*Number of trajectories starting in the cell*
vmax	*Maximum speed of trajectories in the cell*
distance	*Total distance covered by trajectories in the cell*
time	*Total time spend by the trajectories in the cell*
presence	*Number of trajectories in the cell - distributive*
xborder	*Number of trajectories crossing the x cell border*
yborder	*Number of trajectories crossing the y cell border*
tborder	*Number of trajectories crossing the t cell border*
speed	*Average speed of trajectories in the cell*

- Each tuple represents a summarization of the measures that are delimited by the borders of the *cell*.
- The *base cell* are delimited by the *tid, xid* and *yid* values.
- Measures *presence, xborder, yborder* and *tborder* to compute the *holistic presence* measure.

Outline

Loader Component

By the *loader* component is possible to define the details of the environment and to compute the *interpolation* procedure.

- The settings of the environment in order to receive the data volume.
  - *Granularity level*
  - *Dimension Hierarchical level*
- Loads the data volume into a buffer table.
- The process happens in an unpredictable and unbounded way, therefore we have to store packages of data into a *buffer* table.
- To release space in the *buffer* table by the exclusion of tuples of the trajectories ended.

Loader Component

By the *loader* component is possible to define the details of the environment and to compute the *interpolation* procedure.

- The settings of the environment in order to receive the data volume.
  - *Granularity level*
  - *Dimension Hierarchical level*
- Loads the data volume into a buffer table.
- The process happens in an unpredictable and unbounded way, therefore we have to store packages of data into a *buffer* table.
- To release space in the *buffer* table by the exclusion of tuples of the trajectories ended.

Loader Component

By the *loader* component is possible to define the details of the environment and to compute the *interpolation* procedure.

- The settings of the environment in order to receive the data volume.
  - *Granularity level*
  - *Dimension Hierarchical level*
- Loads the data volume into a buffer table.
- The process happens in an unpredictable and unbounded way, therefore we have to store packages of data into a *buffer* table.
- To release space in the *buffer* table by the exclusion of tuples of the trajectories ended.

Loader Component

By the *loader* component is possible to define the details of the environment and to compute the *interpolation* procedure.

- The settings of the environment in order to receive the data volume.
  - *Granularity level*
  - *Dimension Hierarchical level*
- Loads the data volume into a buffer table.
- The process happens in an unpredictable and unbounded way, therefore we have to store packages of data into a *buffer* table.
- To release space in the *buffer* table by the exclusion of tuples of the trajectories ended.

Loader Component

By the *loader* component is possible to define the details of the environment and to compute the *interpolation* procedure.

- The settings of the environment in order to receive the data volume.
  - *Granularity level*
  - *Dimension Hierarchical level*
- Loads the data volume into a buffer table.
- The process happens in an unpredictable and unbounded way, therefore we have to store packages of data into a *buffer* table.
- To release space in the *buffer* table by the exclusion of tuples of the trajectories ended.

Loader Component

By the *loader* component is possible to define the details of the environment and to compute the *interpolation* procedure.

- The settings of the environment in order to receive the data volume.
  - *Granularity level*
  - *Dimension Hierarchical level*
- Loads the data volume into a buffer table.
- The process happens in an unpredictable and unbounded way, therefore we have to store packages of data into a *buffer* table.
- To release space in the *buffer* table by the exclusion of tuples of the trajectories ended.

Loader - *Buffer table*

For each *active* row of the *buffer table* the application has to find the related *cell* in the *TDW*. If exists a row in the fact table for that *cell*, the values of *distributive* measures (e.g *numobs, trajinit*) can be updated, else a new row will be inserted in the fact table.

Table: Buffer Table

oid	*Object Identifier*
xvalue	*X spatial value*
yvalue	*Y spatial value*
tvalue	*T time value*
dift	*Time variation between two consecutive positions*
difx	*X spatial variation between two consecutive positions*
dify	*Y spatial variation between two consecutive positions*
dist	*Distance covered between two consecutive positions*
vel	*Speed between two consecutive positions*
idrow	*Identifier of the row*
timestamp	*Timestamp of the observation*

Outline

Aggregation Component

- The pre-aggregated values (computed in the *loading phase*) are stored in the *TDW*.
- The user can choose the query *base cell* either by the usage of the combo-boxes or by clicking on the map.
- The map is divided into a regular grid, this division is done in agreement with the granularity defined by the user.
- The most of the queries can be easily solved computing traditional *SQL* operations on the *fact table*, constrained by the cells borders.
- When the goal is to compute a *roll-up* operation on a *holistic* measure the prototype have to use some *stored procedures* defined in the *Data Warehouse* system.

Aggregation Component

- The pre-aggregated values (computed in the *loading phase*) are stored in the *TDW*.
- The user can choose the query *base cell* either by the usage of the combo-boxes or by clicking on the map.
- The map is divided into a regular grid, this division is done in agreement with the granularity defined by the user.
- The most of the queries can be easily solved computing traditional *SQL* operations on the *fact table*, constrained by the cells borders.
- When the goal is to compute a *roll-up* operation on a *holistic* measure the prototype have to use some *stored procedures* defined in the *Data Warehouse* system.

Aggregation Component

- The pre-aggregated values (computed in the *loading phase*) are stored in the *TDW*.
- The user can choose the query *base cell* either by the usage of the combo-boxes or by clicking on the map.
- The map is divided into a regular grid, this division is done in agreement with the granularity defined by the user.
- The most of the queries can be easily solved computing traditional *SQL* operations on the *fact table*, constrained by the cells borders.
- When the goal is to compute a *roll-up* operation on a *holistic* measure the prototype have to use some *stored procedures* defined in the *Data Warehouse* system.

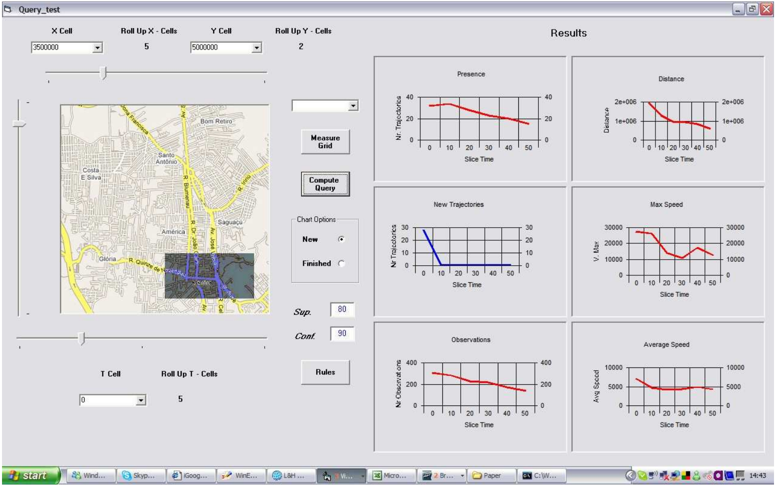
Aggregation Component

- The pre-aggregated values (computed in the *loading phase*) are stored in the *TDW*.
- The user can choose the query *base cell* either by the usage of the combo-boxes or by clicking on the map.
- The map is divided into a regular grid, this division is done in agreement with the granularity defined by the user.
- The most of the queries can be easily solved computing traditional *SQL* operations on the *fact table*, constrained by the cells borders.
- When the goal is to compute a *roll-up* operation on a *holistic* measure the prototype have to use some *stored procedures* defined in the *Data Warehouse* system.

Aggregation Component

- The pre-aggregated values (computed in the *loading phase*) are stored in the *TDW*.
- The user can choose the query *base cell* either by the usage of the combo-boxes or by clicking on the map.
- The map is divided into a regular grid, this division is done in agreement with the granularity defined by the user.
- The most of the queries can be easily solved computing traditional *SQL* operations on the *fact table*, constrained by the cells borders.
- When the goal is to compute a *roll-up* operation on a *holistic* measure the prototype have to use some *stored procedures* defined in the *Data Warehouse* system.

Query Form


Query_test
X Cell
Roll Up X - Cells
Y Cell
Roll Up Y - Cells
3500000
5
5000000
2
Results
Measure Grid
Compute Query
Chart Options
New
Finished
Sup.
80
Conf.
90
Rules
T Cell
Roll Up T - Cells
0
5
Presence
Nr. Trajectories
Slice Time
Distance
Slice Time
New Trajectories
Nr Trajectories
Slice Time
Max Speed
V. Max
Slice Time
Observations
Slice Time
Average Speed
Avg Speed
Slice Time
start
14:43

Mining Module

The knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data. [3]

Definition

Given a set of items $I = I_1, I_2, \ldots, I_m$ and a database of transactions $D = t_1, t_2, \ldots, t_n$ where $t_i = I_{i1}, I_{i2}, \ldots, I_{ik}$ and $I_{ij} \in I$, an ***association rule*** *is an implication of the form $X \Longrightarrow Y$ where $X, Y \subset I$ are sets of items called itemsets and $X \cap Y =$.*

Mining Module

- A pattern related to some measures (*average speed, acceleration, number of distinct trajectories...*) can reveal the occurrence of a *traffic jam* (*real phenomenon*), it can be used to anticipate the occurrence of another related phenomenon and to execute actions to prevent the consequent occurrence.
- The *association rules* task could be used in order to reveal the relationship among the measures stored for each cell

Mining Module

- A pattern related to some measures (*average speed, acceleration, number of distinct trajectories...*) can reveal the occurrence of a *traffic jam* (*real phenomenon*), it can be used to anticipate the occurrence of another related phenomenon and to execute actions to prevent the consequent occurrence.
- The *association rules* task could be used in order to reveal the relationship among the measures stored for each cell

Mining Module

- In the traditional *association rules* algorithms the goal is to identify the items which are related, the source are the records which represent the occurrence of the items for each transaction.
- The proposal is to transform (*discretization process*) each row of the fact table in a transaction in order to execute the *association rules* task.
- We have implemented the data mining module using the *LCM* algorithm ([6]).
- We are not considering the dimension time in the current implementation, the our first goal is to identify the best model in order to adapt the *TDW* data to use the *association rules* techniques.

Mining Module

- In the traditional *association rules* algorithms the goal is to identify the items which are related, the source are the records which represent the occurrence of the items for each transaction.
- The proposal is to transform (*discretization process*) each row of the fact table in a transaction in order to execute the *association rules* task.
- We have implemented the data mining module using the *LCM* algorithm ([6]).
- We are not considering the dimension time in the current implementation, the our first goal is to identify the best model in order to adapt the *TDW* data to use the *association rules* techniques.

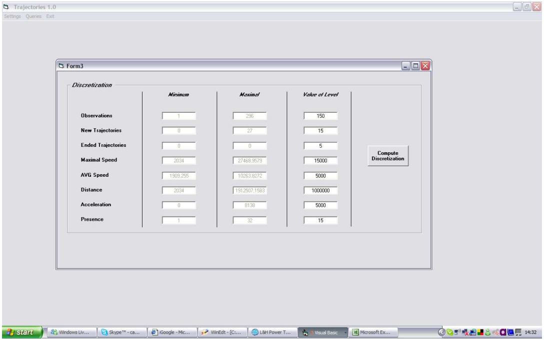
Mining Module

- In the traditional *association rules* algorithms the goal is to identify the items which are related, the source are the records which represent the occurrence of the items for each transaction.
- The proposal is to transform (*discretization process*) each row of the fact table in a transaction in order to execute the *association rules* task.
- We have implemented the data mining module using the *LCM* algorithm ([6]).
- We are not considering the dimension time in the current implementation, the our first goal is to identify the best model in order to adapt the *TDW* data to use the *association rules* techniques.

Mining Module

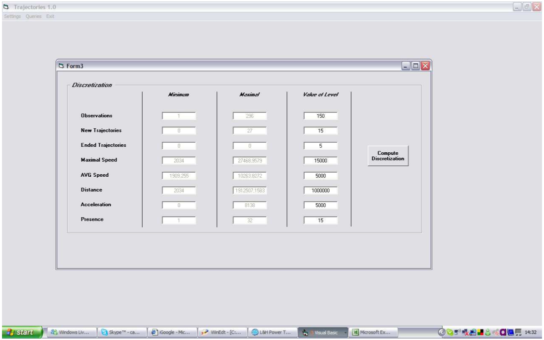
- In the traditional *association rules* algorithms the goal is to identify the items which are related, the source are the records which represent the occurrence of the items for each transaction.
- The proposal is to transform (*discretization process*) each row of the fact table in a transaction in order to execute the *association rules* task.
- We have implemented the data mining module using the *LCM* algorithm ([6]).
- We are not considering the dimension time in the current implementation, the our first goal is to identify the best model in order to adapt the *TDW* data to use the *association rules* techniques.

Mining Module - Discretization



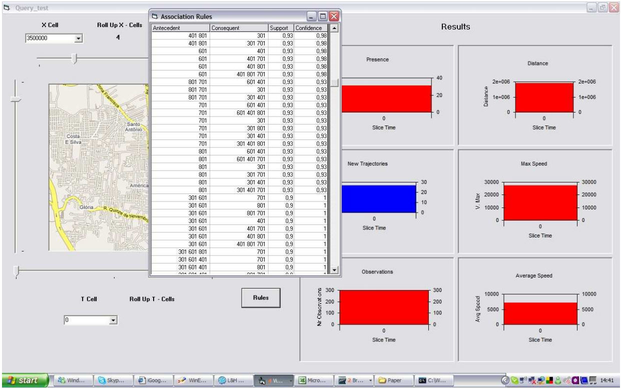
- The user, considering the limit values (minimum and maximal) of measures presented, can set the values of range for discretization, one value for each measure.
- This procedure can be executed at any time, the values of discretization are stored in a temporary buffer, just for the computation of the *patterns*.

Mining Module - Discretization



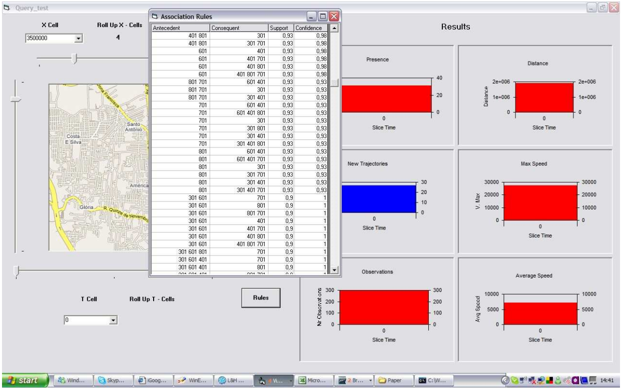
- The user, considering the limit values (minimum and maximal) of measures presented, can set the values of range for discretization, one value for each measure.
- This procedure can be executed at any time, the values of discretization are stored in a temporary buffer, just for the computation of the *patterns*.

Mining Module - Association Rules



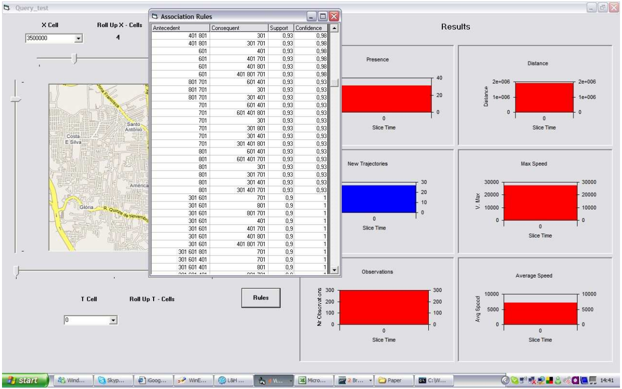
- *Patterns* for a cell or a range of cells.
- *Rules* in the format: *antecedent* → *consequent* (considering the level of discretization)
  - Example: the value 401 represents the interval [2034, 1002034) of the values of the *distance* measure.

Mining Module - Association Rules



- *Patterns* for a cell or a range of cells.
- *Rules* in the format: *antecedent* → *consequent* (considering the level of discretization)
  - Example: the value 401 represents the interval [2034, 1002034) of the values of the *distance* measure.

Mining Module - Association Rules



- *Patterns* for a cell or a range of cells.
- *Rules* in the format: *antecedent* → *consequent* (considering the level of discretization)
  - Example: the value 401 represents the interval [2034, 1002034) of the values of the *distance* measure.

Conclusions - Current/Future Work

- The current stage of the application can solve some problems of a trajectory data warehouse environment.
- The dimensions that we have used are very simple, a possible future work could be to sophisticate the hierarchy of the dimensions.
  - We did not consider a complete spatial dimension, in the preliminary research we have used just a spatial dimension in order to delimited the grid of the map.
  - There are another spatial objects which can be considered and implemented in this model: *line*, *polygon*, etc.
  - To define, design and implement dimensions with these spatial objects is a stimulant line of research.

Conclusions - Current/Future Work

- The current stage of the application can solve some problems of a trajectory data warehouse environment.
- The dimensions that we have used are very simple, a possible future work could be to sophisticate the hierarchy of the dimensions.
  - We did not consider a complete spatial dimension, in the preliminary research we have used just a spatial dimension in order to delimited the grid of the map.
  - There are another spatial objects which can be considered and implemented in this model: *line, polygon*, etc.
  - To define, design and implement dimensions with these spatial objects is a stimulant line of research.

Conclusions - Current/Future Work

- The current stage of the application can solve some problems of a trajectory data warehouse environment.
- The dimensions that we have used are very simple, a possible future work could be to sophisticate the hierarchy of the dimensions.
  - We did not consider a complete spatial dimension, in the preliminary research we have used just a spatial dimension in order to delimited the grid of the map.
  - There are another spatial objects which can be considered and implemented in this model: *line, polygon*, etc.
  - To define, design and implement dimensions with these spatial objects is a stimulant line of research.

Conclusions - Current/Future Work

- The current stage of the application can solve some problems of a trajectory data warehouse environment.
- The dimensions that we have used are very simple, a possible future work could be to sophisticate the hierarchy of the dimensions.
  - We did not consider a complete spatial dimension, in the preliminary research we have used just a spatial dimension in order to delimited the grid of the map.
  - There are another spatial objects which can be considered and implemented in this model: *line, polygon,* etc.
  - To define, design and implement dimensions with these spatial objects is a stimulant line of research.

Conclusions - Current/Future Work

- The current stage of the application can solve some problems of a trajectory data warehouse environment.
- The dimensions that we have used are very simple, a possible future work could be to sophisticate the hierarchy of the dimensions.
    - We did not consider a complete spatial dimension, in the preliminary research we have used just a spatial dimension in order to delimited the grid of the map.
    - There are another spatial objects which can be considered and implemented in this model: *line, polygon,* etc.
    - To define, design and implement dimensions with these spatial objects is a stimulant line of research.

Conclusions - Current/Future Work

- To develop another more complex measures in order to provide values to discover patterns or trend of the trajectories.
- In the current stage of our research we did not implement any *spatial* measure. It delimits the range of queries and analyzes which could be answered and computed.
  Therefore, to develop and implement some spatial measures to compute values along the *spatial* dimensions and hierarchies is very important point which must be researched.
- We have limited the loading phase considering a linear interpolation, but is possible to find some topological situations (e.g roads, bridges) where is very difficult to do the interpolation because of the some constraints in the movement of the object.

Conclusions - Current/Future Work

- To develop another more complex measures in order to provide values to discover patterns or trend of the trajectories.
- In the current stage of our research we did not implement any *spatial* measure. It delimits the range of queries and analyzes which could be answered and computed.
  Therefore, to develop and implement some spatial measures to compute values along the *spatial* dimensions and hierarchies is very important point which must be researched.
- We have limited the loading phase considering a linear interpolation, but is possible to find some topological situations (e.g roads, bridges) where is very difficult to do the interpolation because of the some constraints in the movement of the object.

Conclusions - Current/Future Work

- To develop another more complex measures in order to provide values to discover patterns or trend of the trajectories.
- In the current stage of our research we did not implement any *spatial* measure. It delimits the range of queries and analyzes which could be answered and computed.
  Therefore, to develop and implement some spatial measures to compute values along the *spatial* dimensions and hierarchies is very important point which must be researched.
- We have limited the loading phase considering a linear interpolation, but is possible to find some topological situations (e.g roads, bridges) where is very difficult to do the interpolation because of the some constraints in the movement of the object.

Conclusions - Current/Future Work

- To compute values to support the data mining tasks is a very interesting point of future research.
- The traditional methods of data mining must be adapted in order to work in a *TDW* environment, however the possibility of the usage of these techniques seems to be very promisor.
- In the current stage of the prototype we have implemented a *mining module*
  - The goal is to find *frequent pattern* of the values of the measures.
  - There is a *discretization* process to transform range of measures to items in order to compute the *patterns*.
  - How to determine an adequate level of discretization seems to be a very important point of research.

Conclusions - Current/Future Work

- To compute values to support the data mining tasks is a very interesting point of future research.
- The traditional methods of data mining must be adapted in order to work in a *TDW* environment, however the possibility of the usage of these techniques seems to be very promisor.
- In the current stage of the prototype we have implemented a *mining module*
  - The goal is to find *frequent pattern* of the values of the measures.
  - There is a *discretization* process to transform range of measures to items in order to compute the *patterns*.
  - How to determine an adequate level of discretization seems to be a very important point of research.

Conclusions - Current/Future Work

- To compute values to support the data mining tasks is a very interesting point of future research.
- The traditional methods of data mining must be adapted in order to work in a *TDW* environment, however the possibility of the usage of these techniques seems to be very promisor.
- In the current stage of the prototype we have implemented a *mining module*
  - The goal is to find *frequent pattern* of the values of the measures.
  - There is a *discretization* process to transform range of measures to items in order to compute the *patterns*.
  - How to determine an adequate level of discretization seems to be a very important point of research.

Conclusions - Current/Future Work

- To compute values to support the data mining tasks is a very interesting point of future research.
- The traditional methods of data mining must be adapted in order to work in a *TDW* environment, however the possibility of the usage of these techniques seems to be very promisor.
- In the current stage of the prototype we have implemented a *mining module*
  - The goal is to find *frequent pattern* of the values of the measures.
  - There is a *discretization* process to transform range of measures to items in order to compute the *patterns*.
  - How to determine an adequate level of discretization seems to be a very important point of research.

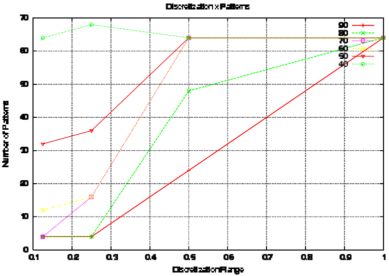
Conclusions - Current/Future Work

- To compute values to support the data mining tasks is a very interesting point of future research.
- The traditional methods of data mining must be adapted in order to work in a *TDW* environment, however the possibility of the usage of these techniques seems to be very promisor.
- In the current stage of the prototype we have implemented a *mining module*
  - The goal is to find *frequent pattern* of the values of the measures.
  - There is a *discretization* process to transform range of measures to items in order to compute the *patterns*.
  - How to determine an adequate level of discretization seems to be a very important point of research.

Conclusions - Current/Future Work

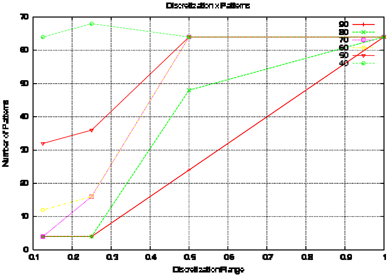
- To compute values to support the data mining tasks is a very interesting point of future research.
- The traditional methods of data mining must be adapted in order to work in a *TDW* environment, however the possibility of the usage of these techniques seems to be very promisor.
- In the current stage of the prototype we have implemented a *mining module*
  - The goal is to find *frequent pattern* of the values of the measures.
  - There is a *discretization* process to transform range of measures to items in order to compute the *patterns*.
  - How to determine an adequate level of discretization seems to be a very important point of research.

Conclusions - Current/Future Work



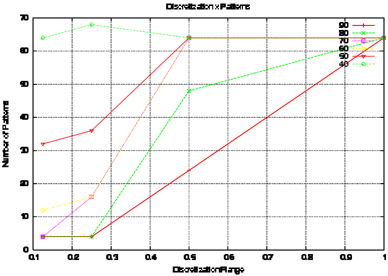
- A lower range of discretization produces a high number of items decreasing the performance of finding *patterns* or *association rules*;
- Very generic *patterns* involving a lot of records without to reveal a real knowledge.
- A high range of discretization generates a high number of *patterns* or *association rules*

Conclusions - Current/Future Work



- A lower range of discretization produces a high number of items decreasing the performance of finding *patterns* or *association rules*;
- Very generic *patterns* involving a lot of records without to reveal a real knowledge.
- A high range of discretization generates a high number of *patterns* or *association rules*

Conclusions - Current/Future Work



- A lower range of discretization produces a high number of items decreasing the performance of finding *patterns* or *association rules*;
- Very generic *patterns* involving a lot of records without to reveal a real knowledge.
- A high range of discretization generates a high number of *patterns* or *association rules*

Bibliography

F. Braz, S. Orlando, R. Orsini, A. Raffaetà, A. Roncato, and C. Silvestri.
Approximate aggregations in trajectory data warehouse.
In *Proc. of ICDE Workshop STDM*, pages 536–545, 2007.

T. Brinkhoff.
Generating network-based moving objects.
In *SSDBM '00: Proceedings of the 12th international Conference on Scientific and Statistical Database Management*, page 253, Washington,DC,USA, 2000. IEEE Computer Society.

Margaret H Dunham.
Data mining introductory and advanced topics.
Prentice Hall, 2003.

J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh.
Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals.
Data Mining and Knowledge Discovery, 1:29–54, 1997.

R. Kimball.
Data Warehouse Toolkit.
John Wiley, 1996.

T. Asai T. Uno, Y. Uchida and H. Arimura.
Lcm: An efficient algorithm for enumerating frequent closed item sets.
In *FIMI'03: Workshop on Frequent Itemset Mining Implementations*, Florida, USA, 2003. IEEE Computer Society.

THANK YOU