ISSN 2039-7941
Anno 2012_Numero 224

rapporti tecnici INGV

Segmentazione delle serie temporali nell'analisi dei dati: un esempio di applicazione a dati sismo-vulcanici

224

Istituto Nazionale di Geofisica e Vulcanologia

Segreteria di Redazione
Francesca Di Stefano - coordinatore
Tel. +39 06 51860068
Fax +39 06 36915617
Rossella Celi
Tel. +39 06 51860055
Fax +39 06 36915617

redazionecen@ingv.it

ISSN 2039-7941
Anno 2012_Numero 224

rapporti tecnici INGV

SEGMENTAZIONE DELLE SERIE TEMPORALI NELL'ANALISI DEI DATI: UN ESEMPIO DI APPLICAZIONE A DATI SISMO-VULCANICI

Placido Montalto[1], Marco Aliotta[1,2], Andrea Cannata[1], Carmelo Cassisi[2]

[1]**INGV** (Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Catania - Osservatorio Etneo)
[2]**UNIVERSITÀ DEGLI STUDI DI CATANIA** (Dipartimento di Matematica e Informatica)

224

Indice

Introduzione

Il presente lavoro ha lo scopo di illustrare l'applicazione del processo di segmentazione delle serie temporali, largamente impiegato nei processi di estrazione di conoscenza da banche dati detto *Knowledge Discovery in Database* [KDD; e.g. Fayyad et al., 1996]. Aggiungendo la dimensione temporale ai dati archiviati, si ottiene quello che in letteratura viene chiamato *time series database* (TSDB). In questo ambito la rappresentazione dell'informazione, cioè il modo di descrivere un insieme di dati mediante caratteristiche descrittive, riveste un ruolo cruciale.

In generale, i passi per l'estrazione di conoscenza da un insieme di dati possono essere riassunti nello schema di figura 1. Partendo da un insieme di dati, che costituisce il dominio di informazione su cui operare, l'approccio KDD prevede un processo di selezione allo scopo di definire un *dataset* di ordine ridotto come, ad esempio, un sottoinsieme di interesse dei dati di partenza. Su quest'ultimo insieme si opera un processo di filtraggio al fine di distinguere ciò che è informazione di interesse dal rumore (*noise*). Il passo successivo riguarda la riduzione dei dati (*data reduction and projection*); in questo caso si opera un processo di selezione e rappresentazione dei dati con l'intento di ridurre la quantità di informazione da processare. Per ottenere modelli e regole per la classificazione dei dati, su quest'ultimo insieme è possibile applicare algoritmi di *machine learning* quali, ad esempio, algoritmi di *clustering* e di classificazione [e.g. Cannata et al., 2011; Cassisi et al., 2011]. In questo contesto, l'obiettivo degli algoritmi di segmentazione è quello di fornire una possibile rappresentazione di una serie temporale mediante una sua approssimazione.

I metodi per perseguire tale obiettivo sono molteplici. Nel lavoro proposto ci focalizzeremo sulle tecniche di approssimazione di una serie di dati mediante un insieme di segmenti. Una delle principali applicazioni è quella di ridurre il numero di punti mantenendo, entro certe scale di osservazione, lo stesso contenuto informativo. In quest'ottica possiamo pensare al processo di segmentazione come ad un modo di 'comprimere' i dati con lo scopo di fornire una rappresentazione più efficiente degli stessi. Gli obiettivi del KDD relativi a questo processo riguardano diversi aspetti, tra cui: misura di similarità tra serie temporali, ricerca di segmenti noti all'interno di una serie temporale, compressione delle serie temporali, analisi dei *trend* [e.g. Di Salvo et al., 2012].

Negli esempi che verranno proposti si mostrerà l'applicazione delle suddette tecniche su serie temporali contenenti molti campioni come, ad esempio, serie temporali relative all'*RMS* (*Root Mean Square*) del segnale sismico e al conteggio degli eventi sismo-vulcanici [e.g. Patanè et al., 2008]. Verrà mostrato come, su lunghi periodi, tali tecniche consentono una compressione delle serie di dati, mantenendo inalterate informazioni come la variazione dei *trend* ed i periodi di anomalia dovuti a brusche variazioni nei dati (*early warning*).

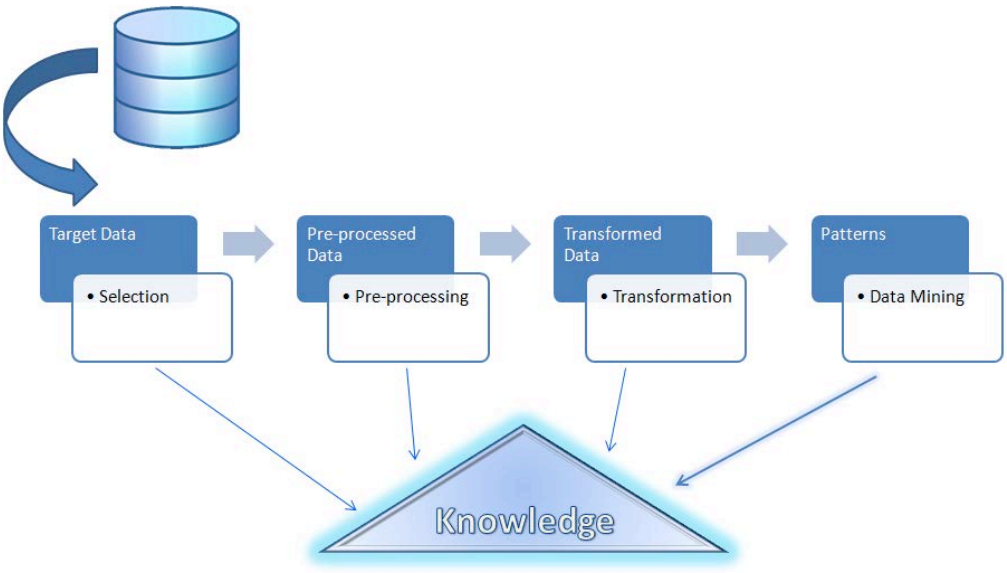

Figura 1. Schema generale del processo di *Knowledge Discovery in Database* (KDD).

1. Segmentazione delle serie temporali

Gli approcci descritti nel seguente paragrafo mostrano come, data una serie temporale, è possibile ottenere una sua rappresentazione mediante un'approssimazione a spezzate (*Piecewise Linear Approximation*: PLA; Keogh et al., 2004). Tale approssimazione è costituita da segmenti consecutivi ottenuti con opportuni criteri che verranno descritti più avanti. Nell'ottica del processo di KDD il processo di segmentazione è strettamente legato agli algoritmi di clustering, in cui ogni segmento può essere assimilato ad un cluster [Salvador and Chan, 2004].

Da un punto di vista matematico una serie temporale s è una sequenza di N campioni $x_1, x_2,...,x_N$. Indicando con s_i un generico segmento avente per estremi due campioni x_i e x_j, con $x_i < x_j$, , una k-segmentazione S_k viene definita come una sequenza $s_1,s_2,..,s_i,...,s_j,..s_k$ di segmenti contigui non sovrapposti. Intuitivamente, il metodo più semplice per operare una segmentazione è quello di considerare una finestra mobile, di dimensione fissa, all'interno della quale eseguire un'approssimazione lineare dei dati. Questo metodo, sebbene sia molto semplice da implementare, ha lo svantaggio di fornire un'approssimazione estremamente scadente nel caso di serie temporali altamente non stazionarie. L'evoluzione di questo approccio consiste nell'applicazione di una finestra mobile di lunghezza variabile in modo da ottenere una migliore approssimazione sia delle variazioni lente che di quelle veloci. Ad esempio, finestre 'larghe' possono essere impiegate per approssimare parti delle serie temporali che presentano variazioni lente, mentre finestre più piccole possono essere applicate per approssimare variazioni veloci. È chiaro che per ottenere una rappresentazione con un numero ridotto di punti, il numero dei k segmenti deve essere molto minore del numero di campioni N della serie stessa. Trovando un numero sufficientemente piccolo di segmenti k, la segmentazione rappresenta un metodo efficiente per l'archiviazione e la trasmissione dei dati. In figura 2 è riportata, a titolo di esempio, un'approssimazione mediante segmenti di un elettrocardiogramma.

Formalmente, il problema della segmentazione può essere affrontato in differenti modi, considerando vincoli come, ad esempio, il numero di segmenti k massimo da utilizzare e/o una soglia di errore calcolata tra i segmenti e la relativa porzione di serie temporale. In genere, le funzioni errore e le soglie impiegate sono definite dall'utente.

In letteratura vengono proposti algoritmi di segmentazione che possono operare in linea oppure fuori linea. I primi si riferiscono ad algoritmi dinamici applicabili in *real-time* su uno *streaming* di dati; i secondi operano fuori linea, ovvero su un *buffer* di dati noto a priori. Come riportato in Keogh et al. [2004], gli algoritmi di segmentazione possono essere suddivisi in tre categorie: *Sliding Window*, *Top-Down* e *Bottom-Up*.

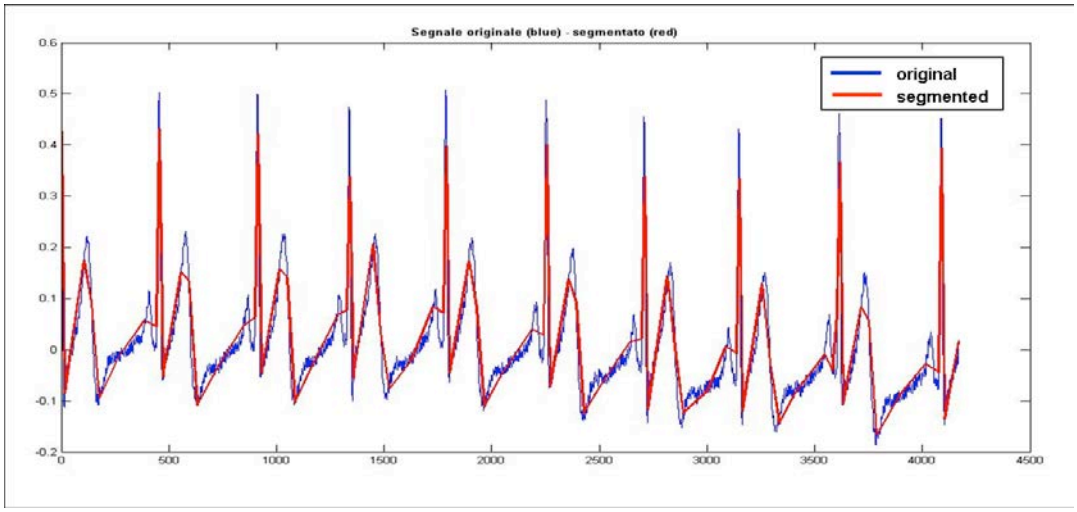

Figura 2. Esempio di segmentazione di un elettrocardiogramma. In blu la serie temporale originale, in rosso l'approssimazione ottenuta mediante spezzate.

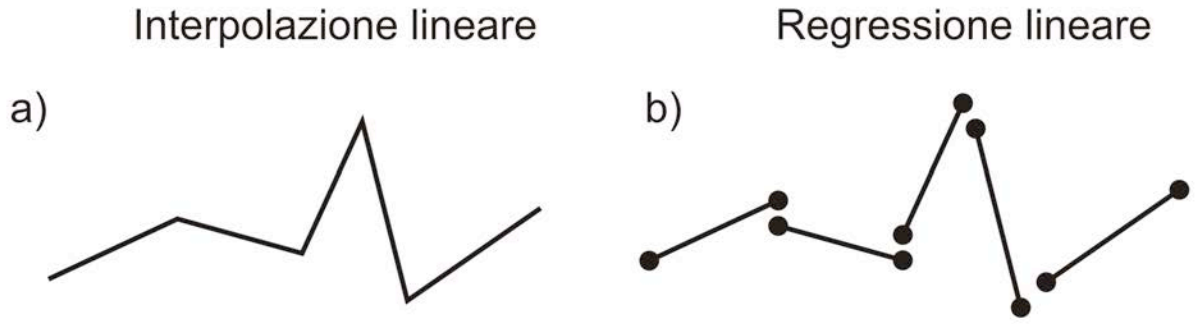

Figura 3. Interpolazione e regressione lineare ottenute usando 5 segmenti. L'interpolazione lineare (a) fornisce una rappresentazione 'continua' della serie in quanto gli estremi dei segmenti coincidono. Diversamente la regressione lineare (b) fornisce una rappresentazione a spezzate (quindi non continua) della serie. In termini di errore di approssimazione la regressione fornisce una migliore approssimazione.

Gli algoritmi di segmentazione possono impiegare sia una interpolazione lineare che una regressione lineare: nel primo caso una sequenza di punti $[x_i\ x_j]$ viene approssimata con una retta avente per estremi i valori x_i e x_j; nel caso della regressione, la sequenza di punti viene approssimata mediante un regressione lineare ai minimi quadrati. Mentre l'interpolazione lineare offre una versione segmentata continua della serie temporale (figura 3a), la regressione lineare fornisce una rappresentazione mediante segmenti discontinui. In quest'ultimo caso la qualità dell'approssimazione in termini di errore è superiore (figura 3b).
Esistono diversi metodi anche per la valutazione dell'errore sull'interpolazione/regressione. La misura utilizzata più comunemente è la media della somma dei quadrati dei residui (MSE), calcolati come differenza verticale tra la retta di interpolazione/regressione x_{si} e i punti della serie temporale x_i:

$$MSE = \frac{\sum_{i=1}^{N} \left(x_{s_i} - x_i \right)^2}{N}$$

Un'altra misura della bontà dell'interpolazione/regressione è data dalla norma infinito (L_∞) tra retta e serie di dati. Quest'ultima è data dalla massima distanza verticale tra la retta di interpolazione/regressione x_{si} e i punti della serie x_i:

$$\|E\| = \max_i \left(\left| x_{s_i} - x_i \right| \right)$$

Le tecniche di segmentazione richiedono dei parametri che direttamente o indirettamente sono legati al numero di segmenti. Tali parametri possono essere il numero k di segmenti con cui approssimare la serie, oppure una soglia di errore legata al valore k. Data una k-segmentazione S_k di una serie temporale s, è possibile definire una funzione errore $E(k)$ come, ad esempio, la somma del quadrato degli errori calcolati per ogni segmento. La scelta ottima del numero di segmenti può essere stimata mediante algoritmi di programmazione dinamica [e.g. Himberg et al., 2001] o mediante algoritmi euristici [e.g. Keogh et al., 2004]. La prima tipologia di algoritmi ha lo svantaggio di essere computazionalmente onerosa per serie temporali con un numero elevato di campioni; gli algoritmi euristici, come le tecniche bottom-up e top-down, sebbene forniscano soluzioni sub-ottime, sono quelli più utilizzati nella pratica. Indipendentemente dal tipo di funzione errore applicata e dal tipo di algoritmo euristico impiegato, i criteri per la stima del numero ottimo k di segmenti possono essere affrontati mediante differenti tecniche, tra cui quella proposta da Salvador and Chan [2004]. Questi propongono un metodo per identificare il numero k di segmenti mediante l'individuazione del 'ginocchio' (*knee*), definito come punto di massima curvatura della curva della funzione errore $E(k)$. Questo ragionamento può essere esteso a qualsiasi funzione errore come, ad esempio, funzioni distanza, funzioni di similarità etc.

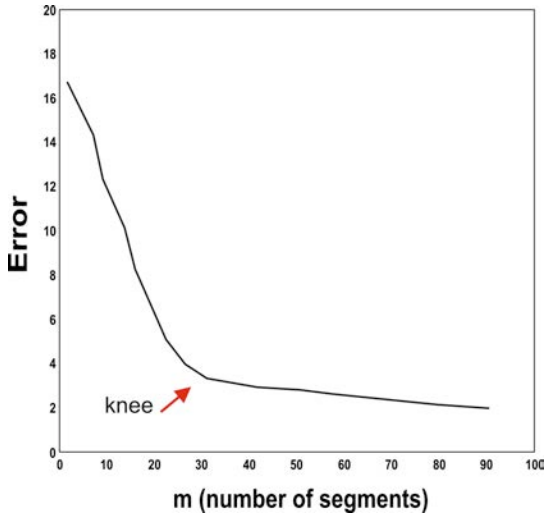

Figura 4. Errore di segmentazione al variare del numero di segmenti.

Il problema della stima del numero ottimo di segmenti richiede di operare il processo di segmentazione un numero di volte sufficientemente elevato su tutta la serie temporale, in modo da ottenere il grafico di figura 4. Da un punto di vista pratico, il caso ottimo è difficile da ottenere quando si ha la necessità di operare una segmentazione in linea su streaming di dati. Come spiegato in Ratanamahatana et al. [2010] questo processo richiede un compromesso tra accuratezza e compressione della serie temporale. Negli esempi che verranno mostrati nel prossimo paragrafo, le soglie di errore sono state scelte empiricamente in modo da ottenere una segmentazione sufficientemente compatta per scopi di visualizzazione e trasmissione dei dati.

Gli algoritmi *Sliding Window* sono di tipo *in linea*, ovvero, come spiegato in precedenza, possono operare in *real-time* su uno *streaming* continuo di dati. Questa classe di algoritmi opera considerando un segmento s, inizializzato ad una lunghezza minima l, il quale viene fatto crescere fino a quando non viene superata una certa soglia di errore definita dall'utente. Il processo viene ripetuto considerando il prossimo dato utile non incluso nel segmento precedente.

Differentemente dall'approccio *Sliding Window*, l'algoritmo *Bottom-Up* opera una suddivisione iniziale della serie temporale in un numero di segmenti che sia il più piccolo possibile (in genere il numero è fissato a $n/2$ con n numero di campioni della serie) (figura 5a). In questo caso si opererà la fusione di ogni coppia di segmenti adiacenti (detta operazione di *merging*) finché l'errore tra il segmento e la relativa porzione di dati si mantiene al di sotto di una certa soglia.

Viceversa, l'algoritmo di tipo *Top-Down* opera un partizionamento ricorsivo della serie fino a quando viene verificato un criterio di arresto (ad esempio un superamento di una soglia errore opportunamente definita). L'algoritmo considera ogni possibile partizione partendo da un numero basso k di segmenti (in questo caso l'inizializzazione considera due segmenti di lunghezza pari alla metà di quella della serie data) (figura 5b). Il numero dei segmenti viene aumentato fino a quando non si ottiene una rappresentazione con un errore di approssimazione definito dall'utente.

Negli esempi che verranno riportati nel paragrafo seguente si mostreranno risultati ottenuti mediante segmentazione di tipo *bottom-up* con interpolazione lineare.

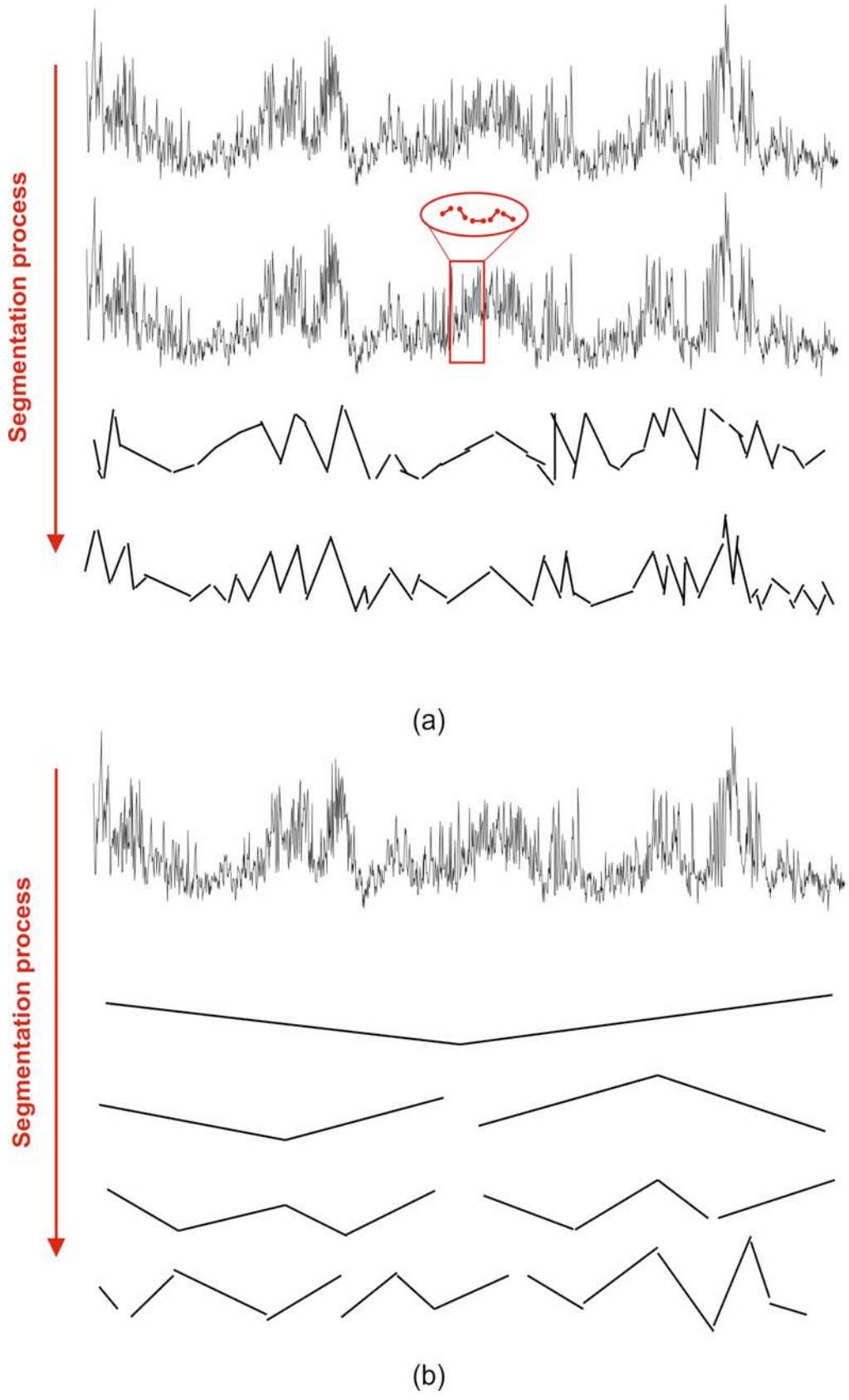

Figura 5. Algoritmo di segmentazione *Bottom-Up* (a) e *Top-Down* (b).

2. Esempio di applicazione dell'algoritmo di segmentazione ai dati sismo-vulcanici

In questo paragrafo si illustrerà una possibile applicazione degli algoritmi di segmentazione precedentemente descritti. In particolare, si mostrerà come l'impiego delle tecniche di segmentazione, applicate su serie temporali relative a dati estratti dai segnali sismici acquisiti in area vulcanica, fornisce un metodo per la compressione, la trasmissione e la visualizzazione dei dati. Le serie temporali utilizzate sono ricavate dall'*RMS* del segnale sismico acquisito alle stazioni sismiche sommitali del vulcano Etna. In particolare, l'*RMS* viene calcolato con una finestra di 10 minuti all'interno della quale viene anche eseguito il trigger automatico dei transienti sismici. Senza entrare nel dettaglio delle procedure di elaborazione dei dati, le informazioni che vengono prodotte mediante queste tecniche sono relative ad una stima dell'ampiezza del tremore vulcanico e del numero di eventi sismo-vulcanici.

Queste informazioni, correntemente utilizzate per il monitoraggio, costituiscono con il passare del tempo una grande mole di dati che devono essere trattati e archiviati. In base all'uso richiesto, spesso ci si trova a voler visualizzare i suddetti dati su intervalli temporali molto ampi. In questo caso l'utilizzo della segmentazione consente di ottenere una versione 'ridotta' delle serie temporali senza perderne il contenuto informativo. Lo schema proposto in figura 6 fornisce una rappresentazione a blocchi del sistema utilizzato per l'elaborazione delle serie temporali sopra citate. Partendo da un database in cui vengono archiviati l'*RMS* del segnale sismico e il numero orario di transienti sismo-vulcanici, si passa all'esecuzione degli algoritmi di segmentazione al fine di ridurre la dimensione delle serie temporali archiviate. I dati segmentati vengono nuovamente organizzati in una struttura analoga a quella di partenza che verrà poi impiegata per scopi di visualizzazione e/o scambio dati.

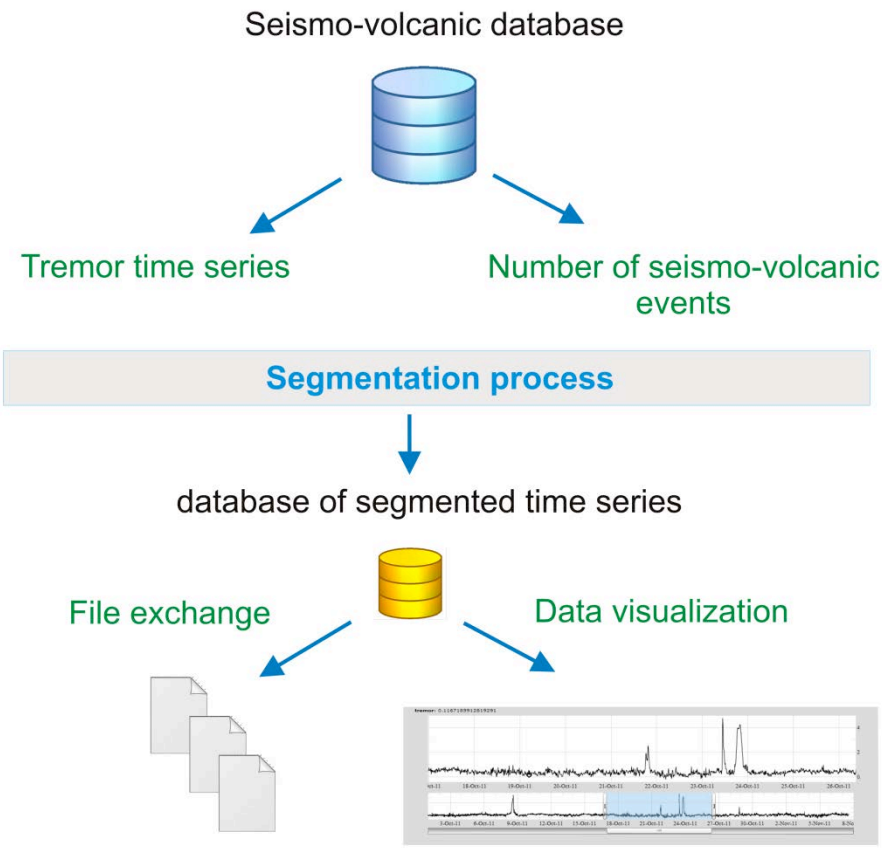

Figura 6. Schema a blocchi del sistema per la segmentazione delle serie temporali.

In figura 7 sono riportati i risultati di due processi di segmentazione relativi alla serie temporale del numero orario di eventi sismo-vulcanici registrati durante il 2009, considerando due differenti soglie di errore. La prima esecuzione, operata con una soglia elevata, fornisce una rappresentazione degli 8760 elementi mediante 42 punti (figura 7a). Come si può apprezzare dalla figura, sebbene i *trend* tendano ad approssimare l'andamento medio dei dati della serie, l'approssimazione mediante spezzate fornisce una rappresentazione poco dettagliata della serie in esame. Per migliorare il grado di approssimazione, l'algoritmo di segmentazione è stato eseguito diminuendo la soglia errore tra segmenti e dati della serie. Una migliore approssimazione, riportata in figura 7b, è stata ottenuta con 199 punti. In quest'ultimo caso la segmentazione ottenuta fornisce una rappresentazione più dettagliata della serie di partenza, approssimando con maggiore dettaglio temporale le variazioni veloci.

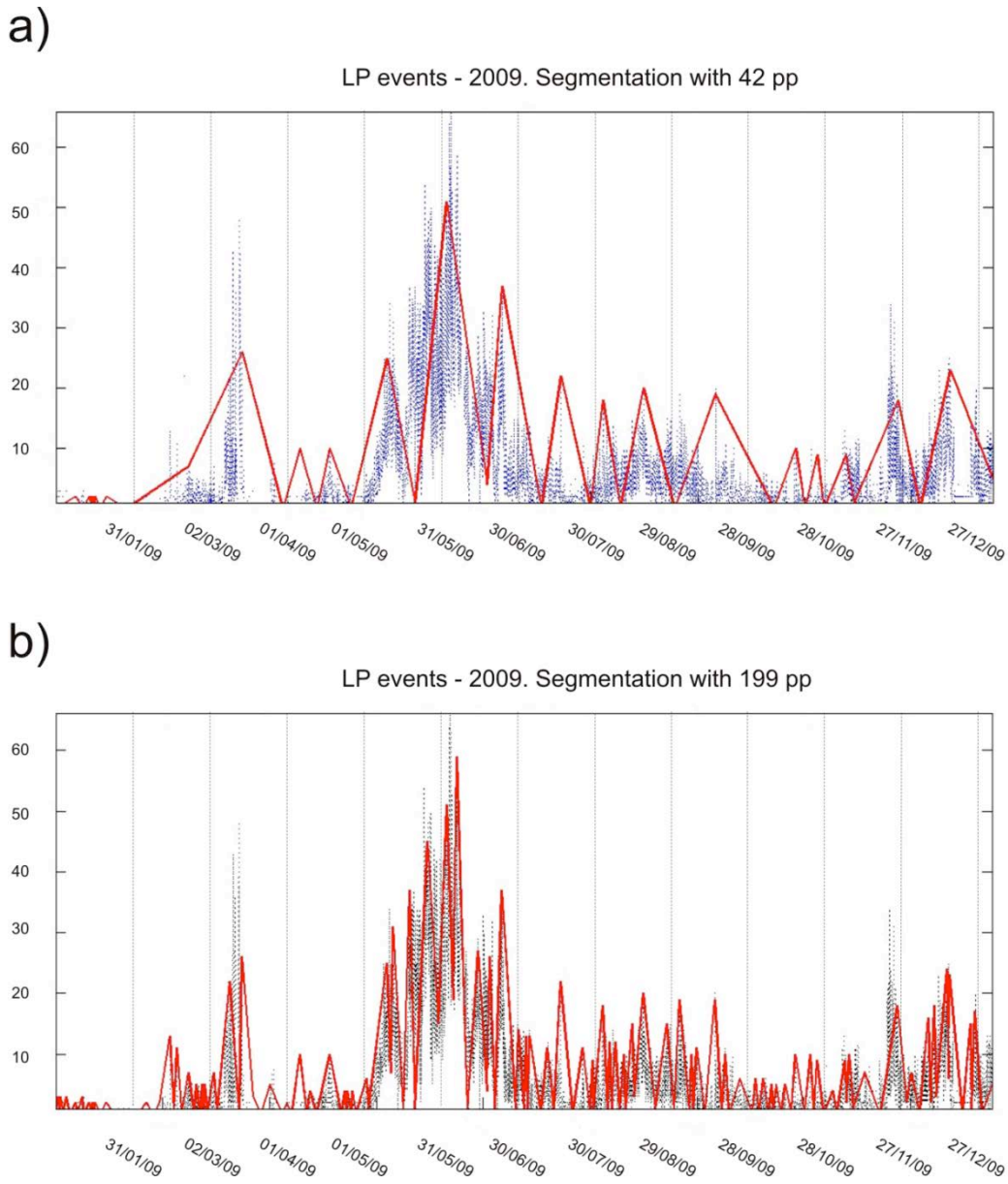

Figura 7. Rappresentazione della serie temporale relativa al numero di eventi sismo-vulcanici registrati durante il 2009; la linea tratteggiata indica la serie temporale originale mentre la linea rossa indica la serie temporale segmentata. a) Rappresentazione mediante 42 punti (soglia di errore elevata); b) rappresentazione mediante 199 punti (soglia di errore più bassa).

Analogamente a quanto visto per le serie temporali relative al numero orario di eventi sismo-vulcanici, l'algoritmo di segmentazione è stato applicato anche alle serie relative all'*RMS* del segnale sismico. Negli esempi riportati, l'ampiezza del segnale sismico è stata calcolata come *RMS* su finestre di 10 minuti filtrate nella banda di frequenza 0.5 – 5.0 Hz. In figura 8 viene riportato il grafico relativo a due diverse approssimazioni della serie composta da 52560 punti, con 532 (figura 8a) e 171 (figura 8b) punti. Come è possibile apprezzare dalla figura, entrambe le rappresentazioni mantengono inalterate le informazioni relative ai trend della serie temporale in esame. Più viene diminuita la soglia di errore tra segmenti e serie temporali di riferimento, tanto più la segmentazione tenderà a seguire fedelmente la serie di partenza. Va sottolineato come, a differenza di una semplice media mobile, il contenuto informativo relativo ai picchi della serie temporale rimanga inalterato. Questo fa si che le informazioni relative a variazioni veloci del segnale non vengano perse.

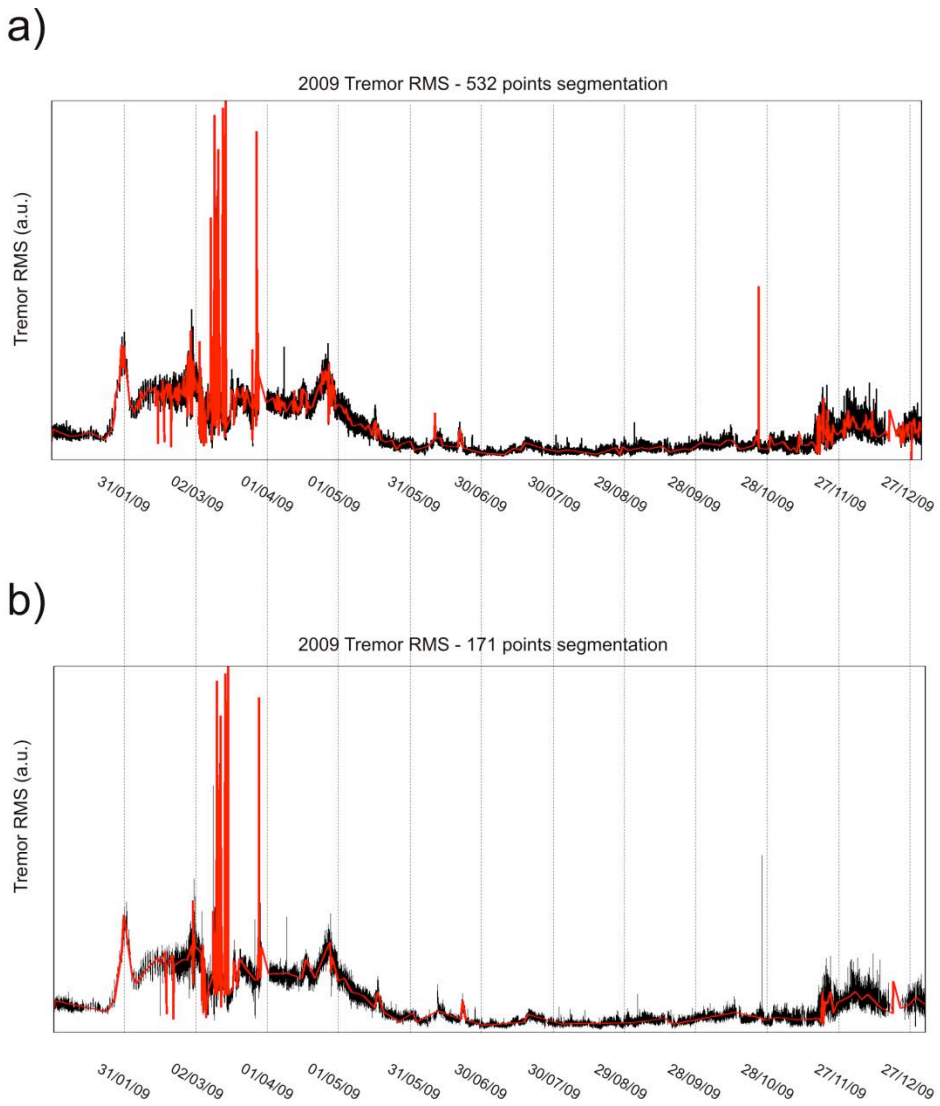

Figura 8. Rappresentazione della serie temporale, composta da 52560 punti, relativa all'*RMS* del vulcanico segnale sismico durante il 2009; la linea nera indica la serie temporale originale mentre, la linea rossa, indica la serie temporale segmentata. a) Rappresentazione mediante 532 punti; b) rappresentazione mediante 171 punti.

3. Conclusioni

Il presente report descrive quanto sviluppato dagli autori per l'analisi delle serie temporali utilizzate per il monitoraggio sismo-vulcanico del vulcano Etna. La necessità di ottenere una rappresentazione ridotta delle serie temporali ha portato alla ricerca ed alla implementazione degli algoritmi di segmentazione oggetto del presente lavoro.

Le metodologie introdotte nel paragrafo 2, largamente applicate nella disciplina del *data mining* su serie temporali, costituiscono ad oggi lo stato dell'arte per quanto riguarda le tecniche di approssimazione di serie temporali. In particolare, l'applicazione dell'algoritmo *bottom-up* ha permesso una compressione elevata dei dati, consentendo quindi una rappresentazione con un numero di punti inferiore rispetto a quello delle serie temporali di partenza. In questo contesto la scelta delle soglie errore, legata indirettamente al numero di segmenti con cui si approssima la serie temporale, è stata scelta in modo empirico. Questa scelta è stata vincolata alla dimensione dei buffer di dati da impiegare per scopi di visualizzazione ed elaborazione. Future implementazioni riguarderanno l'ottimizzazione in linea degli algoritmi *Sliding Window* in modo da operare in real-time sugli streaming di dati ed ottimizzarne l'archiviazione e la visualizzazione.

Bibliografia

Cannata, A., Montalto, P, Aliotta, M., Cassisi, C., Pulvirenti, A., Privitera, E., Patanè, D., (2011). Unsupervised clustering of infrasonic events at Mount Etna using pattern recognition techniques. Geophys. J. Int., 185, 253–264.

Cassisi, C., Giugno, R., Montalto, P., Pulvirenti, A., Aliotta, M., Cannata, A., (2011). DBStrata: a system for density-based and outlier detection based on stratification. In Proceedings of the Fourth International Conference on SImilarity Search and APplications (SISAP 2011). ACM, New York, NY, USA, 107-108.http://dl.acm.org/citation.cfm?doid=1995412.1995432.

Di Salvo, R., Montalto, P., Nunnari, G., Neri, M., Puglisi, G., (2012). Multivariate time series clustering on geophysical data recorded at Mt. Etna from 1996 to 2003. Journal of Volcanology and Geothermal Research, doi: 10.1016/j.jvolgeores.2012.02.007.

Fayyad U. et al., (1996), "From Data Mining to Knowledge Discovery in Databases", *AI Magazine,* 17(3), 37-54.

Himberg, J., Korpiaho, K., Mannila, H., Tikanmäki, J., Toivonen, H. T., (2001). Time series segmentation for context recognition in mobile devices. The 2001 IEEE International Conference on Data Mining (ICDM'01), pp. 203-210.

Keogh, E., Chu S., Hart, D., Pazzani, M., (2004). Segmenting time series: a survey and novel approach. In: Last, M., Kandel, A., Bunke, H. (Eds.), Data mining in time series database. World Scientific Publishing Company, pp. 1-21.

Patanè, D., Di Grazia, G., Cannata, A., Montalto, P., Boschi, E., (2008). The shallow magma pathway geometry at Mt. Etna volcano. Geochem. Geophys. Geosyst., 9, 12, doi:10.1029/2008GC002131.

Ratanamahatana, C. A., Lin J., Gunopulos, D., Keogh, E., Vlachos, M., Das, G., (2010). Mining time series data. *Data Mining and Knowledge Discovery Handbook 2010*, 2nd Edition. Eds. Oded Maimon, Lior Rokach. Springer., pp. 1049-1077.

Salvador, S., Chan, P., (2004). Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, pp. 576-584.

Coordinamento editoriale e impaginazione
Centro Editoriale Nazionale | INGV

Progetto grafico e redazionale
Daniela Riposati | Laboratorio Grafica e Immagini | INGV

Via di Vigna Murata, 605
00143 Roma
Tel. +39 06518601 Fax +39 065041181

http://www.ingv.it

Istituto Nazionale di Geofisica e Vulcanologia