Noname manuscript No.
(will be inserted by the editor)

Managing sensor traffic data and forecasting unusual behaviour propagation

Claudia Bauzer Medeiros · Marc Joliveau · Geneviève Jomier · Florian De Vuyst

Abstract Sensor data on traffic events have prompted a wide range of research issues, related with the so-called ITS (Intelligent Transportation Systems). Data are delivered for both static (fixed) and mobile (embedded) sensors, generating large and complex spatio-temporal series. This scenario presents several research challenges, in spatio-temporal data management and data analysis. Management issues involve, for instance, data cleaning and data fusion to support queries at distinct spatial and temporal granularities. Analysis issues include the characterization of traffic behavior for given space and/or time windows, and detection of anomalous behavior (either due to sensor malfunction, or to traffic events).

This paper contributes to the solution of some of these issues through a new kind of framework to manage static sensor data. Our work is based on combining research on analytical methods to process sensor data, and data management strategies to query these data. The first component is geared towards supporting pattern matching. This leads to a model to study and predict unusual traffic behavior along an urban road network. The second component deals with spatio-temporal database issues, taking into account information produced by the model. This allows distinct granularities and modalities of analysis of sensor data in space and time. This work was conducted within a project that uses real data, with tests conducted on 1000 sensors, during 3 years, in a large French city.

Keywords Intelligent Transportation Systems, Traffic Modelling, Sensor Networks, Time Series

Claudia Bauzer Medeiros
IC, University of Campinas, UNICAMP
13081-970 Campinas, SP, Brazil E-mail: cmbm@ic.unicamp.br · Marc Joliveau
CIRRELT, Université de Montréal
Montréal, Quebec, H3C 3J7 Canada E-mail: marc.joliveau@cirrelt.ca · Florian De Vuyst
Laboratoire Mathématiques Appliquées aux Systèmes, Ecole Centrale Paris
Grande Voie des Vignes 92295 Chatenay-Malabry cedex, France E-mail: florian.de-vuyst@ecp.fr · Geneviève Jomier
LAMSADE, Université Paris-Dauphine place du Maréchal de Lattre de Tassigny 75 775 Paris Cedex 16, France E-mail: genevieve.jomier@dauphine.fr

1 Introduction

Geospatial data are the basis for countless applications in a wide range of domains. This paper is concerned with one such domain – transportation systems. As pointed out by [34], "one of the important innovations in transportation in recent years is the combination of advanced sensor, computer, electronics and communications technologies to the operation of the transportation system". The generic term associated with this domain is *Intelligent Transportation Systems*, or ITS.

ITS research is multidisciplinary, encompassing people from many distinct areas – from computer scientists and mathematicians to sociologists and environmentalists. The final goal is to improve the overall transportation infrastructure offered to all kinds of travelers – from those who need to cross a street to overseas flight passengers. The design and development of software embedded in intelligent vehicles, sensitive to GPS positioning, is another area in which geospatial information management is adopted.

This paper is concerned with the problems involved in the analysis of a specific kind of spatio-temporal data, obtained at real time from a large network of urban traffic sensors. These sensors continuously capture distinct kinds of data on traffic, to be used for analysis in traffic management and planning. At all times, experts must also take into account sensor failures, to avoid incorrect decisions.

Rather than manipulating the original sensor spatio-temporal data series, we first preprocess these data, eliminating noise, filling missing values, and reducing their dimensionality. The result are clean (spatio-temporal) series with nice properties. These series are used to feed our model, which supports forecasting of traffic behavior, including atypical events and congestion patterns.

Cleaned data are stored in a DBMS, on which ITS queries and pattern analyses can be performed, following a mix of standard (time series) database queries and new kinds of queries that invoke the analysis functions of our model. While related work either concentrates on models or in database issues, ours combines both approaches. We do moreover take into account the influence of human activity in urban areas (e.g., street markets, or accidents) to derive better evaluation of traffic conditions.

Section 2 presents an overview of the context of our work. Section 3 presents our solution for preprocessing and summarizing sensor produced data. Section 4 introduces a new traffic variable – congestion – which summarizes traffic behavior along spatio-temporal axes. Section 5 presents propagation graphs – a novel way of examining atypical traffic behavior, and its spatio-temporal propagation along a road network. Section 6 discusses the spatio-temporal queries that can be posed in our framework. Section 7 comments on related work, showing how our work relates to research on spatio-temporal series processing, pattern matching and traffic trend analysis. Finally, section 8 concludes the paper.

2 Overview of the Problem

Our research was conducted within the CADDY project (Control of the Acquisition and storage of massive temporal Data volumes and DYnamic models) [2]. CADDY involved a multidisciplinary research team composed of computer scientists and experts in traffic management and planning. The goal was to develop a computational framework for decision support in urban traffic management.

The sensors used in our research are fixed along street networks, and continuously collect and broadcast several kinds of data on traffic movement. Each sensor is a magnetic loop

that detects the presence of large metalic objects (e.g., cars). Values measured indicated the proportion of metal detected. Data are sent to stations, and forwarded to a central data storage facility. Two major traffic variables are used by experts in this context, producing two distinct (but interdependent) spatio-temporal series – see section 3 for their relationship:

- the vehicle flow rate (q), i.e., the number of vehicles that have passed in front of the sensor for a given time period, usually minute or hour;
- the occupancy rate (τ), i.e., the average space between vehicles in a given time period. Thus, an occupancy rate of 100% means vehicles are bump to bump, while 0% means no vehicles have been observed.

Our source data cover 1000 sensors for 3 years, where each sensor collects data every three minutes. Each day is delivered in a separate set of files (for q and τ), for a grand total of approximately 480x10^6 values[1]. Our data were provided by the CLAIRE traffic supervision system [30] from INRETS (the French National Transportation Research Institute). CLAIRE models an urban street network through an oriented graph, where each edge corresponds to a street segment. Sensor data are provided on each edge, associated with the corresponding spatial location.

Figure 1 shows the flow rate and average occupancy in a weekday, over a 24 hour period, for one sensor that is installed in a specific street segment. If this is a "typical" day, one expects that this sensor will produce similar series throughout the year – i.e., for the sensor portrayed, low traffic in the early hours, with a peak between 8 and 10 AM, another peak at noon and so on, petering out in the evening. The occupancy graph, on the right, shows that the 8-10 AM period is the one where there is a higher density of cars in the corresponding street segment being monitored. Such patterns are used by experts to detect anomalous traffic behavior. There are, however, several kinds of temporal behavior for a given area, depending on the context and human activity associated – e.g., weekends or festivals will provide different patterns.

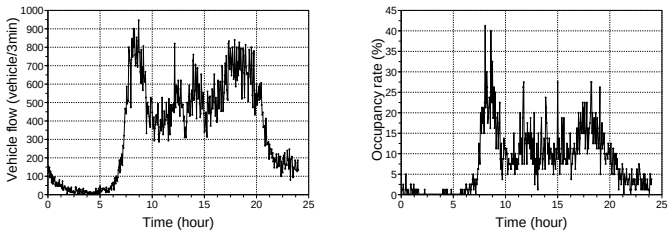

Fig. 1: Flow and occupancy rates, for one sensor, in a given street segment. The horizontal axis corresponds to time, while the y-axis portrays the values measured by the sensor.

Processing these data presents several challenges, since q and τ must be combined to allow experts an overall view, complicating the mathematical analysis. Given the potentially very large number of series and sensors, this gives rise to the so-called *dimensionality problem*, in which experts must handle multidimensional data (here, the set of time series), whose dimensions must be reduced in order to allow them to perform their analyses.

[1] I.e., (3 x 365) x 1000 sensors x (480 measures per day)

At INRETS, data are kept separate for each sensor and each day. This means that there are at least two kinds of long series that can be constructed – (a) per sensor, over time, and (b) per timestamp, for all sensors. Each kind – (a) or (b) – supports a distinct analysis. The first allows studying the behavior, for one sensor, through time (fixed point in space, varying time), while the second presents a snapshot of the entire network, at a given timestamp (fixed time, varying space). Joint analyses need to correlate these factors, resulting in a complex multidimensional data space, as will be seen in section 3.

The relationship between q and τ is defined in the fundamental car-traffic law of transportation theory. This correlation can be graphically represented by the fundamental diagram – see figure 2, which shows the inherent relationship between τ (vertical) and q (horizontal) rates, for one sensor, over one day.

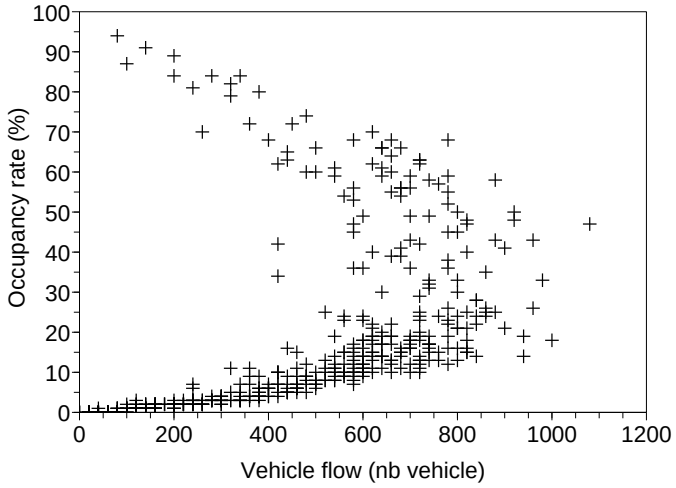

Fig. 2: Illustration of the fundamental car-traffic law of transportation theory – Fundamental diagram for one sensor, on a given day.

Another issue is quality – the data transmitted by the sensors are very noisy, and contain many gaps, mainly due to sensor failures and/or network breakdowns. Thus, the data must be cleaned before they can be processed. The next section presents our solution for the dimensionality and missing value problems. It shows how these two series can be cleaned, and represented with very small errors by a compact descriptor (thus reducing dimensionality).

3 Summarizing Sensor-based Data

This section presents our solution to pre-process the spatio-temporal sensor data series, summarizing them and reducing their dimensionality. It starts by presenting a method that dramatically reduces the dimensionality of each series (q and τ), for all sensors, while managing to maintain almost all the original information. Next, it shows how to derive missing values. Finally, it combines both kinds of series (flow rate and occupancy), to describe the overall

state of each sensor per day, mapped to the interval [0,1]. This interval can be divided into several classes, which represent specific traffic states defined by experts (e.g., “heavy”, “bottleneck”), thus supporting subsequent queries not only on values but also on semantically meaningful states.

3.1 Reducing Dimensionality – the STPCA Method

In [15], we introduced the Space-Time Component Analysis (STPCA), a new method to develop descriptors of spatio-temporal data series. This method is based on applying Principal Component Analysis (PCA) [19] to both spatial and temporal dimensions, as follows.

Assume that there are sensor data available for N days, for a total of S sensors, and that there are no missing or invalid values[2]. Let I be the number of instants in a day for which sensors collect data (in our case, I=480). Data collected on the flow rate or occupancy rate are stored in a matrix $\mathbf{X}^d$, where d symbolizes a date. The time series corresponding to measurements collected by sensor i at day d is given by row $\mathbf{x}_i^d$. The following steps are applied separately to flow rate and to occupancy rate, obtaining two sets of time series processed by STPCA.

1. Assemble day matrices horizontally (i.e., concatenate them one beside the other), for spatial analysis, in a single matrix $\mathbf{Y}$, and vertically (i.e., concatenate them on top of each other) for temporal analysis in a matrix $\mathbf{Z}$. In matrix $\mathbf{Y}$, each column contains all values obtained in a timestamp, and each row corresponds to one sensor, with values varying through time, through all days (fixed location). In matrix $\mathbf{Z}$, columns are the instants of one day and each row corresponds to data from one sensor, for one single date (fixed time); there are as many rows for each sensor as there are capture dates. Matrix $\mathbf{Y}$ is S by (IxN), while $\mathbf{Z}$ is (SxI) by N.
2. Compute singular value decomposition for matrices $\mathbf{Y}$ and $\mathbf{Z}$, as follows
   For spatial correlation matrix $\mathbf{M}^s = \mathbf{Y}\mathbf{Y}^T$, compute the K first spatial eigenvectors $(\boldsymbol{\Psi}^k)_{k=1\ldots S}$, with $K \ll S$, storing them in matrix $\mathbf{P}$. For temporal correlation matrix $\mathbf{M}^t = \mathbf{Z}^T\mathbf{Z}$, compute the L first temporal eigenvectors, with $L \ll I$, $(\boldsymbol{\Phi}^l)_{l=1\ldots I}$, storing them in matrix $\mathbf{Q}$.

$$\mathbf{P} = col\left(\boldsymbol{\Psi}^1, \boldsymbol{\Psi}^2, \ldots, \boldsymbol{\Psi}^K\right).$$

$$\mathbf{Q} = col\left(\boldsymbol{\Phi}^1, \boldsymbol{\Phi}^2, \ldots, \boldsymbol{\Phi}^L\right).$$

3. Finally, the STPCA estimate $\hat{\mathbf{X}}^d$ of a day matrix $\mathbf{X}^d$ is defined by:

$$\hat{\mathbf{X}}^d = \mathbf{P}\mathbf{P}^T\,\mathbf{X}^d\,\mathbf{Q}\mathbf{Q}^T.$$

   We point out that the reduced order matrix is given by:

$$\mathbf{X}_d^r = \mathbf{P}^T\,\mathbf{X}^d\,\mathbf{Q},$$

of size $K \times L$ where K and L are chosen to be small. Experiments done with very small values of these parameters, namely, $K = L = 3$, corresponding to a reduction factor of order 10^4, demonstrate the ability of STPCA to compute a good approximation. – see [15] for details.

[2] As will be seen, STPCA is preceded by an error-correction procedure, described later in the paper

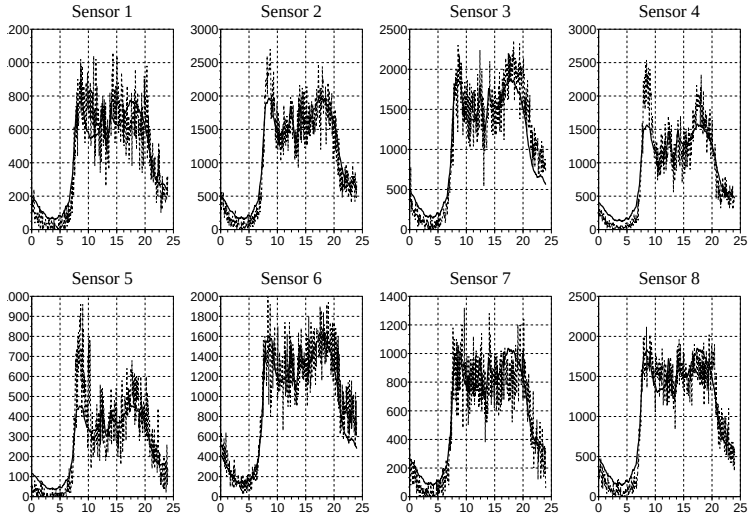

Fig. 3: Illustration of the flow rate time series, for one day, of 8 randomly chosen sensors - horizontal axis represents time. Original and STPCA time series using $K = 3$ and $L = 3$. The STPCA approximation is portrayed in bold, while actual raw data is in gray.

Intuitively, STPCA summarizes the "average" typical traffic behavior for a sensor, over all days in a period. Figure 3 illustrates flow rate time series measured by 8 sensors randomly chosen (grey curves), and their STPCA estimate (black curves), for $K = L = 3$. Domain experts considered such approximations adequate for traffic analysis and trend forecasting, in spite of divergences at specific points – e.g., the case of sensor 5, between 8 and 10 AM, for this specific day. These differences are due to the fact that, for these cases, STPCA uses approximations based on an entire day. We might also have just considered shorter periods, for every day, in which case the results would be more precise. As will be seen, these variations can be analyzed and accounted for.

3.2 Filling missing values

STPCA cannot be directly applied to a data set containing missing values. Our solution to this problem [18] is the following. We use the Expectation Maximization (EM) [5] algorithm to estimate separately the spatial correlation matrix $\mathbf{M}^s$ and the temporal correlation matrix $\mathbf{M}^t$. We compute a complete estimation of the data set by using the k nearest time series of each time series. We then project this estimation on the principal component of $\mathbf{M}^s$ and $\mathbf{M}^t$ approximations. Our experiments [18] show that results obtained by STPCA on data sets – even those with an incompleteness degree that can go as high as 40% – stay very close to those obtained by STPCA on the corresponding complete data sets.

4 Attaching Semantics to Traffic Variables

4.1 Computing Traffic Congestion

We now introduce a new variable to support traffic analysis, the *traffic congestion* (E), derived from the fundamental diagram, and computed for each timestamp. To better illustrate how it is computed, we reproduce the fundamental diagram shown in figure 2 in a schematic way – the black curve on figure 4, which shows the relation between τ (vertical) and q (horizontal) rates, for one sensor, over one day. This new variable is based on computing two kinds of value for each sensor i, per day:

- the average maximum flow rate ($\tilde{q}_i$), given by the mean of the maximum flow rate measured at sensor i for each day. It corresponds to a near optimal flow rate value with respect to traffic in front of sensor i ;
- variable $\tilde{\tau}_i$, which computes the daily average occupancy rate value when traffic reaches its maximum flow rate in a day.

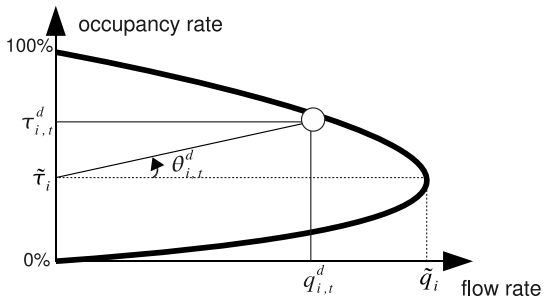

Fig. 4: Fundamental diagram with traffic congestion variable E, for sensor i

At a fixed time t for day d, the vehicle flow rate and the occupancy rate of a sensor i are respectively given by $q^d_{i,t}$ and $\tau^d_{i,t}$. For each measurement, we compute $\theta^d_{i,t}$ – the angle between the line given by $y = \tilde{\tau}_i$ and the line linking the points $(0, \tilde{\tau}_i)$ and $(q^d_{i,t}, \tau^d_{i,t})$. From $\theta^d_{i,t}$, we define $e^d_i(t)$ – the value of the traffic congestion E at sensor i at day d and time t:

$$e^d_i(t) = 1/2 + 1/\pi \times arctg\Big(\frac{(\tau^d_{i,t} - \tilde{\tau}_i)}{100} \times \frac{\tilde{q}_i}{q^d_{i,t}}\Big).$$

This formula is based on first computing angle $\theta^d_{i,t}$ using trigonometric properties – i.e.,

$$\theta^d_{i,t} = arctg\Big(\frac{(\tau^d_{i,t} - \tilde{\tau}_i)}{100} \times \frac{\tilde{q}_i}{q^d_{i,t}}\Big)$$

where there is a division by 100 because it is the maximum value of τ. The resulting angle (which is between $-\pi/2$ and $\pi/2$) is transformed into a value between 0 and 1.

4.2 Applying STPCA to congestion

We now proceed to apply STPCA to our congestion function. We can mathematically show that STPCA is able to produce accurate approximations, taking advantage of the otion of *energy* of a vectorial space. Total energy is equal to the trace of the correlation matrices $\mathbf{M}$, and is given by:

$$tr(\mathbf{M}) = \sum_{i=1}^{N} \lambda_i(\mathbf{M}),$$

where $\lambda_i(\mathbf{M})$ represents the i^{th} eigenvalue of matrix $\mathbf{M}$, and N represents the number of eigenmodes. As correlation matrices $\mathbf{M}^s$ (spatial correlation matrix) and $\mathbf{M}^t$ (temporal correlation matrix) are determined while applying STPCA, energy captured by the first m spatial and temporal eigenmodes are respectively given by:

$$\frac{\sum_{i=1}^{m} \lambda_i(\mathbf{M}^s)}{tr(\mathbf{M}^s)} \quad \text{and} \quad \frac{\sum_{i=1}^{m} \lambda_i(\mathbf{M}^t)}{tr(\mathbf{M}^t)}$$

Intuitively, the original data set corresponds to a vectorial space containing 100% energy. Hence, the more a set of eigenmodes captures energy, the more accurate the estimation obtained when projecting data on the vectorial space defined by these eigenmodes (and thus the better the dimensionality reduction).

Our experiments, discussed in [15], show that on both (spatial and temporal) dimensions, the first eigenmode contain more than 98% of the energy. Moreover, the $K = 4$ first spatial eigenmodes and $L = 6$ first temporal eigenmodes capture more than 99.5% of the energy. This allows us to reproduce very accurately the typical traffic behaviour of each sensor with a reduced dimensional space.

4.3 Interpreting congestion states – symbolic representation

We recall that the values of $e_i^d(t)$ are normalized between 0 and 1. If values are close to 0, traffic is very fluid (low flow and occupancy rates), while values close to 1 represent a large bottleneck. Values close to 0.5 correspond to nearly optimal traffic, with high flow rate. Traffic congestion E thus combines two different variables into one without loss of information. Moreover, it gives a normalized and intelligible view of traffic.

In order to work with traffic congestion, this function was discretized with help of experts into seven intervals, each of which corresponding to a different traffic state. These intervals were assigned symbols, thereby introducing a symbolic description of traffic congestion curves (for symbolic representation, see [14], and discussion on related work at section 7):

- "C": Sparse - few vehicles, typically night traffic;
- "TH": Tendency to heavy traffic – intermediate state between C and H, tending to increase in traffic;
- "H": Heavy traffic – state corresponding to quasi-optimal traffic congestion, with high flow but no bottleneck;
- "RC": Return to sparse – intermediate state between C and TH, where traffic moves from heavy towards sparse;
- "$S1$": Saturation level 1 – corresponds to a light density, with slow flow and increase in occupancy level;
- "$S2$": Saturation level 2 – severe traffic bottleneck, very slow flow and heavy traffic;
- "$S3$": Saturation level 3 – bottleneck, vehicles are almost static;

These traffic states are computed from thresholds applied to E, as well as to the sign of the derivative of the congestion function. Table 1 shows how to compute these symbolic variables at every timestamp t. Each symbol is associated with a value between 1 and 7, to facilitate computation of pattern similarity. Symbolic representation is thus mapped to a step-wise function. Figure 5 represents temporal series of traffic congestions and the corresponding symbolic representation.

Table 1: Discretization of $e_i^d(t)$ into seven intervals.

Symbolic state	Associated value	Range of $e_i^d(t)$	Derivative for $e_i^d(t)$
C	1	$e_i^n(t) < 0.2$	/
RC	2	$0.2 \leq e_i^n(t) < 0.45$	negative
TH	3	$0.2 \leq e_i^n(t) < 0.45$	positive
H	4	$0.45 \leq e_i^n(t) < 0.52$	/
S1	5	$0.52 \leq e_i^n(t) < 0.6$	/
S2	6	$0.6 \leq e_i^n(t) < 0.7$	/
S3	7	$e_i^n(t) \geq 0.7$	/

Besides helping reducing the problem dimensionality and helping user visualization, the use of symbolic representation helps data discretization. This is also useful for subsequent computations across time and space, such as mutual information – see next.

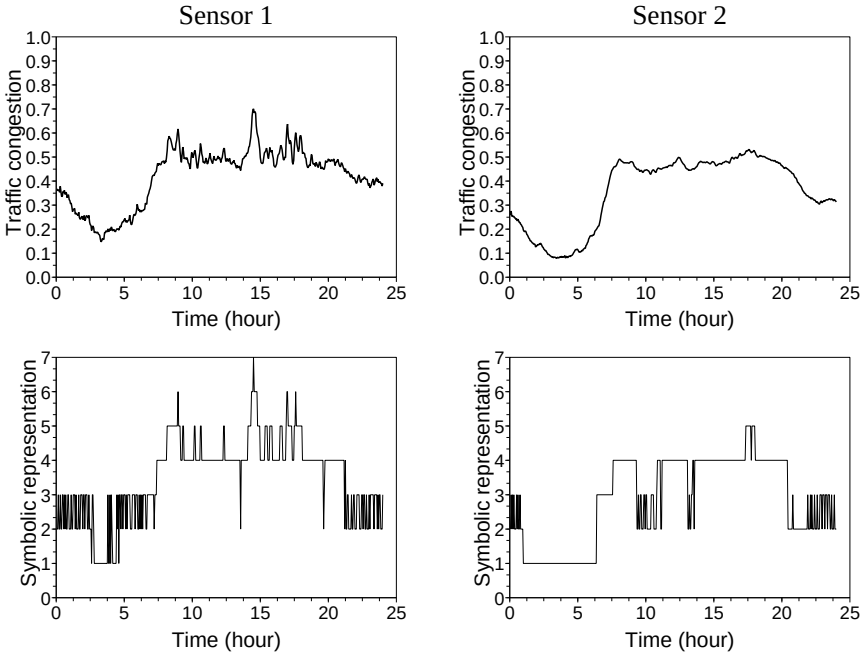

Fig. 5: Traffic congestion graphs for two distinct sensors (at the top) and corresponding symbolic representation: $C = 1$, $RC = 2$, $TH = 3$, $H = 4$, $S1 = 5$, $S2 = 6$, $S3 = 7$.

5 Handling atypical traffic behavior

Up to now, this paper has applied the STPCA method to time series to summarize and forecast typical behavior. However, it can also be a means to detect atypical situations. We say that traffic behavior is *atypical* if the distance between measured traffic congestion and its STPCA estimation is beyond a given threshold defined by the user – i.e., the activity measured by the sensor is not representative of the usual behaviour computed by STPCA.

Atypical behavior means that STPCA has either provided an underestimation of an overestimation or traffic congestion. In the first case, traffic is more intense than forecast by STPCA; in the second, it is more fluid. Once atypical situations are detected, we provide a new kind of mechanism to study and describe spatio-temporal propagation of such behavior – the *propagation graphs*[3]

5.1 Propagation graphs

A propagation graph describes how local traffic perturbations at a given instant propagate to other areas in subsequent periods. As such, these graphs portray spatio-temporal propagation of atypical traffic events, computed with respect to congestion.

Figure 6 gives a short illustration of such a graph. Vertices represent sensors, and edges represent propagation of atypical behavior. The figure shows, for instance, that some atypical event detected at sensor 1 at time t persists at the same location at time $t+1$ (i.e., the situation was propagated through time, for the same spatial reference). It also shows that the event detected at sensor 2 did not persist. Also in the figure, we can see that at time $t+1$ sensor 1 is also affected by the atypical event detected at sensor 5 at time t (i.e., the problem was propagated in time and space). Events at sensors 3 and 5 at $t+1$ affect each other at $t+2$. The rest of the figure can be interpreted the same way.

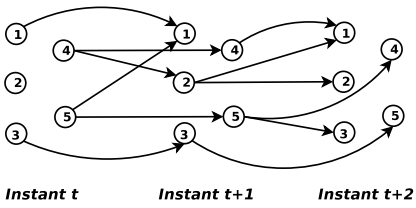

Fig. 6: Example of a propagation graph on three successive timestamps.

These graphs are constructed from appropriate subsets of the measured data, using an algorithm based on the combination of Isomap and Mutual information (see [16] for details on this algorithm). Mutual information [31] is provided by probability theory and information theory, and measures the mutual dependence of two variables. It is given by :

$$I(X,Y) = H(X) + H(Y) - H(X,Y),$$

[3] The work of [16] uses the term "pattern" to refer to propagation graphs. Here, we adopt the name "graph" to avoid confusion with pattern matching in time series.

Input: Symbolic representation congestion time series for a set of sensors

Step 1 - Compute embeddings

For each timestamp t:

1.1 Compute mutual information	*Compute mutual information $I(i,j)$ at time t between each pair of sensors i, j*
1.2 Construct neighborhood graph (*Isomap step 1*)	*Connect vertices i and j by an edge if distance $d_x(i,j) = 1 - I(i,j)$ is smaller than threshold ϵ or if j is one of the K nearest neighbors of i using d_x. Edge weights are given by $d_x(i,j)$ values.*
1.3 Compute shortest paths (*Isomap step 2*)	*Initialize $d_G(i,j) = d_x(i,j)$ if i and j are linked by an edge, $d_G(i,j) = \infty$ else. Then for parameter $k = 1 \ldots N$ replace $d_G(i,j)$ by $\min\{d_G(i,j), d_G(i,k) + d_G(k,j)\}$. Construct matrix $\mathbf{D}_G = \{d_G(i,j)\}$ containing shortest path distances for each pair of points in G.*
1.4 Construct embedding (*Isomap step 3*)	*Apply multidimensional scaling to matrix $\mathbf{D}_G$ to obtain a n-dimensional embedding $\mathbf{Y}^t$ at each timestamp t.*

EndFor

Step 2 - Create propagation graph	*Create an edge between each sensor i on embedding $\mathbf{Y}^t$ and i's K nearest neighbors restricted to an ϵ maximum distance radius on embedding $\mathbf{Y}^{t+1}$ according to Euclidean distance. Keep only edges with high propagation probability.*

Fig. 7: Algorithm to construct propagation graphs.

where $H(X)$ and $H(Y)$ respectively measure the entropy of variables X and Y, and $H(X,Y)$ corresponds to the cross entropy for these variables. Mutual information values are defined between 0 and 1. The higher its value, the stronger the relationship among events. When its value is zero, the events are independent.

Isomap [35] is an approach to solve dimensionality reduction problems that uses local metric information to derive the underlying global geometry of a data set. This method can be decomposed in three steps. The first step determines which points are neighbors, based on the distances $d_x(i,j)$ between pairs of points i and j from the data set. Neighborhood relations are then represented as edges in a weighted graph G. The second step of Isomap estimates the geodesic distances $d_G(i,j)$ between all pairs of points by computing their shortest path in G. The final step applies classical multidimensional scaling (MDS) to the geodesic distances matrix $\mathbf{D}_G$, reducing its dimensionality. This last step projects the original data into a n-dimensional Euclidean space that best preserves the data's estimated intrinsic geometry. We call this projection a *n-dimensional embedding* – an Euclidean space of dimension n, in which the Euclidean distance between its points is representative of their similarity.

Figure 7 presents a high level view of our algorithm to construct propagation graphs. It receives as input congestion information, computed from measured data, and reduced to a symbolic representation. The key in our approach is to compute mutual information on a symbolic description of the traffic congestion and then use it as distance in the first step of

Isomap. Each point in the data set corresponds to a specific sensor, and a n-dimensional embedding is computed for each timestamp. Generally, even very small values of n (e.g., $n = 2$ or $n = 3$) are enough to capture data's geometry. The distance between two points on the embedding is used to compute the similarity of traffic behavior for the corresponding sensors (the smaller the distance, the more similar traffic is).

To construct propagation graphs, we link points between consecutive embeddings. Our hypothesis is that, if traffic congestion is atypical on sensor i at time t, traffic at this same sensor will still be atypical at $t + 1$ – obviously depending on the time interval between two successive timestamps. Moreover, an atypical event at sensor i and time t will affect all of is neighbors in the corresponding n-dimensional embedding at time $t + 1$.

To increase readability, we introduce the following notation, used in the rest of the text

(i, t) – node in a propagation graph, denoting sensor i at time t;

$< (i, t), (j, t + 1) >$ – an edge in the propagation graph – notice that edges always link embeddings at consecutive timestamps, i.e., there are no edges within an embedding, and no edges linking embeddings which are separated by more than one timestamp;

$\mathcal{A}^{i,t}$ – an atypical event detected at sensor i at time t – i.e., at graph node (i, t). We recall that atypical events are detected whenever the congestion measured at (i, t) is different (over some threshold) from its STPCA average behavior estimate.

$P(\mathcal{A}^{j,t+1}|\mathcal{A}^{i,t})$ – probability that, if atypical event $\mathcal{A}^{i,t}$ is detected at (i, t), then there will be an atypical event $\mathcal{A}^{j,t+1}$ at $(j, t + 1)$.

Step 2 starts by creating edges $< (i, t), (j, t + 1) >$ – i.e., linking each sensor i in an embedding at time t with the same sensor i and all its neighbors at timestamp $t + 1$. Graph edges are next assigned weights $P(\mathcal{A}^{j,t+1}|\mathcal{A}^{i,t})$. These probabilities are computable from available information. Finally, graph edges with weights below a certain threshold are eliminated.

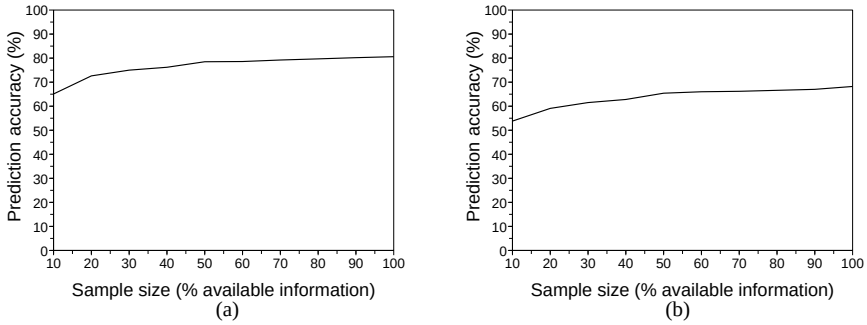

Fig. 8: Prediction accuracy depending on the data sample size for (a) predictions within 3 minutes (consecutive timestamps) and (b) predictions for a 15 minute interval.

Numeric experiments showed the accuracy of propagation graphs. If all data are used, they manage to foretell approximately 80% of atypical event propagations in a 3 minute period, and 68% of atypical event propagation in a 15 minute period – see figure 8. Moreover, as shown in the figure, graphs constructed using only 30% of the available data predict up to 75% and 61% of the propagation respectively for 3 and 15 minute periods. This confirms

the hypothesis that atypical situations at (i, t) propagate in spatio-temporal patterns in the neighborhood of sensor i.

5.2 Graphical representation of propagation graphs

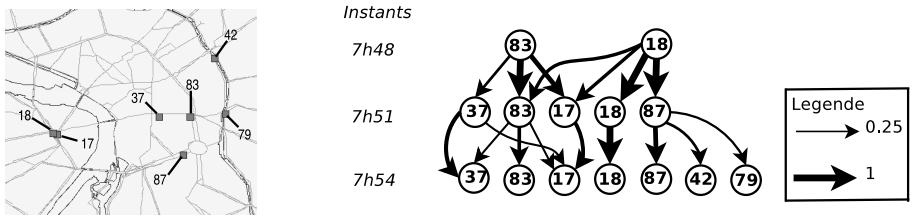

Fig. 9: Illustration of spatio temporal propagation pattern extract.

In order to help traffic experts to quickly grasp atypical propagation patterns, we designed a graphical representation of propagation graphs – see the right side of Figure 9. The left part of the figure represents the traffic network's topology. Sensors are illustrated by squares, roads by gray lines, and rivers by thin black lines. The right part of the figure is the graphical representation itself, corresponding to the period between $7h48$ and $7h58$. Edge directions indicate propagation sense, and edge thickness represents propagation probability $P(\mathcal{A}^{j,t+1}|\mathcal{A}^{i,t})$. The highest this value, the thickest the corresponding arrow.

The graphical notation helps visual identification of situations which seem logically acceptable – for instance, an atypical event at sensor 83 (at $7h48$) lasting for 6 minutes at its site, though decreasing in intensity after the first 3 minutes. We can also see in the graph that traffic on sensor 83 affects traffic on sensor 17 after 3 minutes – again, a seemingly reasonable situation if one considers the network topology, since these sensors lie along the same route. However, the graph also shows surprising cause-effect abnormalities – e.g., from sensor 18 to sensor 87, though they are not directly connected in the network. Looking at such a graphical representation, experts are able to derive spatio-temporal traffic propagation behavior patterns that can help them set up traffic control strategies. We now proceed to show how experts can use our model to detect anomalous events and study their propagation.

5.3 Distinguishing atypical events - spatio-temporal propagation of overestimation and underestimation events

An atypical situation is detected when the difference between traffic congestion measured at a sensor and the STPCA estimate for the congestion (based on average historical behavior) is above a certain threshold, which is defined by experts after preliminary simulations. Two kinds of atypical events can be distinguished:

– traffic underestimation (STPCA underestimates congestion): traffic is more intense than usual. It corresponds to atypical situations for which measured congestion is greater than its STPCA estimate. $\mathcal{U}^{i,t}$ denotes the detection of a traffic underestimation event at sensor i and time t.

– traffic overestimation (STPCA overestimates congestion): traffic flows more smoothly than usual. It corresponds to atypical situations in which measured congestion is smaller than its STPCA estimate. $\mathcal{O}^{i,t}$ denotes the detection of a traffic overestimation event at sensor i and time t.

Hence,

$$\begin{cases} \{\mathcal{O}^{i,t}\} \cup \{\mathcal{U}^{i,t}\} = \{\mathcal{A}^{i,t}\} \\ \{\mathcal{O}^{i,t}\} \cap \{\mathcal{U}^{i,t}\} = \varnothing. \end{cases}$$

We now proceed to identify two kinds of propagation situations:

– underestimation graph ($\mathcal{G}_{\mathcal{U}}$): For each underestimation event detected at (i,t), it shows the events propagation. Edges are weighted by probabilities $P(\mathcal{A}^{j,t+1}|\mathcal{O}^{i,t})$, that we refer to as traffic *underestimation propagation tendency*.
– overestimation graph ($\mathcal{G}_{\mathcal{O}}$): For each overestimation event detected at (i,t), it shows the events propagation. Edges are weighted by probabilities $P(\mathcal{A}^{j,t+1}|\mathcal{U}^{i,t})$, that we refer to as traffic *overestimation propagation tendency*.

$\mathcal{G}_{\mathcal{U}}$ and $\mathcal{G}_{\mathcal{O}}$ are computed independently using the algorithm of Figure 7, applying either underestimation propagation tendency $P(\mathcal{A}^{j,t+1}|\mathcal{U}^{i,t})$ (for $\mathcal{G}_{\mathcal{U}}$), or overestimation propagation tendency $P(\mathcal{A}^{j,t+1}|\mathcal{O}^{i,t})$ to construct $\mathcal{G}_{\mathcal{O}}$.

Remark that the nature of the propagation is unknown – i.e., an underestimation event observed at node (i,t) can trigger over- or underestimation events at node $(j,t+1)$ – the graph will only show some kind of propagation has a high probability. We can only tell what kind of atypical situation was observed at the beginning (the nodes at the graph source). We call these events *causal atypical events*, to denote they are at the origin of propagation of some kind of atypical behavior.

To compare underestimation and overestimation propagations, we introduce the notion of *differentiation rate* $\tau_D(\mathcal{G}_{\mathcal{U}}, \mathcal{G}_{\mathcal{O}})$. It computes the proportion of edges that appear in either the overestimation or the underestimation graph, but not on both:

$$\tau_D(\mathcal{G}_{\mathcal{U}}, \mathcal{G}_{\mathcal{O}}) = \frac{1}{|\mathbf{T}|} \sum_{t \in \mathbf{T}} \frac{1}{|\mathbf{I}_t|} \sum_{i \in \mathbf{I}_t} \frac{ExclusiveNeigh_{i,t}}{TotalNeigh_{i,t}}$$

where

$\mathbf{T}$ is the set of time periods considered to construct the graph;
I_t contains all graph nodes (i,t) which are connected to at least one node $(j,t+1)$ in either $\mathcal{G}_{\mathcal{U}}$ or $\mathcal{G}_{\mathcal{O}}$;
$ExclusiveNeigh_{i,t}$ is the total number of edges $< (i,t),(j,t+1) >$ which appear in either $\mathcal{G}_{\mathcal{U}}$ or $\mathcal{G}_{\mathcal{O}}$, bot not in both ;
$TotalNeigh_{i,t}$ is the number of distinct edges $< (i,t),(j,t+1) >$ that appear in $\mathcal{G}_{\mathcal{U}}$ or $\mathcal{G}_{\mathcal{O}}$.

The goal of the differentiation rate is to check whether a causal atypical event triggers different propagations, depending on whether it is an under- or overestimation; High values of this rate indicate that a causal underestimation event does not propagate to the same sensors as a causal overestimation event – i.e., the graphs are different depending on the nature of the causal event.

In fact, if we separately consider the two kinds of causal atypical events, the graphs become slightly more accurate (i.e., 81% for a 3 minute interval, and 69% for a 15 minute interval, with slightly less false positives). If all available data are used, the graphs show that the differentiation rate reached 75% for 3 minute predictions, and 86% for a 15 minute

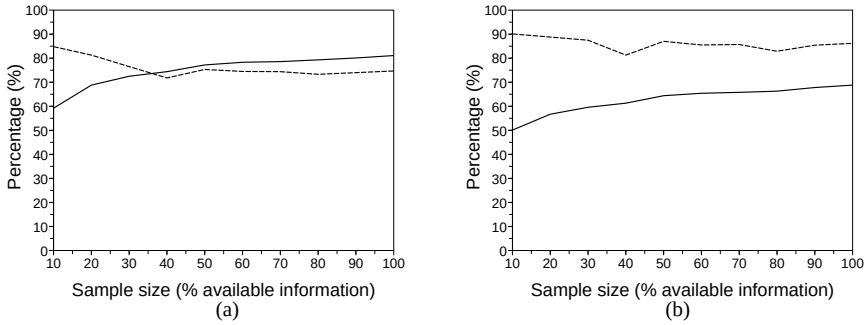

Fig. 10: Prediction accuracy (solid line) and differentiation rate (dashed line) evolution depending on the amount of data used to create the graph (a) for a 3 minute interval prediction and (b) for 15 minute interval predictions, when causal atypical situations are differentiated.

prediction. Such values indicate that, on the average, 75% of the sensors affected after 3 minutes by an underestimation event are not the same as those affected by an overestimation event. If graphs are constructed using only 30% of the data, then they are able to predict 72% of propagations for 3 minute intervals, and 60% at 15 minute intervals, with differentiation rates of respectively 76% and 87% (figure 10).

Hence, the distinction between causal over- and underestimation events allows more precise analysis and forecast of traffic anomalies. The level of quality of the prediction is the same, while at the same time taking into account the nature of the causal situation and its propagation throughout the traffic network.

5.4 Cause-consequence relations

Intuitively, an atypical event at (i, t) can propagate throughout the network and provoke atypical events (of either kind, under- or overestimation) at subsequent timestamps. Let us illustrate this idea. Consider 3 sensors i, j and k that are placed in this spatial order along some traffic axis, distant from each other by a few kilometers. Let us suppose that an accident occurs just in front of sensor j, thereby creating traffic problems - i.e., congestion decreases (an underestimation event, in which actual traffic conditions will be worse than the average forecast by $e_i^d(t)$). This may block traffic at j, and thus the traffic flowing from its position will diminish in the subsequent periods. This will cause an overestimation event at k (there will be less vehicles than expected). Moreover, if the bottleneck at j takes longer, the atypical situation may extend back to sensor i that precedes j spatially, also halting traffic at i. This is an example that shows that an underestimation at j propagates an overestimation to k, and as underestimation to i.

Sensor spatial configuration thus allows forecasting the kinds of propagation relations that can be established among sensors. Underestimation and overestimation graphs (resp. $\mathcal{G}_{\mathcal{U}}$ and $\mathcal{G}_{\mathcal{O}}$) are constructed according to probability computations (resp. $P(\mathcal{A}^{j,t+1}|\mathcal{U}^{i,t})$ and $P(\mathcal{A}^{j,t+1}|\mathcal{O}^{i,t})$). As mentioned before, these graphs do not consider the nature of the propagation – they just show propagation of a causal atypical event. Experts may want to know more about the type of propagation – i.e., specific cause-effect edges. This does not require building new graphs – rather, we will perform a distinct graph analysis.

In more detail, we now compute probability values $P(\mathcal{U}^{j,t+1}|\mathcal{U}^{i,t})$ and $P(\mathcal{O}^{j,t+1}|\mathcal{U}^{i,t})$ for edges in $\mathcal{G}_{\mathcal{U}}$; and probability values $P(\mathcal{U}^{j,t+1}|\mathcal{O}^{i,t})$ and $P(\mathcal{O}^{j,t+1}|\mathcal{O}^{i,t})$ for the edges in $\mathcal{G}_{\mathcal{O}}$. For instance, $P(\mathcal{O}^{j,t+1}|\mathcal{U}^{i,t})$ is the probability that an underestimation event at (i,t) is propagated as an overestimation event to $(j,t+1)$.

Propagation of $\mathcal{A}^{i,t}$ to $\mathcal{A}^{j,t+1}$ may have the following outcomes:

- $\mathcal{A}^{j,t+1}$ is an overestimation;
- $\mathcal{A}^{j,t+1}$ is an underestimation;
- there is no propagation (atypical event is restricted to (i,t)).

Our cause-effect study will only consider the first two outcomes, since we are interested in finding out whether propagation types are predictable or not. In this simplification, we ignore the fact that atypical events may not propagate to all their neighbors in a n-embedding. Let us simulate a context in which propagation prediction is 100% correct, i.e.,

$$P(\mathcal{A}^{j,t+1}|\mathcal{A}^{i,t}) = 1;$$

which is equivalent to:

$$\forall < (i,t),(j,t+1) > \in \mathcal{G}_{\mathcal{O}},\ P(\mathcal{A}^{j,t+1}|\mathcal{O}^{i,t}) = P(\mathcal{O}^{j,t+1}|\mathcal{O}^{i,t}) + P(\mathcal{U}^{j,t+1}|\mathcal{O}^{i,t}) = 1,$$
$$\forall < (i,t),(j,t+1) > \in \mathcal{G}_{\mathcal{U}},\ P(\mathcal{A}^{j,t+1}|\mathcal{U}^{i,t}) = P(\mathcal{O}^{j,t+1}|\mathcal{U}^{i,t}) + P(\mathcal{U}^{j,t+1}|\mathcal{U}^{i,t}) = 1.$$

Under these conditions, and for the time intervals considered (3 and 15 minutes) we observed the following from our experiments:

- For $\mathcal{G}_{\mathcal{U}}$, more than 95% of edges $< (i,t),(j,t+1) >$ obey either $P(\mathcal{U}^{j,t+1}|\mathcal{U}^{i,t}) \geq 0.9$ or $P(\mathcal{O}^{j,t+1}|\mathcal{U}^{i,t}) \geq 0.9$;
- For $\mathcal{G}_{\mathcal{O}}$, more than 95% of edges $< (i,t),(j,t+1) >$ obey either $P(\mathcal{U}^{j,t+1}|\mathcal{O}^{i,t}) \geq 0.9$ or $P(\mathcal{O}^{j,t+1}|\mathcal{O}^{i,t}) \geq 0.9$;

In other words, for more than 95% of the time, we can predict the type of atypical event propagated to $(j,t+1)$ with an error margin inferior to 10% (since all probabilities are superior to 0.9). This leads to the conclusion that propagation is not random, and that we can even develop very reasonable propagation forecasts.

Nevertheless, propagation may not always happen. Thus, to finish this study, we now consider the third case – when there is no propagation. Propagation probability can also be used to compute estimation errors. For instance, if $P(\mathcal{U}^{j,t+1}|\mathcal{O}^{i,t}) = 0.88$ (probability that overestimation at (i,t) propagates as underestimation at $(j,t+1)$), then this means that the underestimation event at $(t+1)$ is predicted with an error margin of 12%.

Figure 11 shows curves that indicate prediction accuracy. For instance, figure 11(a) shows that, for a 3 minute period, if is possible to predict over 30% of cause-effect propagations with an error margin below 10%. If we accept an error margin of 20% or less, more than 50% of the situations can be predicted. For 15 minute periods, results show that our predictions are slightly less reliable (see part (b) of the figure) – for an error margin of 20%, we are able to correctly predict more than 33% of cause-effect propagations. Given our results for perfect prediction, we can safely say that these errors are mainly due to the difficulty of forecasting the presence of any kind of propagation, than of determining the nature of the propagation itself.

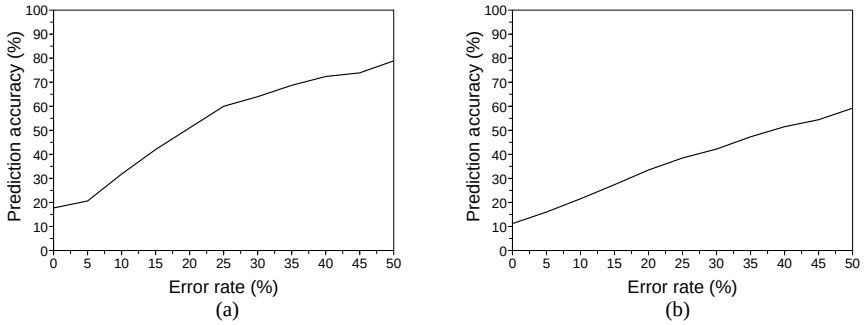

Fig. 11: Evolution of prediction accuracy depending on the accepted error rate for predictions on (a) a 3 minute and (b) a 15 minute period.

6 Querying and Mining Traffic Data

In [29] we described the structure of a data warehouse to store the raw sensor data. The design of this warehouse took into consideration the multiple spatial and temporal aspects that must be considered when dealing with traffic data, hence allowing aggregation along several dimensions.

We now extend this proposal by supporting the storage of all kinds of data discussed previously together with the raw data – i.e., the series with missing values filled, traffic congestion (E) as well as STPCA summarizations (for q, τ and E), as well as their symbolic discretization. This multitude of data representations allows several kinds of analyses not reported elsewhere. This section indicates the classes of queries that can be posed. Though not implemented, they exemplify new kinds of analyses we can obtain.

6.1 Pattern Analysis

Pattern analysis is deeply related with similarity search. The goal of similarity search is to find, in a database, all series which are similar to another series provided as input.

Let I be an input series, which can represent τ, q or E series. $Q1$ shows standard pattern analyses, whose implementation can be solved by similarity processing algorithms typical of data mining processes – see section 7. Thus, such queries can be processed on the cleaned data set using these standard procedures.

– Q1 - Retrieve all series which are similar to I , but only for sensor i and/or day d

$Q2$ through $Q5$, however, are new kinds of query. They can only be answered due to our spatio-temporal representations, e.g., a query on “typical behavior” requires STPCA.

– Q2 - What is the traffic congestion behavior of sensor i for day d
  This query returns the symbolic representation of E
– Q3 - What is the typical flow rate q behavior of sensor i
  This query returns the STPCA estimate for q, capturing its behavior over time
– Q4 - Show the sensors whose typical traffic congestion behavior is closest to that of sensor i on day d

This query is processed in three steps. First, the congestion function is computed; next, its STPCA approximation is calculated. Finally, the sensors returned are those whose STPCA approximation (of E) is closest to measured values. The notion of "close" is a threshold defined by the user.

– Q5 Show all sensors with a high margin of outliers for day d
Here, the goal is to retrieve sensors where there is a high percentage (user defined) of overestimated or underestimated values.

Outliers can be detected by comparing a curve created by applying STPCA to q or τ, and the original series, for a given day and sensor. Figure 12 shows examples of such a difference – values computed for a typical day, for a sensor, and values captured on Christmas. The possibility of describing typical behavior allows checking for anomalies, and also detecting for which kinds of day an approximation must be tuned – e.g., eliminating these days from global matrices **Y** and **Z**, and computing STPCA for them alone.

We point out that our work does not differentiate between regular and special days (e.g., holidays). Hence, normal behavior in a special day would appear as atypical. This kind of differentiated analysis is left for future work.

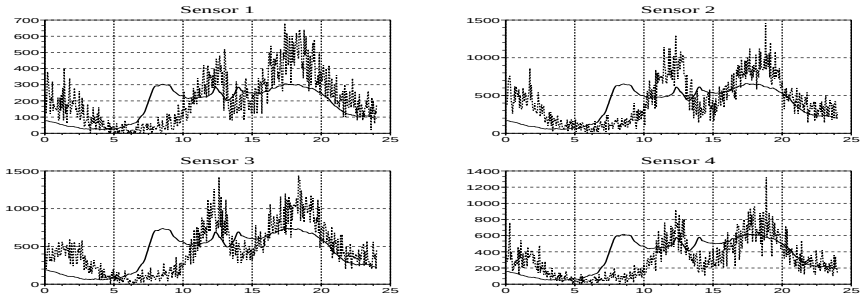

Fig. 12: Comparison of actual measured values for a sensor series (gray) and their STPCA estimate (black), for an atypical date - outliers.

6.2 Spatio-temporal Database Queries

Two kinds of spatio-temporal database queries can be considered – those on the different numeric (measured) series, and those on symbolic representations applied to E. The first can be divided into two basic situations – temporal and spatial queries; spatio-temporal queries are classically a result of the combination of both. Examples on queries on numeric series are

– Q6 - Spatial information – retrieve all series on q (or τ, or E) for sensors within a given region.
Standard spatial window query, returning data on all sensors whose coordinates fall within the input region. It returns a set of series (assuming each day is stored separately)

– Q7 - Temporal information – retrieve all series on q (or τ, or E) within a given period
Standard temporal query where the predicate will be checked against sensor timestamps. Screen copy on figure 13 is an example of the kind of output display of this query.

– Q8 – Spatio-temporal information – a combination of Q6 and Q7.

We finally come to queries on symbolic data. This allows users to retrieve information with associated semantics, closer to the user mental framework.

– Q9 – What are the sensors that on day d had more than 30% of $Heavy$ traffic congestion values
– Q10 – What are the sensors within a given region that in at least one day had more than 30% of $Heavy$ traffic congestion values (spatial restriction to a region, and then executing $Q9$ repeatedly for each sensor in that region)

7 Comparison to Related Work

This paper concerns methods and mathematical tools to summarize and interpret time series, captured by traffic sensors, which can be subsequently submitted to distinct kinds of manipulation. Our research thus combines work on time series summarization, management and pattern mining, and work on spatio-temporal databases for handling sensor data for traffic management. This paper extends our work in [17] by introducing propagation graphs, and discussing the kinds of propagation patterns that may be analyzed. This section briefly comments on related work in time series, and research on traffic variables as a whole.

7.1 Intelligent Transportation Systems

Spatio-temporal database research on traffic networks involves countless issues. In many cases, the network is transformed into a (spatial) graph, which is used as a basis for planning maintenance and expansion of the network. A large percentage of the graph structures analyzed in the context of traffic databases concern road topology - and are thus a spatial (but not necessarily temporal) structure. Our propagation graphs, instead, are spatio-temporal constructs. This kind of formalism is more common to work in transportation sciences - however, in such a domain, the main concern is temporal evolution within a fixed region (and thus the emphasis is on time, but not necessarily space).

Graphs are either theoretical constructs (i.e., the interest is to analyze algorithm performance for arbitrary complex topologies) or correspond - like ours - to actual road topology. However, in these graphs, nodes are street intersections, and edges are road segments, whereas our nodes are sensor positions and edges are traffic propagation links. Thus, our propagation graphs must be always analyzed in conjunction with network topology graphs.

Operations research and graph theory play important roles – e.g., [28] is concerned with routing models, using computation of flow in the network graph. The work of [24] also takes advantage of network graphs and graph properties to optimize kNN and distance range queries in traffic, while [4] perform continuous kNN queries on a network, for mobile objects.

Graphs and network structures are furthermore used to forecast and analyze trajectories of mobile objects (e.g.,[23] in indexing trajectories). There is also extensive research on models for mobile objects and trajectories – e.g., [9, 33]. These models assume that the identity of each object is known, and that the sensors are placed on the moving objects.

Several other problems give rise to database-related research on traffic issues. For instance, the work of [12] concerns detecting specific spatial relationships among moving

vehicles along a trajectory (e.g., ”in-front-of” relationship). This kind of research is concerned with small granularity details, considering individual vehicles whose data, collected periodically, are stored in databases.

Our work is based on using traffic estimates computed on average sensor-provided data, which is used to predict traffic congestion. In this sense, it can be contrasted, for instance, to [36], in which historical data on trajectories along given routes is used to forecast real-time trajectories. Though the idea is the same, the latter is concerned with individual vehicles (and their historical data); in this research, real-time data is used to update trajectory forecast, and thus improve future trajectory studies. Since our historical data is on global flow and occupancy (and not on individual car behavior), we can save considerable storage space. On the other hand, we cannot provide information on individual vehicles.

Computer networks is another area in computer science in which there is extensive investigation in combining research in network properties and sensor traffic information. In particular, in the area of *VA-NET* - Vehicular Ad Hoc Networks - all vehicles gather data on traffic conditions, and broadcast it to other vehicles, thereby providing real-time information to drivers. Problems therefore involve filtering relevant information, providing decision tools, and managing mobile telecommunications (e.g., what data are relevant to which set of vehicles – see, for instance [26]). Additional issues consider sensor placement and information aggregation (to optimize use of bandwidth). Though our work is not directly related to this kind of research, it can be applied in solving some of the problems investigated, in particular when it comes to propagation of atypical behavior.

Last but not least, in transportation sciences, there is extensive research on Intelligent Transportation Systems from the point of view of traffic experts. In such research, traffic flow can be schematized on top of roads, or use digraphs for study of variables – e.g., see [32]. This also includes work on propagation of phenomena. However, to the best of our knowledge, none of these papers have considered atypical event propagations like us, in special when it comes to considering our research on propagation graphs.

7.2 Summarization of time series

Many techniques are used to perform summarization and search operations on time series, such as machine learning (supervised and non-supervised training), linear regression, dynamic programming and signal decomposition [10, 11, 8, 37, 20]. According to Faloutsos [6], many of these operations are tightly interconnected. Hence, there have been proposals to construct a basic set of primitives to be used to perform them – e.g., the combination of pattern discovery and similarity search can be used to predict values. All functionalities require some sort of comparison, to recognize patterns, where exact matching is virtually impossible.

A recent research direction concerns *dynamic* series, in which data sets may be updated during the summarization and mining activity, in particular concerning stream data (e.g., from sensors). Our work is concerned only with static series (where data are collected and stored in a database), and thus stream data analysis and dynamic summarization will not be considered here.

When dealing with data containing time series, the main concern is usually to reduce the temporal dimension, preserving the original information within some predefined error threshold.

Signal processing methods include DFT - *Discrete Fourier Transform* [1,7] and DWT - *Discrete Wavelet Transform*, proposed by [3]. Each such method is based on representing the series by coefficients that summarize it at some granularity.

SVD - *Singular Value Decomposition* is based on considering the time series as a set of n-dimensional vectors. The goal is to project these vectors into a k-dimensional space, where $k < n$, maximizing the energy (i.e., conserving information). Each series is next represented by coefficients applied to the k different functions, thus defining the basis for the projections.

Segmentation methods approximate a series by a set of linear segments. Examples include PAA - *Piecewise Aggregate Approximation* [20,37], APCA - *Adaptive Piecewise Constant Approximation* [21] and PLA - *Piecewise Linear Approximation*. The first approximates a time series with k elements into m intervals of the same size, where k is a multiple of m. The series is next represented by a set of segments (step function), where the y-value for each segment is given by the average of the values for that time interval. The second improves PAA - the difference is that, instead of regular intervals, the number and length of intervals varies according to the series. In PLA, instead of step functions, segments connect the actual measured points. [22,13] propose several algorithms to determine segment extremities.

In symbolic representation - e.g., [25,14], the series is somehow converted into a sequence of symbols, i.e., a series of values is transformed into a string. This representation allows the use of text matching algorithms to compare series. Symbols are obtained by classification. Symbolic representation can also be based on intervals adapted to data profiles – see [13] for an overview of symbolic representation algorithms and models.

Several representations are a result of composing distinct kinds of methods. TIDES [27] is an example of an approximation that combines more than one of the previous techniques. It first reduces the original series using PLA. Next, it represents each segment by its angle with respect to the y axis, and associates symbols to classes of angles (thus combining symbolic representation and PLA).

STPCA and SVD both sumarize time series while maximizing energy. We now compare both. Comparisons are done according to two criteria: (a) estimation error, given by the average Euclidean distance between the real and the estimated time series and (b) reduction factor. It is very interesting to compare their respective accuracy, since their parameters as tuned according to the same conditions. SVD is not really a spatio-temporal method – it applies a PCA either on the spatial or the temporal dimension. SVD performed on the spatial dimension is also known as S-mode PCA, while SVD applied on temporal dimension is named T-mode PCA.

In this comparison, T(S)-mode PCA is applied with parameter $L = 3$ and STPCA is applied with parameters $K = 3$ and $L = 3$. STPCA totally outperforms the other methods while producing an estimation error of 0.29 which is the same order of those obtained by S-mode PCA (0.28) and T-mode PCA (0.25) with a 10^4 reduction factor – far greater than those obtained with S-mode PCA (160) and T-mode PCA (60).

8 Conclusions and Ongoing Work

This paper presented research conducted within a multidisciplinary project, in the domain of spatio-temporal sensor data processing for urban traffic. It combines research on analytical methods to pre-process, clean and summarize multiple sensor data sources, and research on

spatio-temporal database management. As part of the work in CADDY, we have constructed a prototype that allows the visual exploration of sensor data, shown in figure 13.

Different kinds of time series can be processed by STPCA, a method that improves data analysis considering the spatio temporal aspect of the dataset. This method allows a considerable reduction of the dimensionality of data sets, while at the same time preserving the original information, as shown by our energy experiments. Moreover, it offers a compression factor which outperforms those obtain by classical S-mode PCA and T-mode PCA with an equivalent reconstruction error. STPCA also dominates these methods when it comes to estimating missing values.

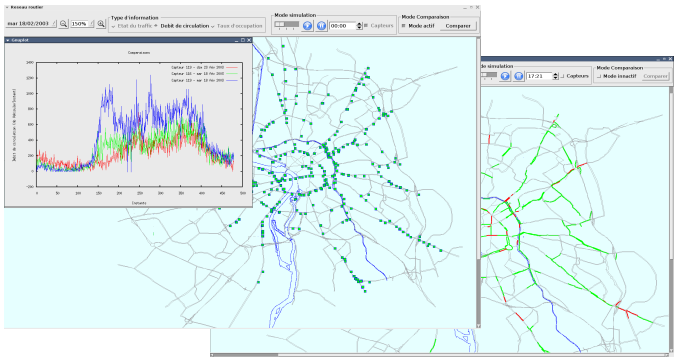

Fig. 13: Screen copy of prototype.

Another contribution is the introduction of the notion of traffic congestion, a new kind of traffic variable which combines flow and occupancy information. The paper shows how one can use symbolic representation to describe congestion, thereby helping experts analyze traffic conditions.

One major contribution is our study on propagation of atypical traffic events (characterized by deviations from STPCA estimates). The use of propagation graphs, and the analysis described on these graphs, show that they are a promising and effective tool to support decision making in traffic management, and thus intelligent transportation systems. Our studies backed up several empirical observations. However, they also show some unexpected traffic behavior. In particular, our experiments with propagation graphs indicate that the urban areas affected by an atypical traffic event differ according to the nature of the causal event – i.e., atypically slow traffic does not propagate in the same spatio-temporal patterns as atypically smooth traffic.

Finally, we describe new kinds of queries and analyses that can be performed using all these new methods and analytical tools. Though some of these queries are common in time series databases, most of them provide support to novel spatio-temporal analyses.

Future work involves theoretical and implementation issues. The latter mainly concern developing more tools within the prototype to directly support the classes of queries described. This will require, among others, linking the prototype to pattern recognition algorithms. An interesting kind of query that we are considering concerns exploration of propagation graphs. In fact, these directed graphs can be stored using some kind of linear storage structure (e.g., linked lists). Such structures can also be queried and mined, to analyze

propagation effects, and eventually trigger alerts to experts. Since these are probabilistic structures, this will require work on statistical databases.

We furthermore need to explore other extensions to STPCA. This method is geared towards studying individual sensors, for specific patterns, to determine an average typical behavior for each sensor. STPCA can be extended to spatial aggregation (several sensors) over an area. Another issue concerns deriving other kinds of traffic behavior description – e.g., for atypical days, due to variation in human activity in a given area, such as holidays or festivals. A combination of these two extensions would support a wider variety of traffic pattern descriptions – e.g., for distinct events. This, in turn, would allow new kinds of decision support in real time traffic management, including interaction with intelligent car systems.

Acknowledgements This work was partially financed by CNPq (Brazil) and by the French Research Program ”ACI Masse de Données 2003”.

References

1. R. Agrawal, C. Faloustos, and A. Swami. Efficient similarity search in sequence databases. In *Proc. 4th International Conference on Foundations of Data Organization and Algorithms*, pages 69 – 84, 1993.
2. CADDY. The CADDY Website - http://norma.mas.ecp.fr/wikimas/Caddy, 2007.
3. K. Chan and A.W. Fu. Efficient time series matching by wavelets. In *Proc. 15th IEEE International Conference on Data Engineering*, pages 126 – 133, 1999.
4. H.-J. Cho and C.-W. Chung. An Efficient and Scalable Approach to CNN Queries in a Road Network. In *Proceedings 31st VLDB conference*, pages 865–876, 2005.
5. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood for incomplete data via the em algorithm. *Journal of the Royal Statistical Society series B*, 39:1–38, 1977.
6. C. Faloutsos. Tutorial: Sensor Data Mining: Similarity search and pattern analysis. In *28th International Conference on Very Large Data Bases*, Hong Kong, China, August 2002.
7. C. Faloutsos, H. Jagadish, A. Mendelzon, and T. Milo. A signature technique for similarity based queries. In *Proc. of the International Conference on Compression and Complexity of Sequences*, pages 2–20, 1993.
8. C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings 1994 ACM SIGMOD Conference, Mineapolis, MN*, pages 419–429, 1994.
9. R. Guting, M. Bohlen, E. Erwig, C. Jensen, N. Lorentzos, M. Schneider, and M. Vazirgianis. A Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems*, 25(2):1–42, 2000.
10. J. Han and M. Kamber. Data mining: Concepts and techniques. In *ACM SIGMOD*, volume 31, June 2002.
11. J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu. Freespan: frequent pattern-projected sequential pattern mining. In *KDD ’00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 355–359, New York, NY, USA, 2000. ACM Press.
12. K. S. Hornsby and K. King. Modeling Motion Relations for Moving Objects on Road Networks. *GeoInformatica*, 12(4):477–495, 2008.
13. B. Hugueney. *Representations symboliques de longues series temporelles (Symbolic representations of long temporal series)*. PhD thesis, University Paris 6, 2003.
14. B. Hugueney. Adaptive Segmentation-Based Symbolic Representations of Time Series for Better Modeling and Lower Bounding Distance Measures. In *Proc. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 542–552, 2006.
15. M. Joliveau. *Reduction of Urban Traffic Time Series from Georeferenced Sensors, and extraction of Spatio-temporal series - in French*. PhD thesis, Ecole Centrale Des Arts Et Manufactures (Ecole Centrale de Paris, 2008.
16. M. Joliveau and F. De Vuyst. Recherche de motifs spatio-temporels de cas atypiques pour le trafic routier urbain. In *Extraction et Gestion de Connaissances EGC 08, Revue des Nouvelles Technologies de l’Information - RNTI - E11, F. Guillet et B. Trousse*, pages 523–534. Cépaduès, 2008.
17. M. Joliveau, C. B. Medeiros, G. Jomier, and F. De Vuyst. Managing Sensor Data on Urban Traffic. In *Proceedings 3rd SeCoGIS Workshop*, 2008.

18. M. Joliveau and F. De Vuyst. Space-time summarization of multisensor time series. case of missing data. In *Int. Workshop on Spatial and Spatio-temporal data mining, IEEE SSTDM*, 2007.
19. I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
20. E. Keogh, K. Chakrabarti, M. Pazzani, and Mehrotra S. Dimensionality reduction for fast similarity search in large time series databases. *Journal of Knowledge and Information Systems*, 2000.
21. E. Keogh, K. Chakrabarti, M. Pazzani, and Mehrotra S. Locally adaptive dimensionality reduction for indexing large time serie databases. In *Proc. of the international IEEE Conference on Data Mining (ICDM)*, pages 151 – 162, 2001.
22. E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmentation time series. In *Proc. of ACM SIGMOD International Conference*, pages 289 – 296, 2001.
23. K. Kim, M. Lopez, S. Leutenegger, and K. Li. A Network-based Indexing Method for Trajectories of Moving Objects. In *LNCS 4243*, pages 344–353, 2006.
24. Hans-Peter Kriegel, P. Kröger, P. Kunath, M. Renz, and T. Schmidt. Proximity queries in large traffic networks. In *Proc. ACM GIS*, pages 1–8, 2007.
25. J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *DMKD '03: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM Press, 2003.
26. C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve. Data Aggregation and Roadside Unit Placement for a VANET Traffic Information System. In *Proceedings VANET'08*, pages 58–65. ACM Press, 2008.
27. L. Mariotte, C. B. Medeiros, and R. Torres. Diagnosing Similarity of Oscillation Trends in Time Series. In *International Workshop on spatial and spatio-temporal data mining - SSTDM*, pages 243–248, 2007.
28. T. Mautora and E. Naudin. Arcs-states models for the vehicle routing problem with time windows and related problems. *Computers and Operations Research*, 34:1061–1084, 2007.
29. C. B. Medeiros, O. Carles, F. Devuyst, G. Hebrail, B. Hugueney, M. Joliveau, G. Jomier, M. Manouvrier, Y. Naja, G. Scemama, and L. Steffan. Towards a data warehouse for urban traffic (in french). *Revue des Nouvelles Technologies de L'Information*, RNTI(B2):119–137, 2006.
30. G. Scemama and O. Carles. Claire-SITI, Public road Transport Network Management Control: a Unified Approach. In *12th IEEE Int. Conf. on Road Transport Information and Control (RTIC 04)*, 2004.
31. C.E. Shannon and W. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1963.
32. W. Shen and H. M. Zhang. On the morning commute problem in a corridor network with multiple bottlenecks: ITS system-optimal traffic flow patterns and the realizing tolling scheme. *Transportation Research part B*, 43:267–284, 2009.
33. S. Spaccapietra, C. Parent, M. L. Damiani, J. A. Macedo, F. Porto, and C. Vangenot. A conceptual view on trajectories. *Knowledge and Data Engineering*, 65(1):126–146, 2008.
34. R. Stough and G. Yang. Intelligent Transportation Systems. In Eolss Publishers, editor, *Encyclopedia of Life Support Systems (EOLSS), Developed under the Auspices of the UNESCO*, Oxford ,UK, 2003.
35. J.B. Tenenbaum and J.C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290:2319–2323, 2000.
36. D. Tiesyte and C. Jensen. Similarity-based Prediction of Travel Times for Vehicles Traveling on Known Routes. In *Proceedings ACM GIS*, 2008.
37. B. K. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary Lp norm. In *Proc. of the 26th VLBD Conference*, pages 385 – 394, 2000.