

Discovering trend clusters in sensor data streams

Scoperta di raggruppamenti di tendenza in flussi di dati provenienti da sensori

Donato Malerba and Annalisa Appice

Abstract Nowadays sensors are deployed everywhere in order to support real-time data applications. They periodically gather information along a number of attribute dimensions (e.g., temperature and humidity). Applications typically require monitoring these data, fast computing aggregates, predicting unknown data, or issuing alarms. To this aim, this paper introduces a recently defined spatio-temporal pattern, called trend cluster, and its multiple applications to summarize, interpolate and detect outliers in sensor network data. As an example, we illustrate the application of trend cluster discovery to air climate data monitoring.

Abstract *I sensori sono oggi installati ovunque, al fine di supportare applicazioni in tempo-reale. Essi collezionano periodicamente dati per un certo numero di attributi (per es., temperatura e umidità). Le applicazioni tipicamente richiedono il monitoraggio di questi dati, la computazione veloce di loro aggregati, la predizione di valori mancanti o la generazione di allarmi. A tal fine, questo articolo introduce un nuovo pattern di dati spatio-temporale, chiamato trend cluster (raggruppamento di tendenza), e le sue molteplici applicazioni nel creare sommari, calcolare interpolazioni e scoprire anomalie in dati provenienti da una rete di sensori. Come esempio, si illustra una applicazione della scoperta di trend cluster al monitoraggio di dati climatici.*

Key words: summarization, interpolation, outlier detection.

1 Introduction

Nowadays, the pervasive geosensor networks which measure several physical variables (e.g., atmospheric temperature, pressure, humidity or energy production) en-

Donato Malerba, Annalisa Appice
Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro", via Orabona 4, I-70125
Bari, Italy e-mail: donato.malerba@uniba.it, annalisa.appice@uniba.it

able us to monitor and study dynamic physical phenomena at granularity details that were impossible before. A major challenge posed by a geosensor network is to combine the sensor nodes in computational infrastructures, which are able to produce globally meaningful information from data obtained by individual sensor nodes. These infrastructures should use appropriate primitives to account for both the spatial dimension of data, which determines the ground location of a sensor, and the temporal dimension of data, which determines the ground time of a reading.

A recently investigated primitive is the *trend cluster*. It models a cluster of neighbour sensors, which measure data whose temporal variation is similar over a time window. A trend cluster is a local pattern which accounts for the spatial autocorrelation of a geophysical variable for a given time window. Spatially related sensors are expected to measure similar values along time. When this happens, information generated by sensors can be efficiently summarized by a set of trend clusters. The computational efficiency of the summarization can be improved by an incremental strategy which continuously maintain the current set of trend clusters as the time window slides. Trend clusters can also be used for spatio-temporal interpolation, that is, to estimate unknown data in a specific geographical location at a specific time point. Finally, their analysis is also useful to detect possible changes in the underlying data distribution as well as anomalies.

In this work, we report recent studies [1, 2, 3, 4] on the discovery of trend clusters in data generated by sensor networks, and we describe their use in algorithms developed for summarization, interpolation and anomaly detection. We also discuss an application to monitoring air climate data.

2 Basics

Let us consider a numeric geophysical field Z , periodically sampled at the discrete time points of a time line T , from the fixed-to ground sensors of a network K . A *data snapshot* $\langle K(t), z_K(t) \rangle$ is the set of observations sampled for Z from the set of sensors $K(t) \subseteq K$, which are active at the time point $t \in T$. $z_K(t)$ is the field function $z_K(t) : K_t \mapsto Z$ that assigns the sensor location $(x, y) \in K_t$ to the value here observed for Z at t . A *sensor data stream* $z(T, K)$ is a stream of data snapshots, which arrive at the consecutive time points of T , that is,

$$\langle K(t_1), z_K(t_1) \rangle, \langle K(t_2), z_K(t_2) \rangle, \dots$$

A *trend cluster* is a spatio-temporal pattern $\mathcal{T}$, initially defined in [2], for the summarization of a sensor data stream. It is a triple $(H, \mathcal{C}, \mathcal{Z})$, where: (1) H is a time interval $[t_{start}, t_{end}]$ such that $t_{start}, t_{end} \in T$ and $t_{start} < t_{end}$; (2) $\mathcal{C} \subseteq K$ is a set of “spatially close” sensors that sample data of Z such that they show “similar trend” along the time horizon H ; (3) $\mathcal{Z}$ is the time series of the trend data sampled from sensors of $\mathcal{C}$ along H . Each point in the time series is a set of aggregating statistics (e.g. mean, minimum, maximum) of data measured by the sensors in $\mathcal{C}$.

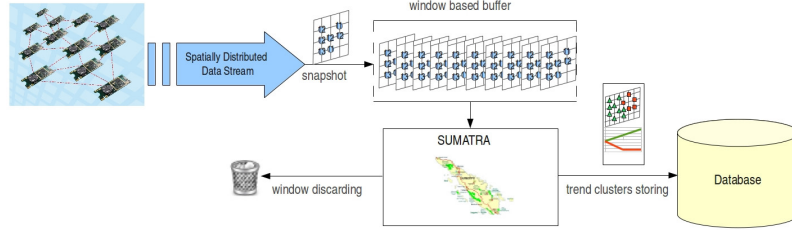


Fig. 1 SUMATRA framework.

In [2, 3], we present an algorithm to discover the trend cluster model of a sensor data stream by resorting to the count based model. This model decomposes a stream into consecutive (non-overlapping) windows of fixed length. Trend clusters, discovered with the count-based model, are used for both the summarization [2] and the interpolation [3] of the stream. In [4], we describe an incremental learning strategy to discover the trend cluster model of a sensor data stream by resorting to the sliding window model. This is a first-in, first-out data structure, so that when a snapshot is acquired and inserted in the window, the oldest snapshot is discarded. Trend clusters, discovered with the sliding-window model, are used for the detection and classification of outliers in the stream [2].

3 Summarization

SUMATRA [1, 2] is a summarization algorithm that allows us to process a sensor data stream by computing trend clusters along the horizon time of a count-based window (see Figure 1). The summarization process is three-stepped: (1) the snapshots of a window are collected; (2) the trend clusters are computed; (3) the window is discarded, while trend clusters are stored in the database. Input parameters of trend cluster discovery are the window length w , the threshold on spatial closeness d and the similarity threshold for the trend δ .

The discovery process is triggered each time a new data window is acquired from the stream. An unclustered sensor u is randomly chosen as the seed of a new empty cluster $\mathcal{C}$. Then u is added to $\mathcal{C}$ and the trend polyline prototype $\mathcal{Z}$ (i.e. the time series measured by u over the window time horizon) is initialized. Both $\mathcal{C}$ and $\mathcal{Z}$ are expanded by using u as the seed of the expansion process. The expanded trend cluster is added to the pattern set, while the clustering process is iteratively repeated until all sensors are assigned to a cluster. The expansion process of $[\mathcal{C}, \mathcal{Z}]$ is driven by a seed sensor u and it is recursively defined. First, the neighborhood $\eta(u)$ is constructed by considering the unclustered sensors that are distant less than d from the seed. Then, the candidate cluster $temp\mathcal{C} = \mathcal{C} \cup \eta(u)$ and the associated trend polyline prototype $temp\mathcal{Z}$ (i.e. time series of the cluster aggregates over window

time horizon) are computed. The trend purity of $[temp\mathcal{C}, temp\mathcal{Z}]$ is computed. If the trend purity property is satisfied, then sensors of $\eta(u)$ are clustered into $\mathcal{C}$ and the last computed $temp\mathcal{Z}$ is assigned to $\mathcal{Z}$. Otherwise, the addition of each sensor of $\eta(u)$ to $\mathcal{C}$ is evaluated node-by-node. In both cases, sensors newly clustered in $\mathcal{C}$ are iteratively chosen as seeds to continue the expansion process. The expansion process stops if no new sensor is added to the cluster. In this process, the trend purity of a trend cluster $[\mathcal{C}, \mathcal{Z}]$ is computed as follows:

$$purity([\mathcal{C}, \mathcal{Z}]) = \begin{cases} true & \text{iff } \frac{1}{|\mathcal{C}|} \sum_{u \in \mathcal{C}} sim(u, \mathcal{Z}) = 1 \\ false & \text{otherwise} \end{cases}, \quad (1)$$

where $|\mathcal{C}|$ denotes the cardinality of $\mathcal{C}$ and $sim(u, \mathcal{Z}) = 1$ if $\forall t = 1, \dots, w : |u[t] - z[t]| \leq \delta$, 0 otherwise.

4 Interpolation

The interpolation algorithm TRECI [1, 3] uses trend clusters to estimate unknown measures of the sensor data stream in a location or time point of interest. It operates in two phases. The on-line phase (see Figure 2(a)) consumes data snapshots and calls SUMATRA for the trend cluster discovery. For each trend cluster, a shape-dependent sample of clustered sensors is extracted. The sampling algorithm is designed to keep only the information which is useful to sketch the real spatial extent of the clustered region. At the same time, a (polynomial) regression model of the time law underlying the trend time series is determined by resorting to a forward selection-based computation, and only regression coefficients are used as a model of the trend. The off-line phase (see Figure 2(b)), which is repeatable, retrieves the spatio-temporal knowledge surrounding the space-time point to be interpolated from the database. This knowledge is used to determine an Inverse Distance Weighting (IDW) based estimate of the field.

5 Outlier detection and classification

SWOD [4] is a semi-supervised algorithm to detect and classify outliers in a sensor data stream. The entire process, illustrated in Figure 3, is iterated when a data snapshot is acquired from the stream. The trend cluster model is incrementally built with the sliding window model and a forecasting model is computed from trend-clusters. Then, for each sensor the observed value is compared to its forecast, in order to identify the outliers. Since we use three exponential smoothing models (Brown, Holt and Winters), we select the one that minimizes the mean square of residuals maintained for the trend time. The outliers are classified as changes and anomalies. Each de-

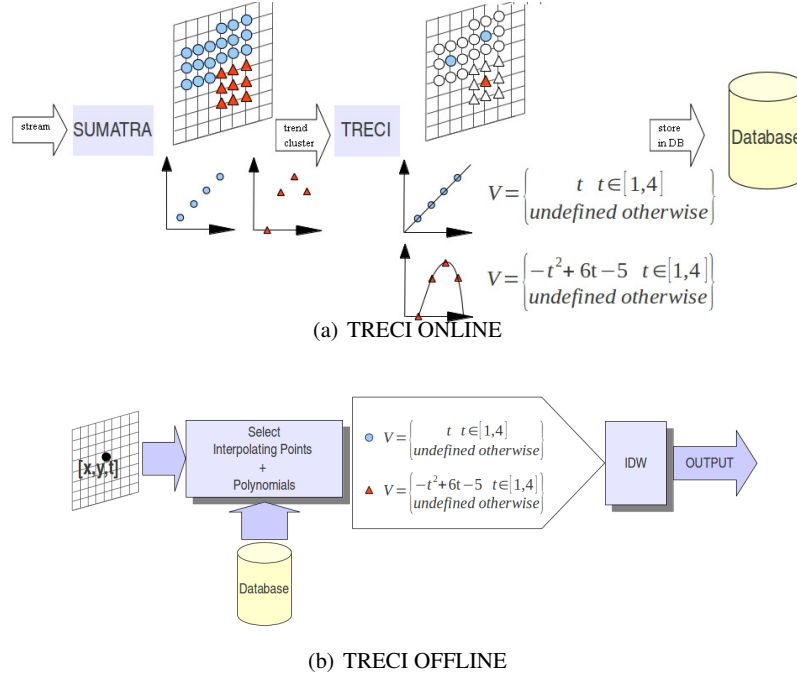


Fig. 2 TRECI: on-line summarization step and off-line interpolation step.

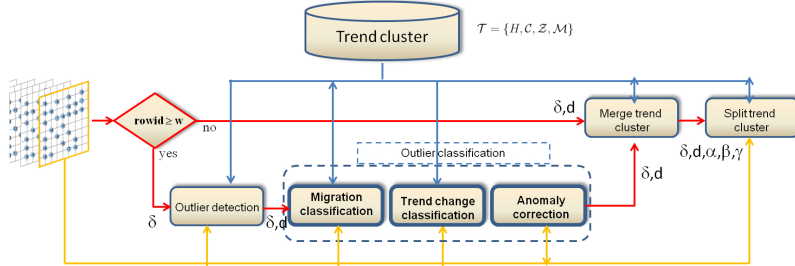


Fig. 3 SWOD: outlier detection and classification.

tected outlier is classified according to a specific interpretation of an anomaly and a data change. Anomalies are isolated abnormal event. Even when they come in groups or prolonged chunks, we cannot expect any specific spatial correlation between them. Changes are pervasive modifications of the statistical properties of the data. The classification process is three-phased.

In the first phase, we look for border outliers, which are inliers with the forecasting model of a neighbor sensor (Figures 4(a)-4(b)). They are migrant sensors, so they

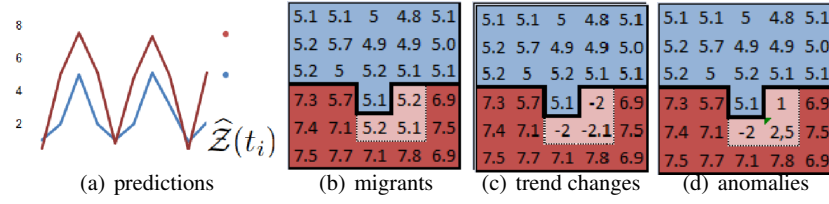


Fig. 4 Outlier classification.

are simply assigned to the model of the migration area. In the second phase, we look for changing sensors in the set of spatially correlated outliers, i.e. outliers that can be grouped in spatial clusters (Figure 4(c)). Outliers grouped in a dense cluster are classified as changes and removed from the bundle of outliers. Since the forecasting model of these sensors is no longer accurate, they are put in a quarantine period until a window of snapshots is collected and coefficients of a new model are estimated. Finally, in the third phase, we classify left-over outliers as anomalies (Figure 4(d)) and correct them in the snapshot by using the value predicted by the forecasting model that is used in the outlier detection phase. Once outliers are classified, the trend cluster model is updated according to an incremental learning strategy that slides the trend cluster model and updates it to fit the snapshot processed in the last round. This strategy performs a merge-and-split procedure that operates with each anomaly corrected in the snapshot as well as each migrant sensor re-assigned to the destination trend cluster.

6 The application

We illustrate an application of the trend cluster discovery to monitor air climate data measured across South America. We consider monthly-mean temperature (in $^{\circ}\text{C}$) recorded between 1960 and 1990 over a 0.5 degree by 0.5 degree of latitude/longitude grid of South America¹. The sensors are centered on 0.25 degrees for a total of 6477 sensors. The temperature ranges in $[-7.6 \div 32.9]^{\circ}\text{C}$. We discover trend clusters with $d = 0.5^{\circ}$ and use them for summarization, interpolation, outlier detection and classification.

First, we analyze the accuracy, efficiency and summarization degree of the trend cluster model, which is discovered by varying the window length between 1, 6, 12 and 24, as well as the similarity threshold δ between 2°C , 4°C and 8°C . The accuracy is measured by the root mean square error (see Figure 5(a)), while the efficiency is measured by both the time (in milliseconds) spent summarizing a window (see Figure 5(b)) and the computation time (in milliseconds) spent summarizing the entire stream

¹ http://climate.geog.udel.edu/~climate/html_pages/archive.html

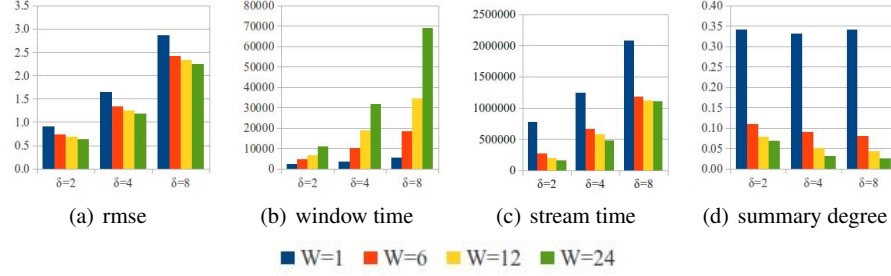


Fig. 5 South American Airclimate: summarization results.

Table 1 South American Airclimate: interpolation error (rmse) computed on the whole stream.

Baseline	50% Sensors switching-off	50% Time points jumping-on	50% Sensors switching-off and 50% time points jumping-on
1.97	2.48	2.72	2.90

(see Figure 5(c)). The summarizing degree is the ratio of the size of the trend cluster summary to the size of the original stream (see Figure 5(d)). Results on accuracy reveal that the root mean square error (rmse) is always significantly below δ . The trend clusters discovered window-by-window ($w > 1$) summarize a stream better than spatial clusters, discovered snapshot-by-snapshot ($w = 1$). By enlarging w the accuracy of the summary increases. Results on computation times reveal that computing trend clusters in a window is more time-consuming than computing spatial clusters in a snapshot. On the other hand, the total time spent processing the entire stream and computing spatial clusters at each snapshot is higher than the time spent computing trend clusters (independently of the window size). Finally, results on summarization degree show that by increasing δ , although the accuracy of summarization decreases slightly, the size of the compressed stream is much lower. Further results are reported in [2].

Second, we analyze the accuracy of the IDW-based interpolation that is yield with a trend cluster model discovered with $w = 12$ and $\delta = 4$. Several experimental settings are considered. First, 50% of the sensors is switched-off in each snapshot for the trend cluster discovery. Second, 50% of snapshots is jumped in the streaming line for the trend cluster discovery. Finally, 50% of the sensors is switched-off and 50% of the snapshots is jumped at the same time for the trend cluster discovery. The error performed in interpolating the stream as a whole is compared to the baseline case where no sensor is switched-off and no snapshot is jumped. Results, collected in Table 1, reveal that the interpolation error (rmse) is below δ when both new spatial and new temporal points are considered in the interpolation phase (i.e. both sensors are switched-off in the network and snapshots are jumped in the streaming line of the summarization phase). This result confirms the effectiveness of the interpolation even when a sparse network is processed. Further results are reported in [2].

Table 2 South American Airclimate: outlier detection and classification. Average computation time (in msec), maximum number of sensors per snapshot, number of anomalies (A), migration change (M) and spatial pervasive change (C). Averages are computed per snapshot.

time	Nr of sensors	SWOD			time	A
		A	M	C		
1329.6	6477	3.873	176.857	19.634	157.9	70.968

Finally, we analyze the trend cluster model discovered with the sliding window model (window length $w = 12$ and similarity threshold $\delta = 4$) and we evaluate the outlier detection and classification based on this model. We compare outliers detected by using the trend cluster model (SWOD) to the outliers detected the exponential smoothing model learned sensor by sensor (TSA-ES). Results reported in Table 2 reveal that the number of outliers classified as anomalies is lower in SWOD, which detects more data changes. The analysis of the computation times reveals that the SWOD discovery process of a data snapshot is fast, even when the network consists of thousands of sensors. Further results, reported in [4], investigate the accuracy of the outlier and change detection in artificial data streams.

7 Conclusion

We have presented a spatio-temporal pattern, called trend cluster, which is local both in space and time and accounts for both spatial and temporal autocorrelation. We have shown how this pattern can be efficiently discovered from streaming sensor data by resorting to the count-based stream model, as well as the sliding window stream model. Interestingly, it is useful for various tasks (summarization, interpolation and outlier detection).

References

- [1] A. Appice, A. Ciampi, F. Fumarola, and D. Malerba. *Data Mining Techniques in Sensor Networks - Summarization, Interpolation and Surveillance*. Springer Briefs in Computer Science. Springer, 2014.
- [2] A. Appice, A. Ciampi, and D. Malerba. Summarizing numeric spatial data streams by trend cluster discovery. *Data Mining and Knowledge Discovery*, 2014. (in press).
- [3] A. Appice, A. Ciampi, D. Malerba, and P. Guccione. Using trend clusters for spatiotemporal interpolation of missing data in a sensor network. *Journal of Spatial Information Science*, 6(1):119–153, 2013.
- [4] A. Appice, P. Guccione, D. Malerba, and A. Ciampi. Dealing with temporal and spatial correlations to classify outliers in geophysical data streams. *Information Sciences*, 2014. (in press).