Pattern Sequenziali e Serie Temporali

Francesco Folino

9 Dicembre 2004

Pattern Sequenziali:intuizione

Obiettivo: personalizzare ed ottimizzare le offerte di vendita ai clienti in base agli acquisti fatti da ciascun cliente in precedenza.

Analisi: studiare il comportamento nel tempo degli acquisti dei clienti !

Metodo: “il 5% dei clienti ha acquistato prima X, poi Y e poi Z”

Requisiti: mantenere traccia degli acquisti dei singoli clienti (nome, fidelity cards, carte di credito, bancomat, e-mail, codice fiscale)

Dominii: vendite al dettaglio, vendite per corrispondenza, vendite su internet, vendite di prodotti finanziari/bancari, analisi mediche

Intra-Transaction (Regole di Associazione) …
e Inter-Transaction (Patterns Sequenziali)

Transazioni con Codice Cliente

Descrizione Concettuale

Insieme di items $\{ i_1, \ldots, i_k \}$

Insieme di clienti $\{ c_1, \ldots, c_m \}$

Transazione $t \subseteq \{ i_1, \ldots, i_k \}$

Insieme di transazioni cliente

$T = \{ (c_1, data_1, t_1), \ldots, (c_n, data_n, t_n) \}$

(Una possibile) Descrizione Logica

Articoli

(Codice: INTEGER, Desc: CHAR)

Vendite

(Cliente: INTEGER, Data: DATE, Articolo: INTEGER)

*Invece di **data** si può definire qualsiasi progressivo delle transazioni del cliente (es: **numero transazione**)*

Esempio

Descrizione Concettuale

Cliente	Data	Trans
3	10/09/1999	{10}
2	10/09/1999	{10, 20}
5	12/09/1999	{90}
2	15/09/1999	{30}
2	20/09/1999	{40,60,70}
1	25/09/1999	{30}
3	25/09/1999	{30,50,70}
4	25/09/1999	{30}
4	30/09/1999	{40,70}
1	30/09/1999	{90}
4	25/10/1999	{90}

Descrizione Logica

Data	Cliente	Articolo
10/09/1999	3	10
10/09/1999	2	10
10/09/1999	2	20
12/09/1999	5	90
15/09/1999	2	30
20/09/1999	2	40
20/09/1999	2	60
20/09/1999	2	70
25/09/1999	1	30
25/09/1999	3	30
25/09/1999	3	30
25/09/1999	3	70
25/09/1999	4	30
30/09/1999	4	40
30/09/1999	4	70
30/09/1999	1	90
25/10/1999	4	90

Sequenze

Insieme di transazioni cliente

$T = \{ (data_1, c_1, t_1), \ldots, (data_n, c_n, t_n) \}$

Sequenza di transazioni per cliente c

$$seq(c) = <t_1, \ldots, t_i \ldots, t_n>$$

ordinate per data

Cliente	Sequenza
1	< {30},{90} >
2	< {10, 20}, {30}, {40,60,70}>
3	<{10}, {30,50,70}>
4	< {30}, {40,70}, {90} >
5	<{90}>

Corrispondenza (uno a uno) di nomi

Libro	Titolo
10	Star Wars Episode I
20	La fondazione e l'impero
30	La seconda fondazione
40	Database systems
50	Algoritmi + Strutture Dati =
60	L'insostenibile leggerezza
70	Immortalita'
90	I buchi neri

Sequenze e Supporti

$\langle I_1, I_2, \ldots, I_n \rangle$ è contenuta in $\langle J_1, J_2, \ldots, J_m \rangle$
se esistono interi $h_1 < \ldots < h_n$ per cui

$$I_1 \subseteq J_{h1}, \ldots, I_n \subseteq J_{hn}$$

< {30}, {90} > è contenuta in < {30}, {40,70}, {90} >
< {30}, {40,70} > è contenuta in < {10,20}, {30}, {40,50,60,70} >
ed in < {30}, {40,70}, {90} >

$$\text{Supporto}(s) = \frac{|\{ c \mid s \text{ è contenuta in } seq(c) \}|}{\text{numero clienti}}$$

Supporto(< {La seconda fondazione}, {I buchi neri} >) = 40%
Supporto(< {La seconda fondazione} >) = 80%

Patterns Sequenziali

Dato MinSupporto e l'insieme delle sequenze

$$S = \{ s \mid Supporto(s) >= MinSupporto \}$$

una sequenza in S è detta *Pattern Sequenziale* se non è contenuta in nessun altra sequenza di S

MinSupporto = 40%

< {30}, {90} > è un pattern sequenziale

Supporto(< {30} >) = 80% ma non è un pattern sequenziale perchè contenuta in < {30}, {90} >

MinSupporto = 50%

< {30}, {90} > non è in S

< {30} > è un pattern sequenziale

Tassonomie

Descrizione Concettuale

Insieme di items $I = \{ i_1, \ldots, i_k \}$

Categoria: I oppure un insieme $G = \{ g_1, \ldots, g_n \}$ **di nomi nuovi**

Tassonomia: Insieme (aciclico) di relazioni tra categorie

Antenato: x antenato di y se (x, y) è nella chiusura transitiva della tassonomia

(Due possibili) descrizioni logiche

Item	Settore
105	97
107	97
108	97
109	98
110	-
111	97
112	97
113	97
114	97
115	97
116	97
117	97
118	97
119	97

Gruppo	Reparto
61	81
65	82
62	81
63	81

Item	Gruppo	Reparto	Settore
105	61	81	97
107	61	81	97
108	61	81	97
109	65	82	98
110	-	-	-
111	62	81	97
112	62	81	97
113	62	81	97
114	62	81	97
115	62	81	97
116	62	81	97
117	62	81	97
118	62	81	97
119	63	81	97

Tassonomia

Sequential Patterns

Categoria G = { g_1, …, g_n } **Transazione t $\subseteq$ { i_1, …, i_k }**

Diciamo che t contiene g **se** g $\in$ t oppure g è un antenato di i $\in$ t

Estensione della nozione di sequential patterns

Cliente	Sequenza
1	< {30},{90} >
2	< {10, 20}, {30}, {40,60,70}>
3	<{10}, {30,50,70}>
4	< {30}, {40,70}, {90} >
5	<{90}>

Libro	Autore	Genere
10	Brooks	Fantascienza
20	Asimov	Fantascienza
30	Asimov	Fantascienza
30	Market	Fantascienza
40	Ullman	Informatica
50	Knuth	Informatica
60	Kundera	Narrativa
70	Kundera	Narrativa
90	Hawkins	Fisica

Supporto(<{Brooks},{Narrativa}>) = 40%

Supporto(<{Asimov},{Narrativa}>) = 20%

Supporto(<{Fantascienza},{Narrativa}>) = 60%

Altre Generalizzazioni

- **Sliding Windows** (transazione contenuta in più transazioni)

  $<I_1, I_2, \ldots, I_n>$ è contenuta in $<J_1, J_2, \ldots, J_m>$
  se esistono $h_1 < u_1 < \ldots < h_n < u_n$ per cui

  $I_1 \subseteq U_{k = h1..u1} J_k , \ldots, I_n \subseteq U_{k = hn..un} J_k$
  transaction-time(J_{ui}) - transaction-time(J_{hi}) < window-size per i = 1..n

  < {30}, {40,70} > è contenuta in < {30}, {40}, {70} >
  se transaction-time({70}) - transaction-time({40}) < window-size

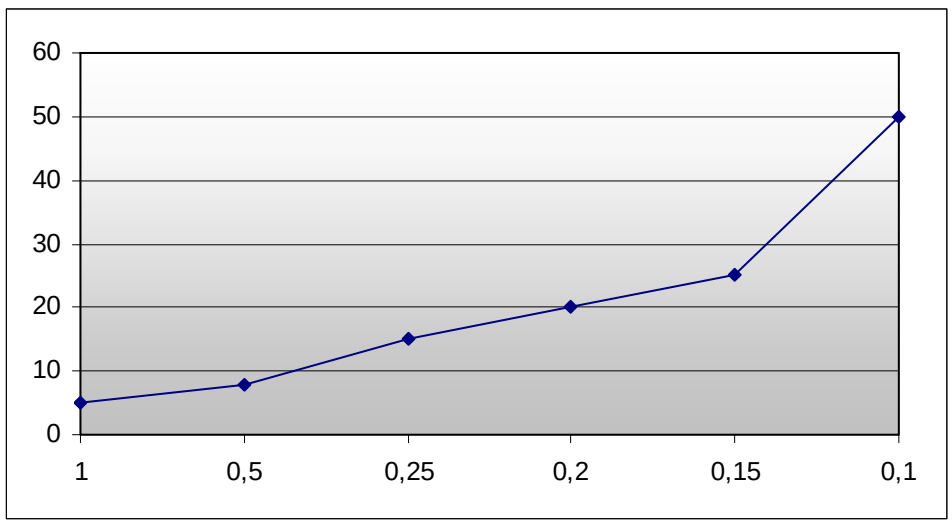
- **Time Constraints** (limite di tempo tra due transazioni)

  $<I_1, I_2, \ldots, I_n>$ è contenuta in $<J_1, J_2, \ldots, J_m>$
  se esistono $h_1 < \ldots < h_n$ per cui

  $I_1 \subseteq J_{h1} , \ldots, I_n \subseteq J_{hn}$
  mingap < transaction-time(J_{hi}) - transaction-time(J_{hi-1}) < maxgap
  per i = 2..n

Aspetti Computazionali

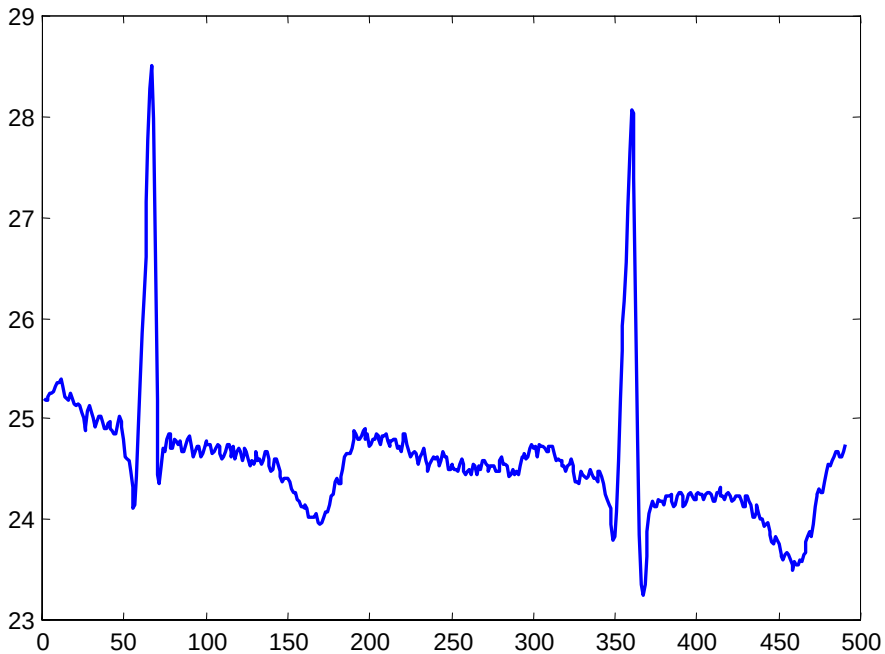
- **Mail Order: Clothes**
  - **16.000 items**
  - **2.900.000 transazioni**
  - **214.000 clienti**
  - **10 anni**
  - **Algoritmo GSP (Shrikant e Agrawal) su IBM RS/6000 250**



Cosa sono le serie temporali?

25.1750
25.2250
25.2500
25.2500
25.2750
25.3250
25.3500
25.3500
25.4000
25.4000
25.3250
25.2250
25.2000
25.1750
••
••
24.6250
24.6750
24.6750
24.6250
24.6250
24.6250
24.6750
24.7500

Una serie temporale è una collezione di osservazioni fatte sequenzialmente nel tempo.

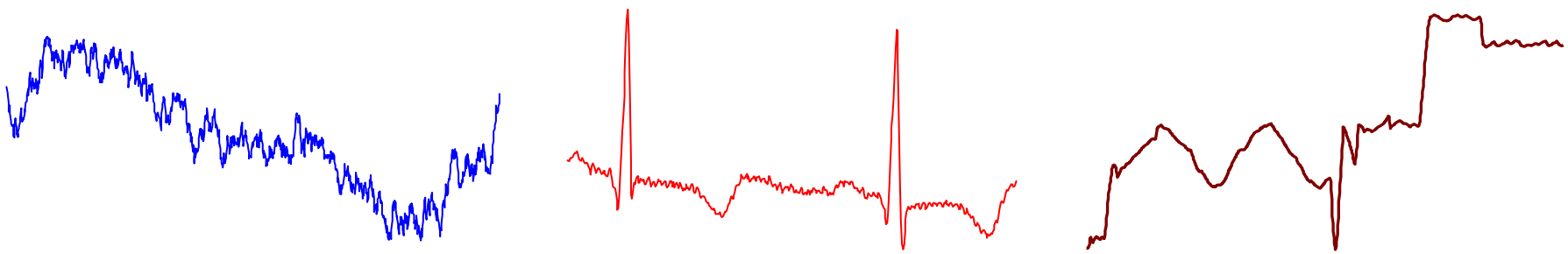


Le serie sono dappertutto! 1/2

Vengono misurate cose del genere...

- *Il tasso di approvazione del Presidente.*
- *La pressione del sangue.*
- *Il valore delle azioni di Yahoo.*
- *Il numero di pagine viste per secondo.*

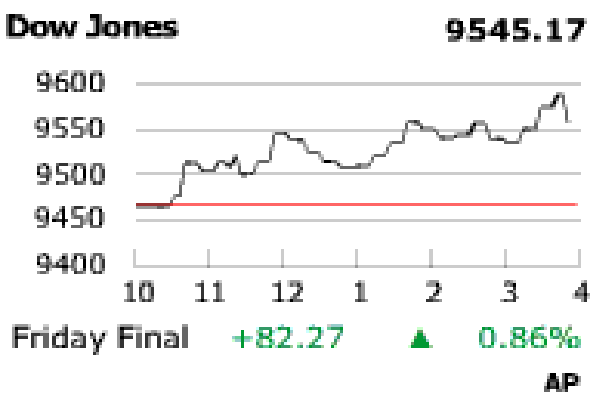
… e le cose cambiano nel tempo.

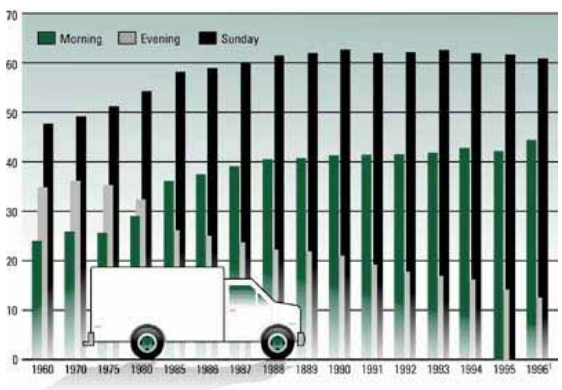


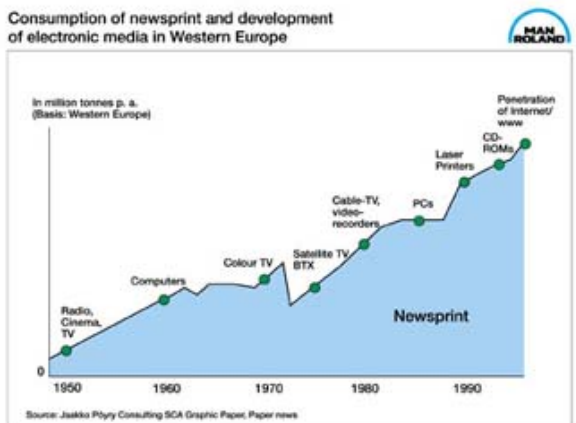
Quindi le serie temporali vengono usate in ogni dominio medico, scientifico oppure finanziario.

Le serie sono dappertutto! 2/2

Un campione casuale di 4,000 grafici da 15 giornali da tutto il mondo publicati dal 1974 al 1989 mostra che più del 75% di tutti i grafici sono serie temporali (Tufte, 1983).





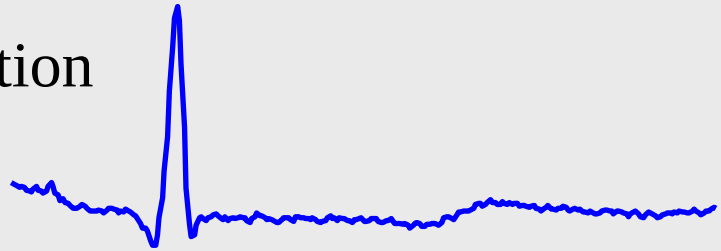


Similarità nelle serie temporali

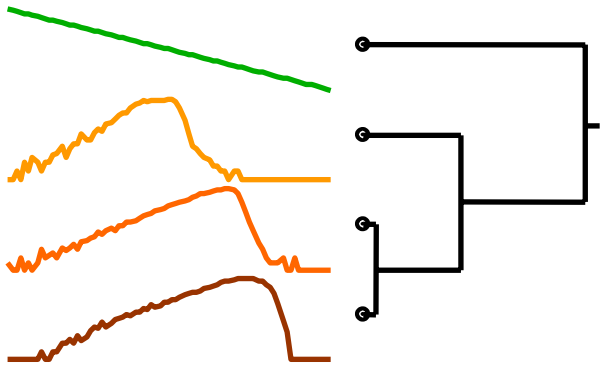
Definire la similarità tra due serie temporali è il cuore del DM sulle serie stesse

Pertanto la similarità sarà la prima cosa da affrontare.

Classification



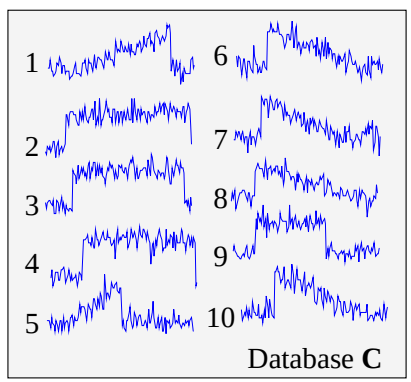
Clustering



Rule Discovery

Query by Content

Query *Q*
(template)



Perchè è così difficile lavorare con le serie temporali? (1)

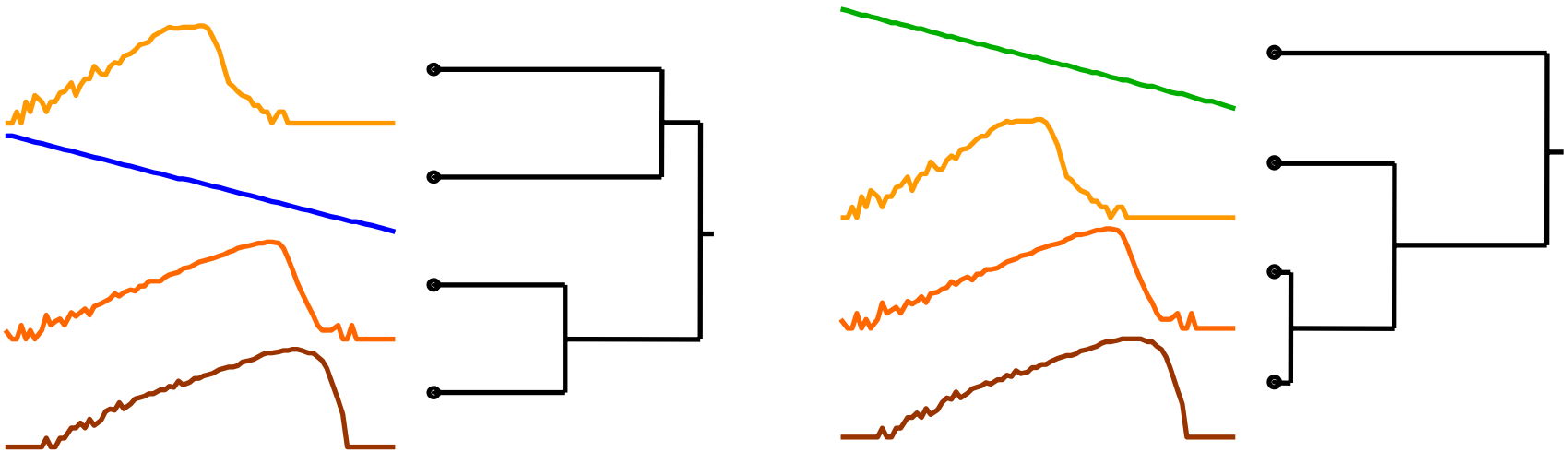
Risposta: Perchè dobbiamo lavorare con DB molto grandi

- 1 Ora di EKG data: 1 Gigabyte.
- Un tipico Weblog: 5 Gigabytes per week.
- Space Shuttle Database: 158 Gigabytes and growing.
- Macho Database: 2 Terabytes, updated with 3 gigabytes per day.

Poichè la maggior parte dei dati risiede su disco, abbiamo bisogno di una rappresentazione dei dati tale da poter essere manipolata efficientemente.

Perchè è così difficile lavorare con le serie temporali? (2)

Risposta: Abbiamo a che fare con soggettive nozioni di similarità.



La definizione di similarità dipende dall'utente, dal dominio e dal processo che manipoliamo.

Perchè è così difficile lavorare con le serie temporali? (3)

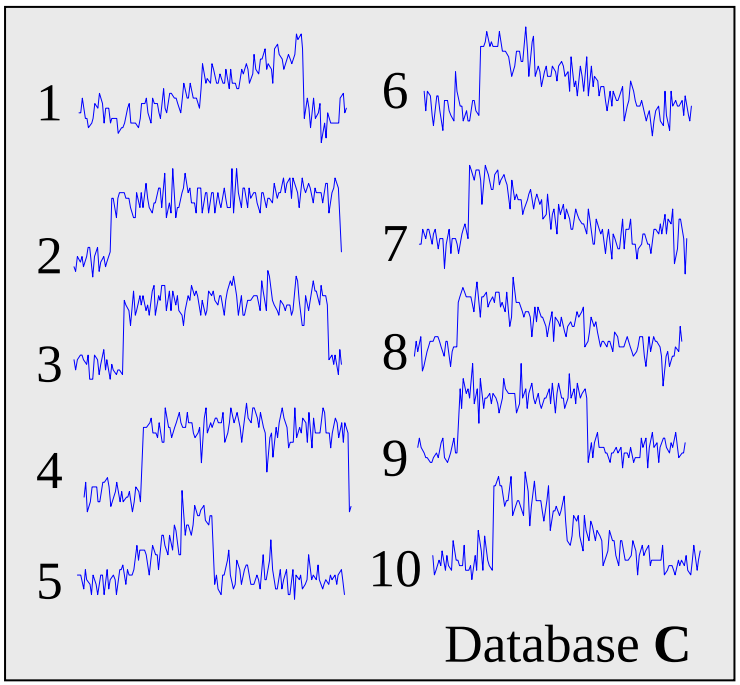
Risposta: Dati misti comportano dei problemi.

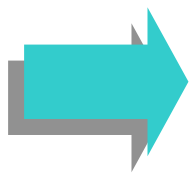
- Differenti formati di dati.
- Differenti frequenze di campionamento.
- Rumore, valori mancanti, etc.

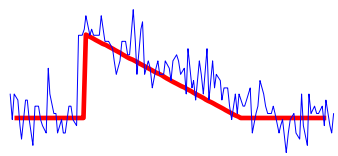
Il problema della similarità ha un duplice aspetto (1)

Query *Q*
(template)

1:Matching Totale



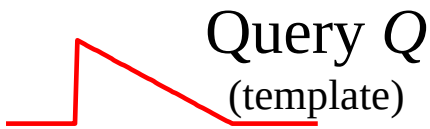




C_6 è il miglior match.

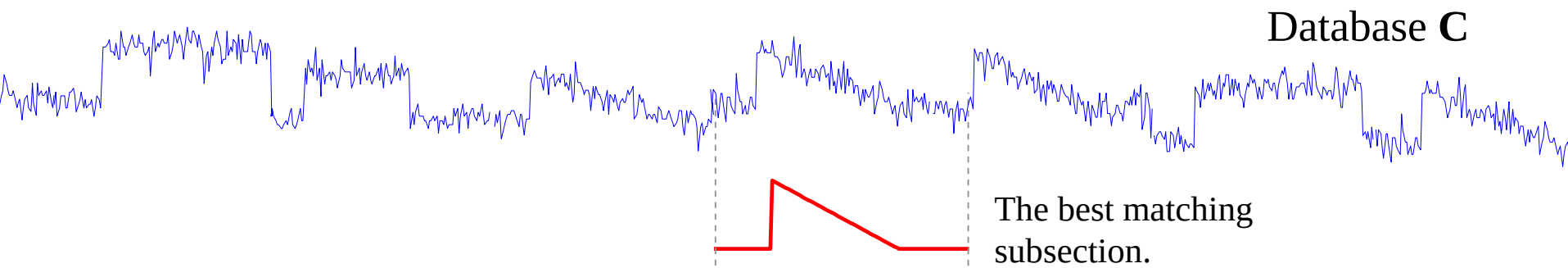
Data una Query *Q*, un database **C** ed una misura di distanza, trovare C_i che meglio metcha *Q*.

Il problema della similarità ha un duplice aspetto (2)



Query *Q*
(template)

2: Matching su una sottosequenza



Data una Query *Q*, un database **C** ed una misura di distanza, trovare la porzione che metcha meglio *Q*.

Osserviamo che possiamo sempre convertire un subsequence matching in un whole matching facendo scorrere una finestra lungo tutta la sequenza.

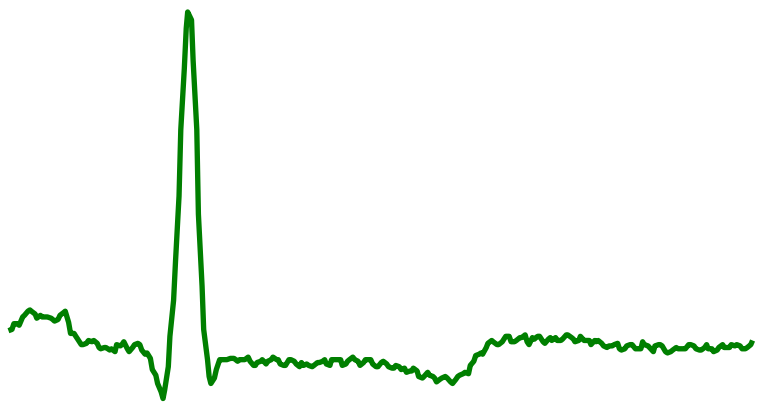
Cosa ci interessa

Andiamo dal dottore perchè avvertiamo dolori alla cassa toracica. L'ECG sembra strano…

Il dottore vuole effettuare una ricerca in un database al fine di trovare ECGS simili, nella speranza di ricavare indizi sul nostro stato...

Due domande:

Come identifichiamo i simili?
Come facciamo a cercare rapidamente?



Definiamo una misura di Distanza

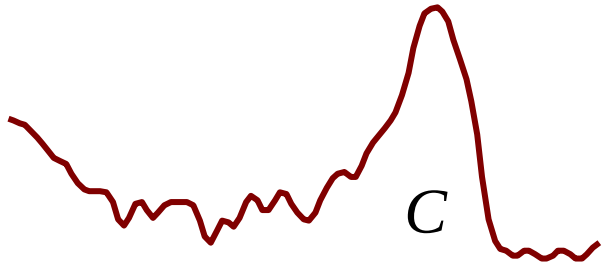
Definizione: Siano O_1 e O_2 due oggetti ricavati dall'universo dei possibili oggetti. La distanza (dissimilarità) è denotata con $D(O_1,O_2)$

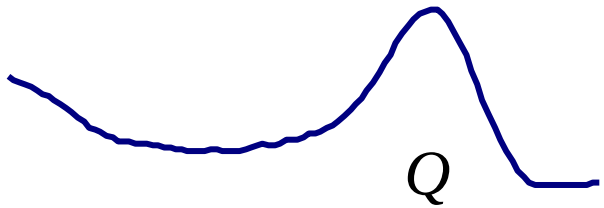
Di che proprietà deve godere tale misura di distanza?

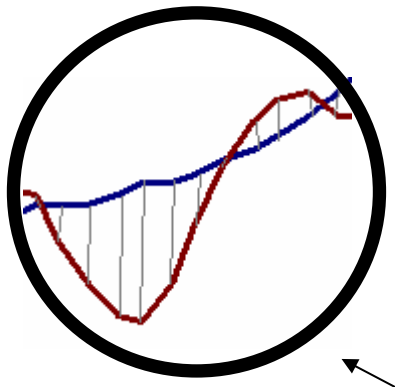
- $D(A,B) = D(B,A)$ *Simmetria*
- $D(A,A) = 0$ *Constanza dell' auto-similarità*
- $D(A,B) = 0$ IIf $A = B$ *Positività*
- $D(A,B) \leq D(A,C) + D(B,C)$ *Diseguaglianza Triangolare*

La metrica di Minkowski

$$D(Q,C) \equiv \sqrt[p]{\sum_{i=1}^{n}(q_i - c_i)^p}$$



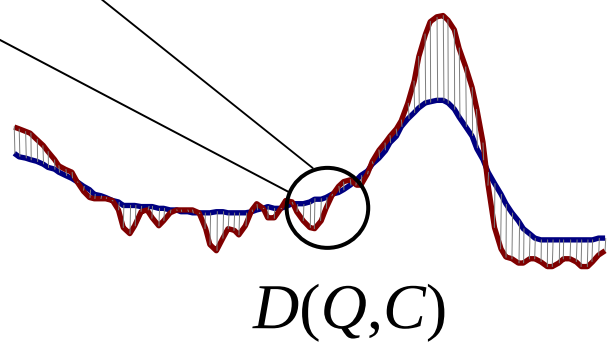




$p = 1$ Manhattan (Rectilinear, City Block)

$p = 2$ Euclidean

$p = \infty$ Max (Supremum, “sup”)



La metrica Euclidea

Date due serie temporali

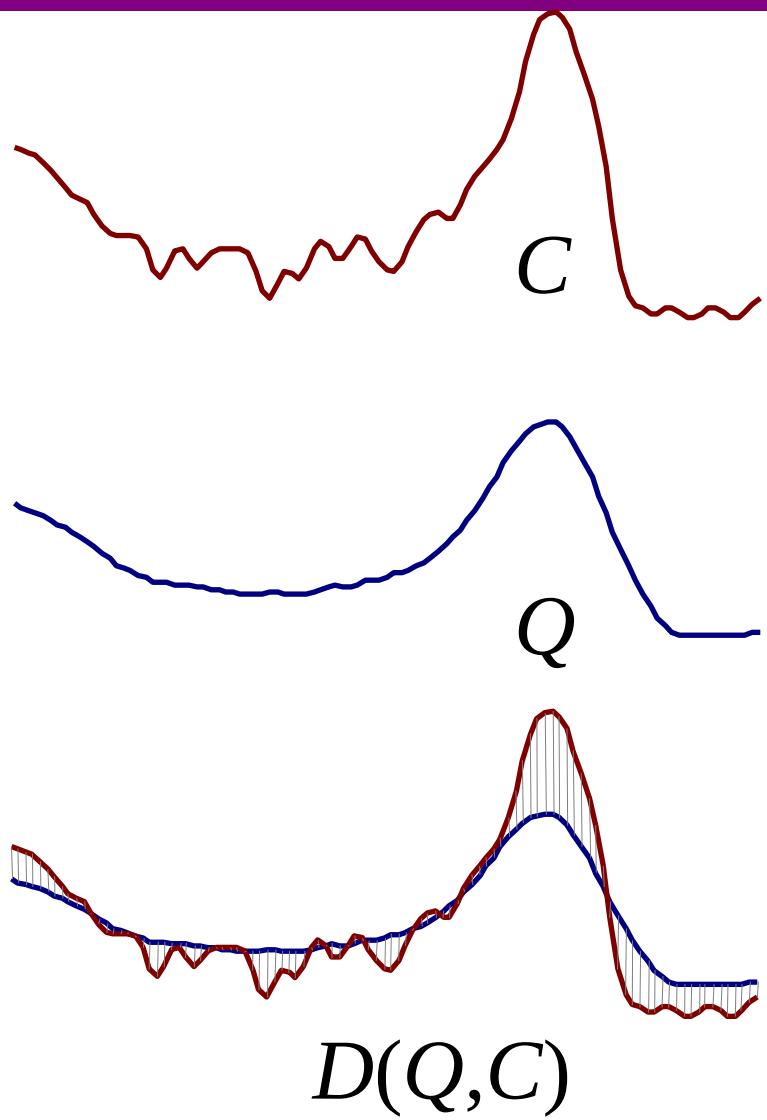
$$Q = q_1 \ldots q_n$$

e

$$C = c_1 \ldots c_n$$

la loro distanza euclidea è definita come:

$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2}$$



Ottimizzazioni nel calcolo della distanza Euclidea

$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2}$$

$$D_{squared}(Q,C) \equiv \sum_{i=1}^{n}(q_i - c_i)^2$$

Invece di utilizzare la
distanza Euclidea
si usa la
distanza Euclidea quadratica

La distanza euclidea e la distanza euclidea quadratica sono equivalenti nel senso tornano lo stesso rankings, clusterings e classificazione.

Preprocessare i dati prima del calcolo della distanza

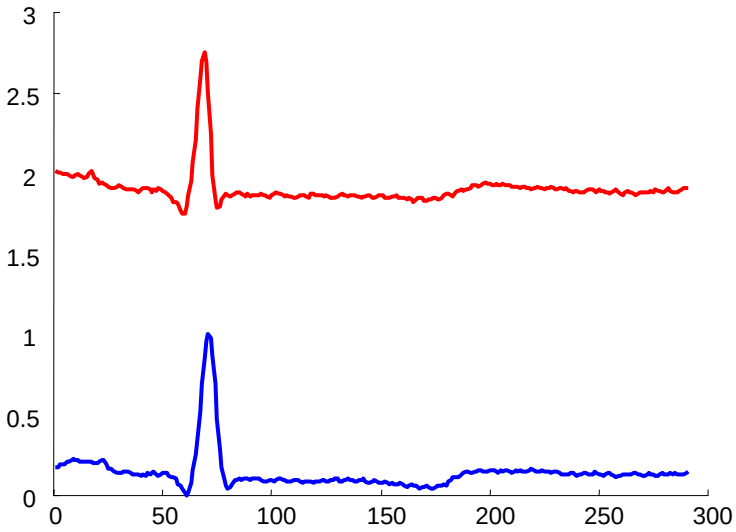
Se provassimo in modo naive a misurare la distanza tra due serie temporali grezze, possiamo ottenere risultati intuitivi.

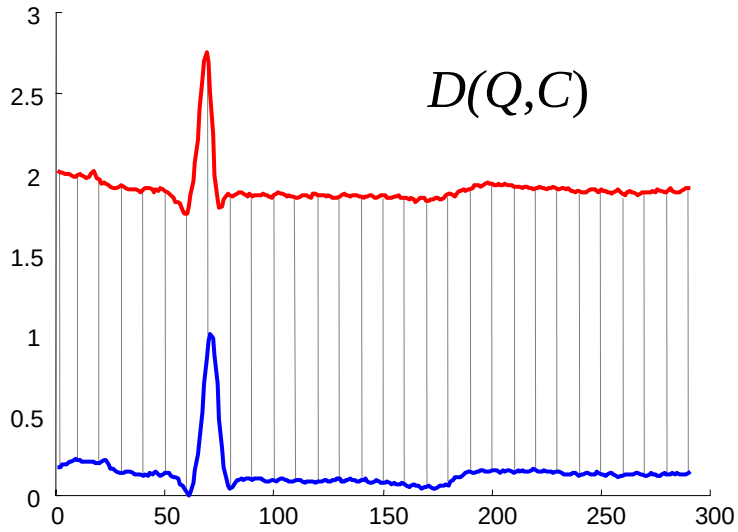
Ciò è dovuto al fatto che la distanza Euclidea è molto sensibile ad alcune distorsioni nei dati. Per molti problemi queste distorsioni non sono significative e quindi possono essere rimosse.

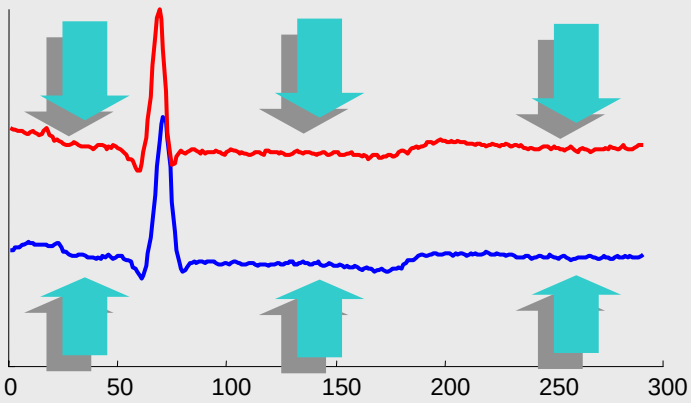
Vedremo ora le 4 più comuni distorsioni e come rimuoverle.

- Offset Translation
- Amplitude Scaling
- Linear Trend
- Noise

Trasformazione I: Offset Translation



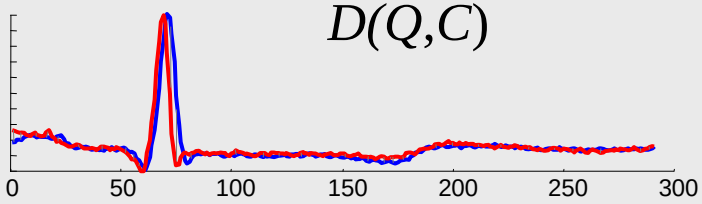




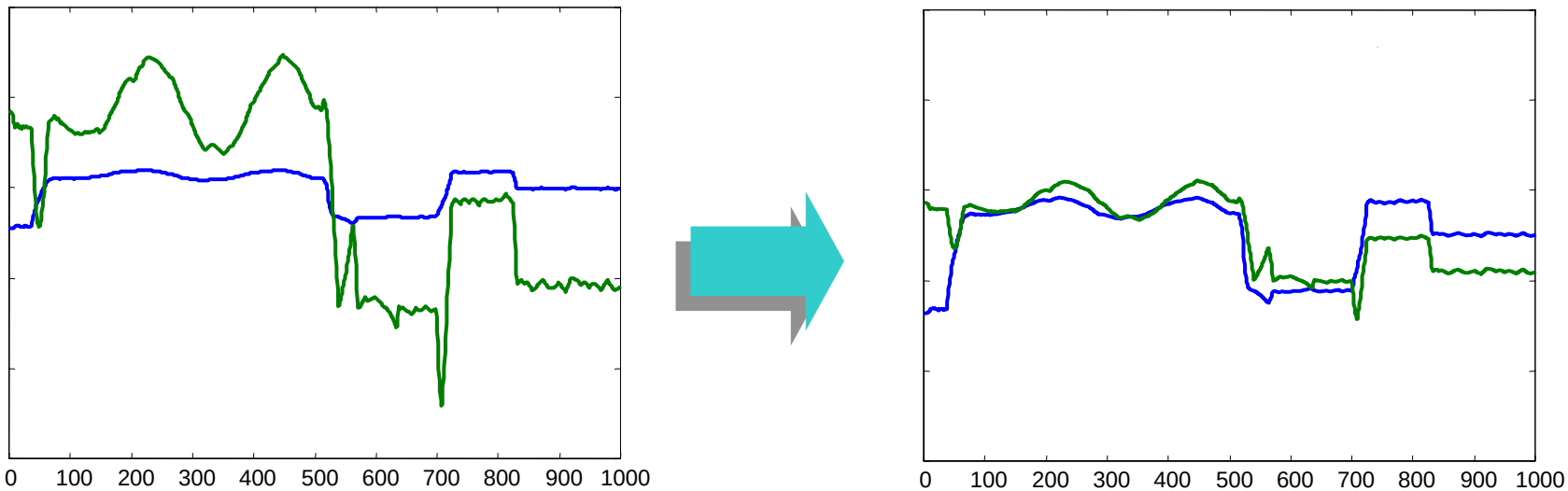
$Q = Q - \text{mean}(Q)$

$C = C - \text{mean}(C)$

$D(Q,C)$



Trasformazione II: Amplitude Scaling

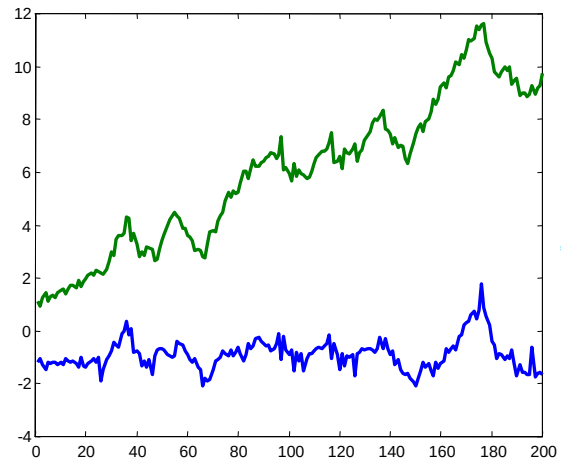


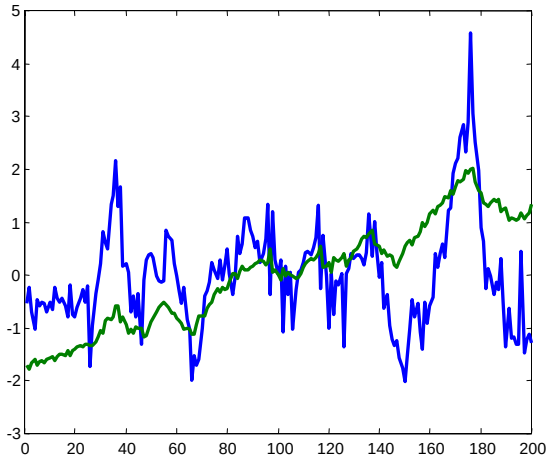
$Q = (Q - \text{mean}(Q)) / \text{std}(Q)$

$C = (C - \text{mean}(C)) / \text{std}(C)$

$D(Q,C)$

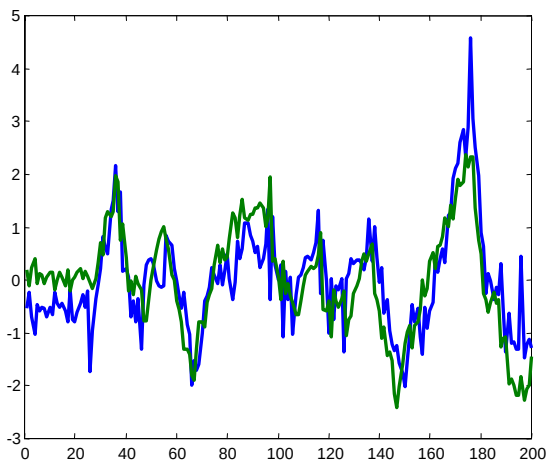
Trasformazione III: Linear Trend





Removed offset translation

Removed amplitude scaling

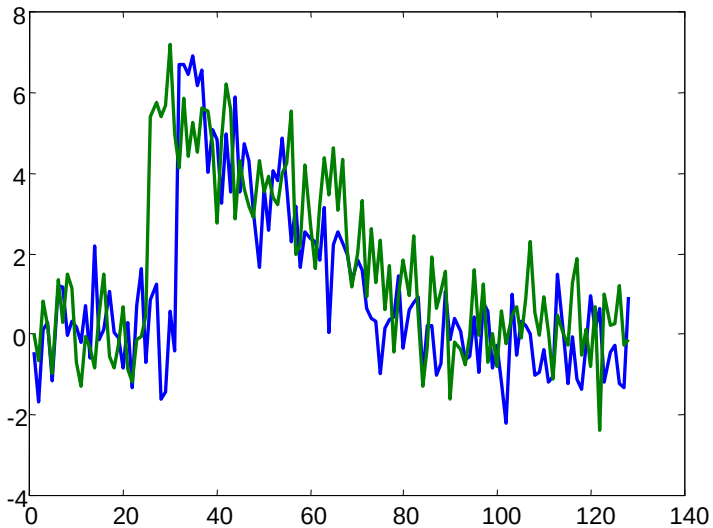


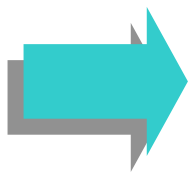
Removed **linear trend**

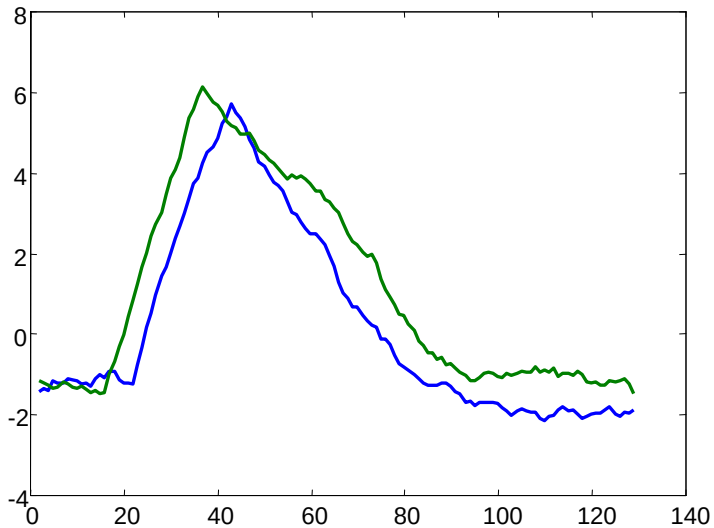
Removed offset translation

Removed amplitude scaling

Trasformazione IIII: Noise







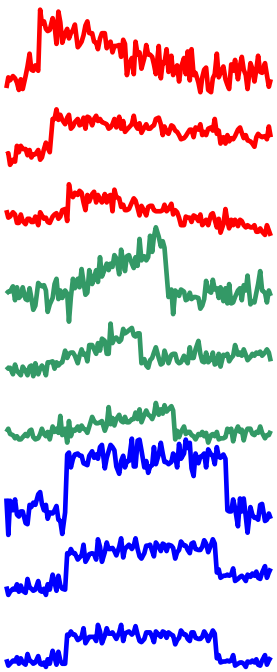
Media ogni datapoint con i suoi vicini.

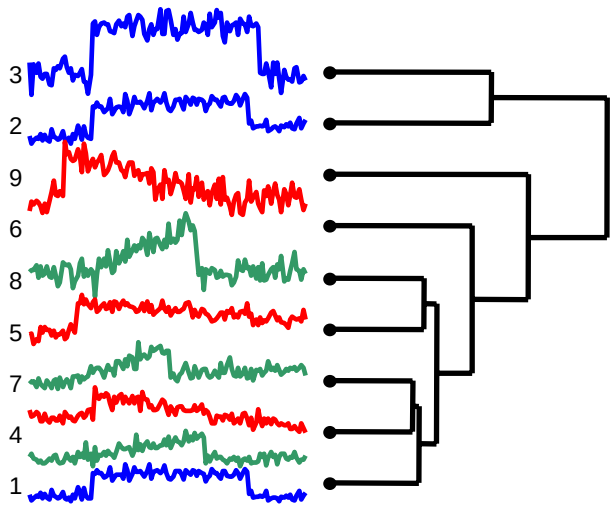
$Q = \text{smooth}(Q)$

$C = \text{smooth}(C)$

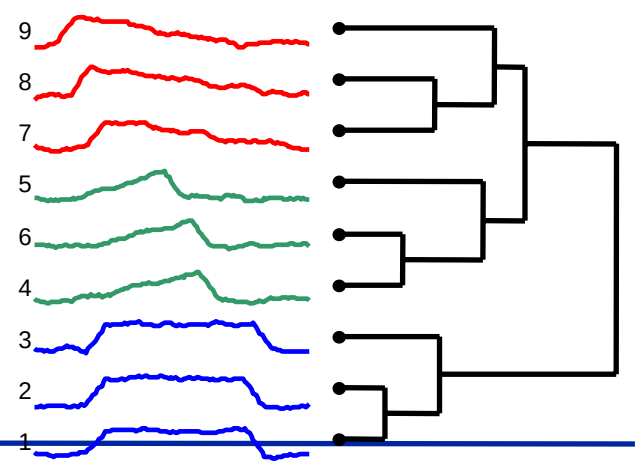
$D(Q,C)$

Un veloce esperimento per dimostrare l'utilità del preprocessing



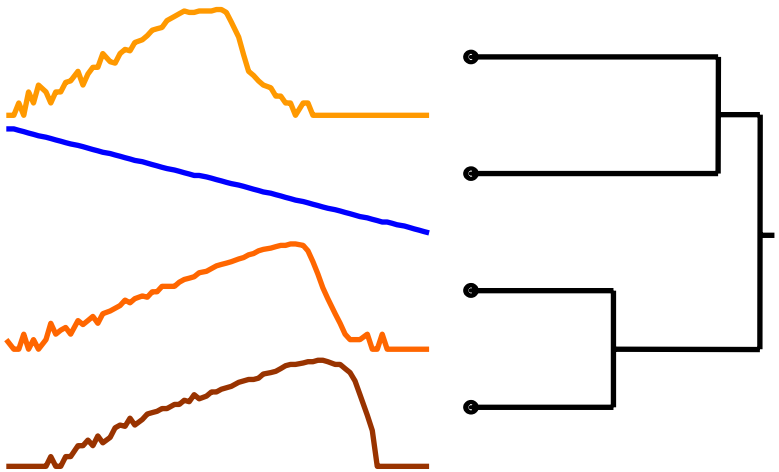


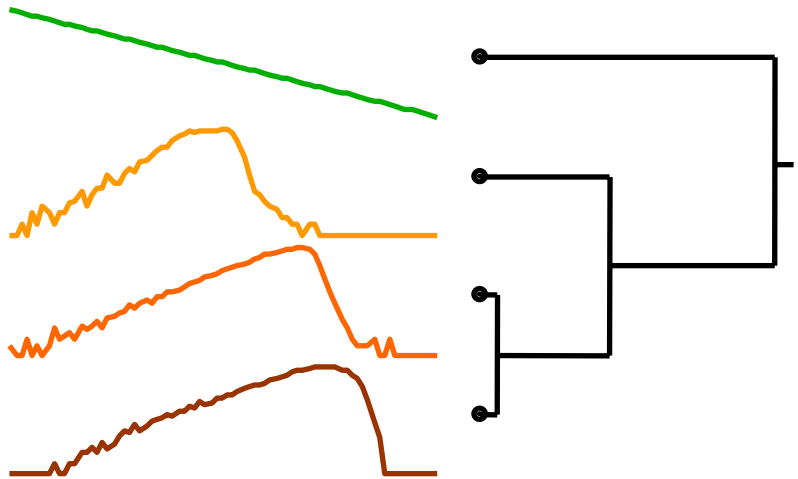
Clustering usando la distanza euclidea sui dati grezzi

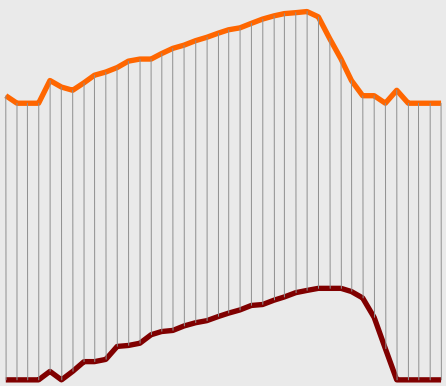


Clustering usando la distanza euclidea sui dati dopo averli preprocessati secondo i passi indicati

Dynamic Time Warping

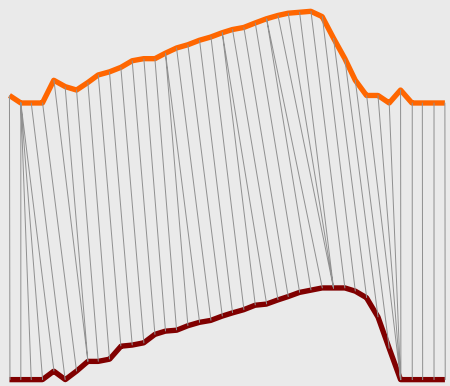






Fixed Time Axis

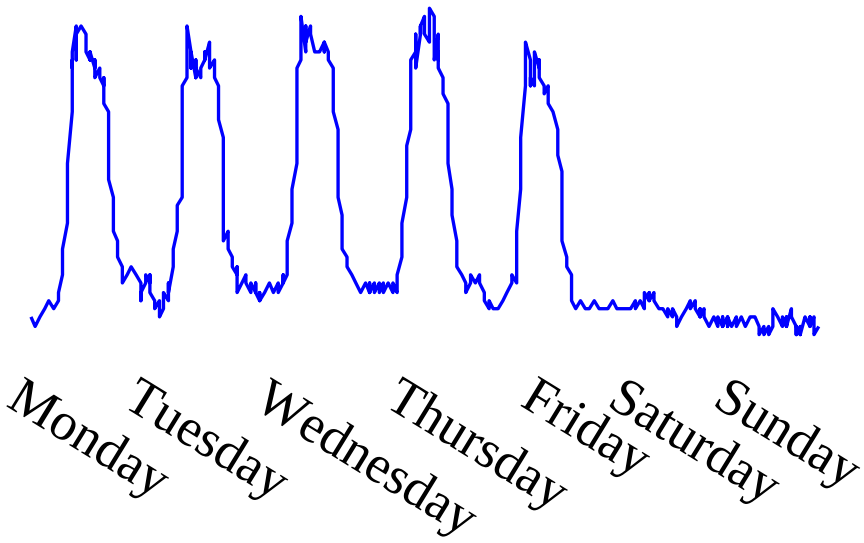
Le sequenze sono allineate “uno a uno”.



“Warped” Time Axis

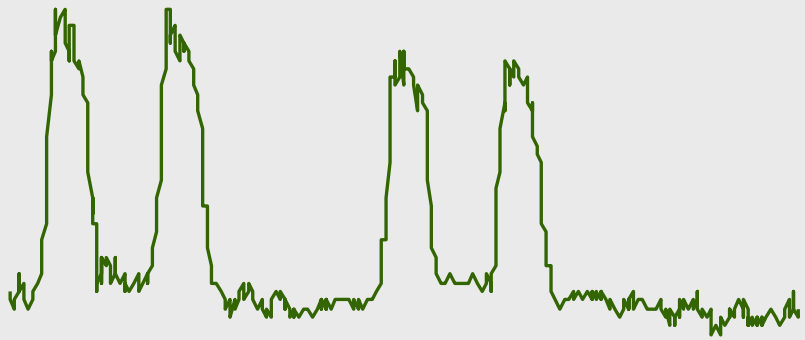
Sono possibili assegnamenti non lineari.

Utilità del Dynamic Time Warping: un esempio

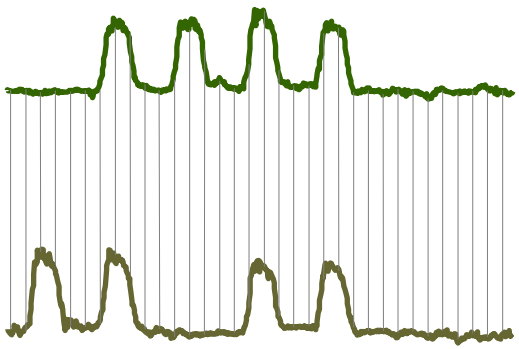


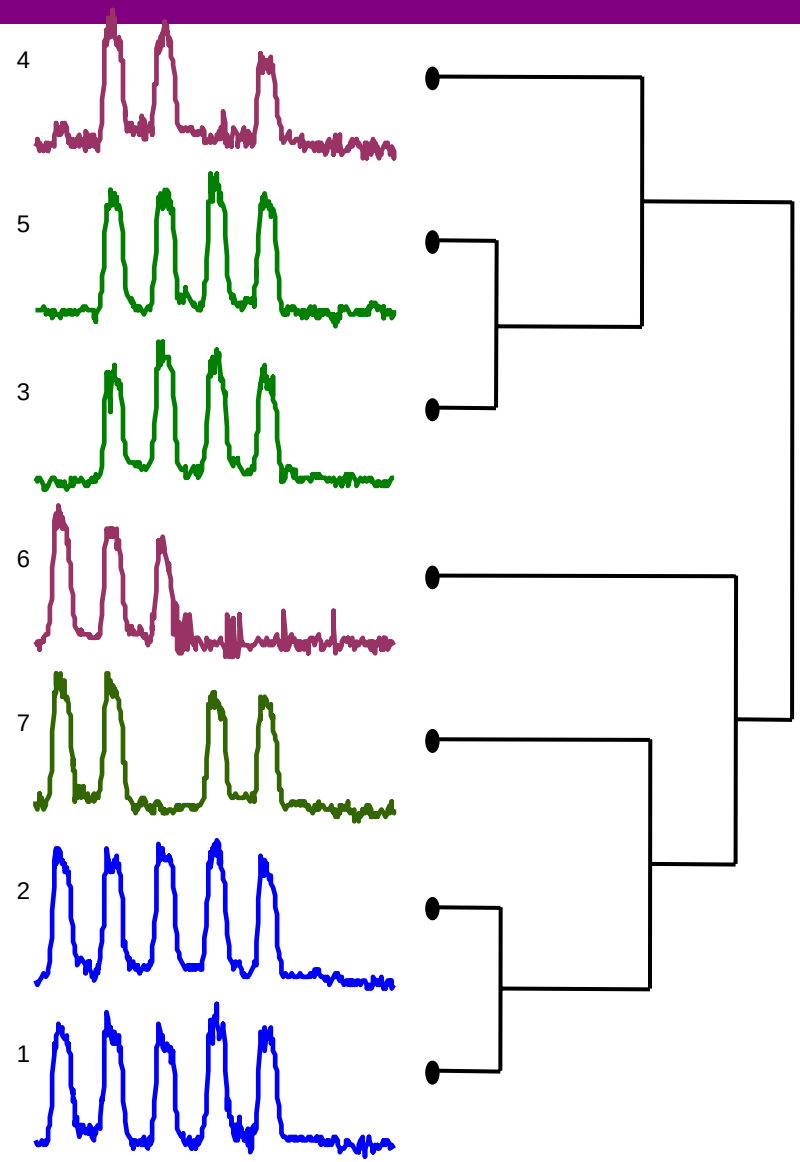
Serie temporali che mostrano la richiesta di energia elettrica. Ogni sequenza corrisponde ad una settimana

Mercoledì era festa nazionale



Clustering gerarchico con la distanza Euclidea.



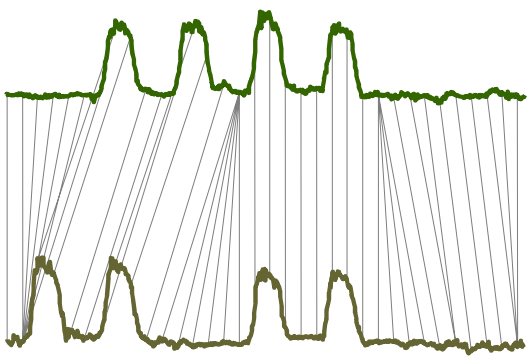


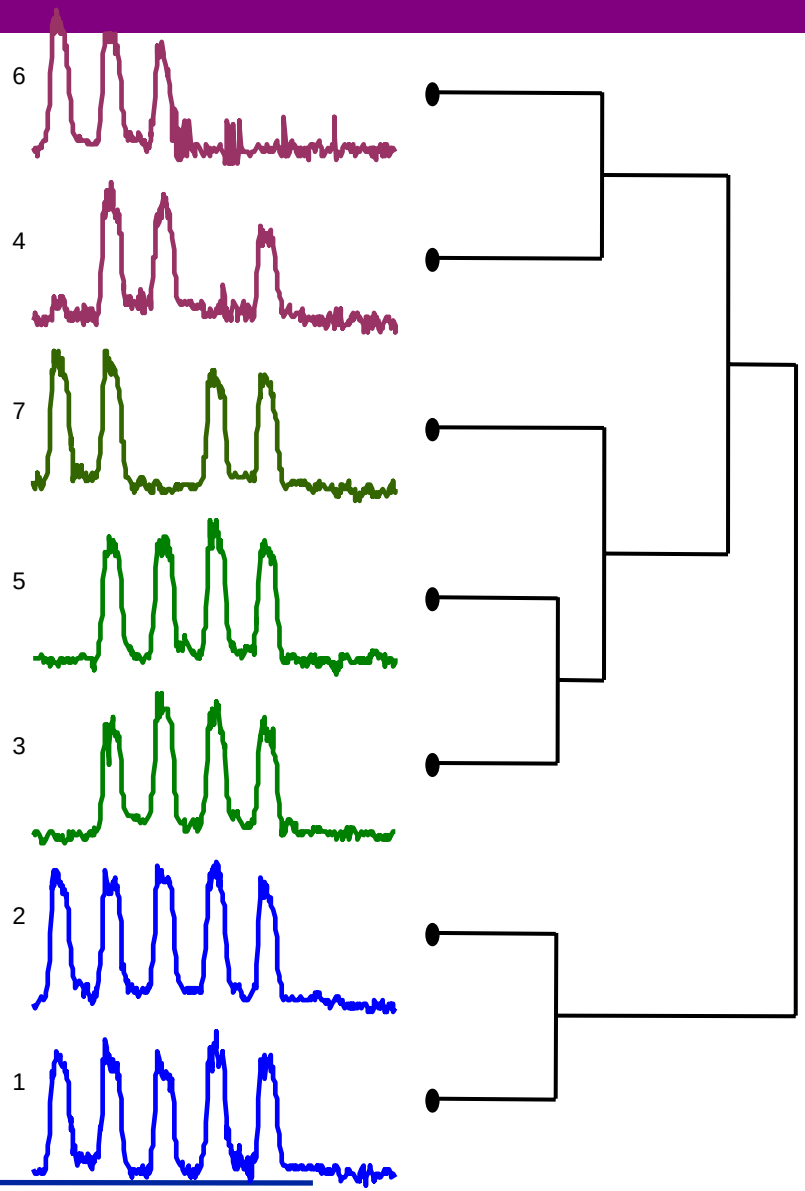
Le due **settimane di 5 giorni** sono correttamente raggruppate.

Notiamo tuttavia che le tre **settimane di 4 giorni** non sono clusterizzate insieme.

Anche le due **settimane di 3 giorni** non sono messe insieme.

Clustering gerarchico con Dynamic Time Warping



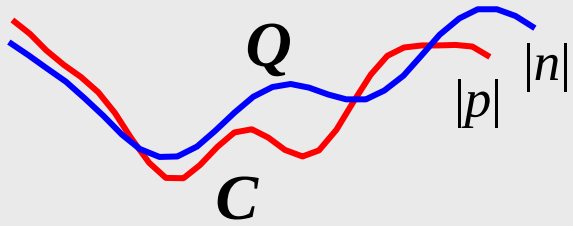


Le due **settimane di 5 giorni** sono correttamente raggruppate.

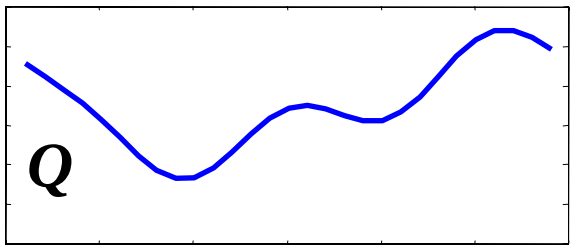
Le tre **settimane di 4 giorni** sono clusterizzate insieme.

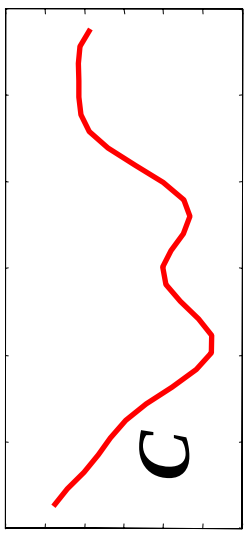
Le due **settimane di 3 giorni** sono ancora poste correttamente insieme.

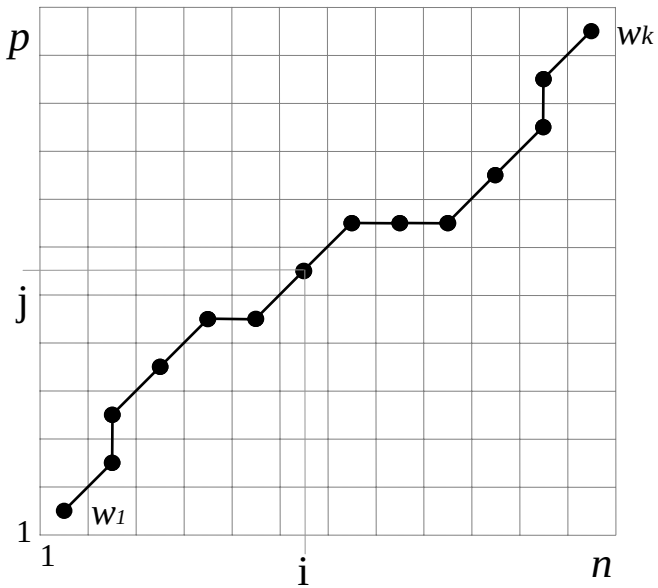
Calcolare la Dynamic Time Warp Distance (1)

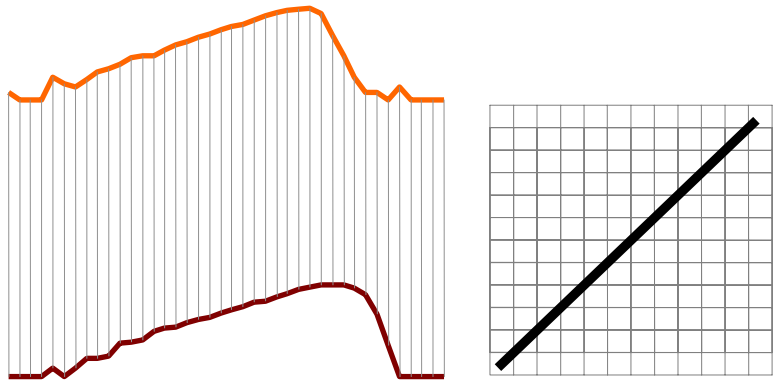


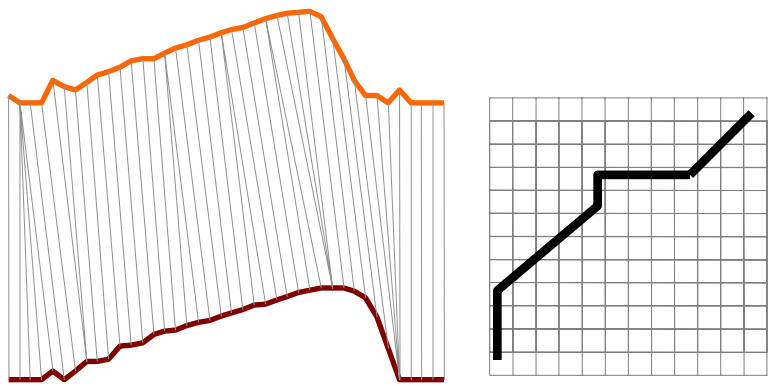
Notiamo che le sequenze di input possono avere lunghezza differente



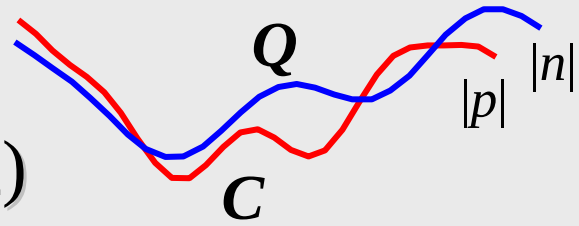


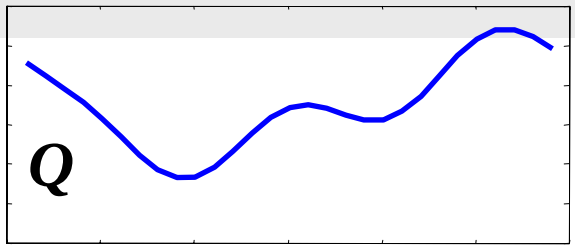


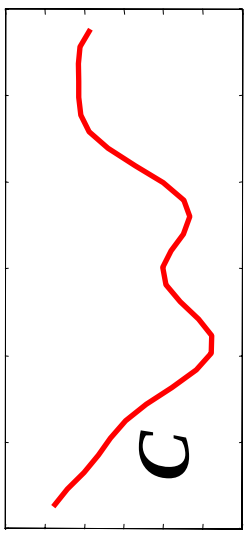


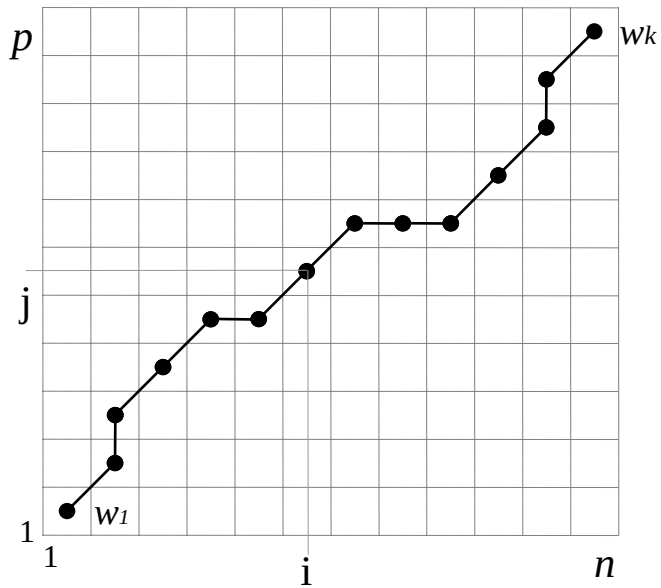


Calcolare la Dynamic Time Warp Distance (2)









Ogni possibile mapping da Q a C può essere rappresentato come un warping path nella matrice di ricerca.

$$DTW(Q,C) = \min\left\{ \sqrt{\sum_{k=1}^{K} w_k} \Big/ K \right\}$$

Nonostante ci siano un numero di cammini esponenziale, riusciamo a trovarli solo in tempoquadratico usando la programmazione dinamica.

$$\gamma(i,j) = d(q_i,c_j) + \min\{ \gamma(i-1,j-1) , \gamma(i-1,j) , \gamma(i,j-1) \}$$